Electronic Attacks

Thomas M. Chen and Jimi Thompson, *Southern Methodist University*
Matthew C. Elder, *Symantec Corporation*

INTRODUCTION

In today's society, computer systems are valuable, and often invaluable, for innumerable business and personal uses. Computer systems and networks are also very tempting as targets, shown by statistics that track the frequency and prevalence of cybercrimes. For example, Symantec Corporation estimates that organizations were hit by an average of 11 attacks daily during the first half of 2004 (Turner, 2004).

Part of the temptation is the ease of electronic attacks. Although not every attack takes advantage of vulnerabilities, it is widely known that computer systems have numerous vulnerabilities. In early 2004, about 48 new vulnerabilities were discovered weekly on average (Turner, 2004). Moreover, 96 percent of them were serious enough to be rated as moderately or highly severe. Attackers are keenly aware of new vulnerabilities because it takes time for organizations to set up adequate protection, e.g., software patching. In early 2004, exploits for new vulnerabilities appeared on average only 5.8 days after announcement of the vulnerability.

Electronic attacks have also become easier since virtually all computers are interconnected by the Internet or private networks. Moreover, mobile and handheld devices with Internet connectivity have steadily grown in popularity. This extensive network environment facilitates remote attacks and makes attacks more difficult to track to their sources. The growing number of networked machines also means more targets to attract attacks.

This chapter gives an overview of electronic attacks, highlighting the basic steps involved in attacks seeking to compromise computer systems. Most of the emphasis here is on network-enabled attacks, but this is not meant to imply that all electronic attacks are carried out through the network. This chapter also describes large-scale attacks such as viruses, worms, denial of service, and spam.

Types of Attackers

Attackers can be categorized in a number of different ways. One distinction often made is the relationship of the attacker to the target, either internal or external. Insider attacks from within an organization are believed to be the most common and most critical in past years. A commonly cited statistic in the late 1990's attributed 70 percent of all attacks to insiders. Insiders have certain advantages that can increase the likelihood of a successful attack, such as the trust of an organization and knowledge regarding systems and their defenses. However, with ubiquitous network connectivity today, external attacks are more likely than ever before (CERT, 2004).

Attackers can also be categorized as either amateurs or professionals. Many people probably visualize an attacker as the stereotypical male teenage "hacker" or "script kiddie" with too much free time. This stereotype has been perpetuated by fictional characters in films such as *War Games* as well as real-life arrested hackers. For example, the most recent case was the arrest of 18-year-old Sven Jaschan in May 2004. He is being prosecuted for writing the four most damaging worms of 2004, including the Netsky and Sasser worms, which accounted for 70 percent of the worms received in the world in the first half of 2004 (Sophos, 2004).

While teenage vandals are undoubtedly responsible for a substantial fraction of electronic attacks, it appears from recent trends that cybercrimes are being increasingly carried out by professionals and organized crime. Professional crimes tend to be more sophisticated and larger scale than amateur crimes. Attacks designed for identity theft and profit are becoming more prevalent. There are

P1: NPP
JWBS001B-74.tex WL041/Bidgoli WL041-Bidgoli.cls August 5, 2005 23:41 Char Count= 0

growing number of channels used for buying and selling lists of compromised computers and stolen identity data. Other professionals known to be involved in electronic attacks include national governments, military agencies, and industrial spies.

Attacker Goals and Motivations

The motivations for electronic attacks depend on the attacker. Because there are many different types of attackers, motivations can be almost anything, ranging from fun and fame to extortion, profit, espionage, revenge, or a political agenda.

The stereotypical teenage hacker is believed to be usually interested in gaining fame or notoriety. For example, according to some media accounts, Sven Jaschan appeared primarily motivated by curiosity and perhaps good intentions, writing Netsky. A to combat two other worms, MyDoom and Bagle.

On the other hand, messages encoded in the Bagle worm suggested that its authors were professionals motivated by profit. This is supported by the worm's actions including installation of backdoors for remote access (Symantec, 2004).

An increasingly common goal is invasion of privacy or theft of confidential data. This is evident from the escalation of spyware and phishing attacks (described later in this chapter).

Attack Targets

An electronic attack will have specific targets depending on the attacker's goals. The target could be particular information on a single machine, or the target could be as broad as the entire network infrastructure.

A recent survey showed that 70 percent of organizations were hit by some type of electronic attack (CERT, 2004). E-commerce was the most frequently targeted sector, as many attackers now are motivated by financial gain (Turner, 2004).

Attack Phases

An electronic attack is commonly carried out through a progression of steps, analogous to the steps of a physical attack (Chirillo, 2002; McClure, Scambray, and Kutz, 2001; Skoudis, 2002). The first step is reconnaissance to collect the necessary intelligence in preparation of the actual attack. The second step is the actual attack, which could have many different goals. During and after the attack, the attacker may try to take actions to avoid detection.

RECONNAISSANCE PHASE

If an attacker wants to compromise a specific computer system, it would obviously be wise to prepare for an attack by first discovering everything possible about the target. The reconnaissance phase can reveal a variety of information—account names, addresses, operating systems, perhaps even passwords—that could increase the success of an attack. Moreover, most reconnaissance techniques are not viewed as malicious or illegal, and may be carried out without a high risk of alarming a potential target.

As one might imagine, many different reconnaissance techniques are possible, and attackers do not follow a unique sequence of steps. Here we outline three general steps to progressively discover more information about a potential target.

Footprinting

The initial step in discovery is called footprinting, fingerprinting, or enumeration. An abundance of information is readily available on the Web. These databases can be interrogated by a number of utilities such as *nslookup* or *dig*.

The whois databases contain information about the assignment of Internet addresses, registration of domain names, and individual contacts. Domain names such as www.mycompany.com are registered through the Internet Network Information Center (InterNIC), a consortium of several companies and the U.S. government. For a domain name, the InterNIC whois database can provide the registrant's name and address, domain servers, and contact information.

For information about ownership of ranges of IP addresses, the American Registry for Internet Numbers (ARIN) database provides a mechanism for finding contact and registration information for resources including IP addresses, autonomous system numbers, and registered organizations in the Americas. European IP address assignments can be discovered from Réseaux IP Euoropéens Network Coordination Centre (RIPE NCC). Likewise, Asian IP address assignments are maintained by the Asia Pacific Network Information Center (APNIC).

Another useful database is the Domain Name System (DNS). DNS is a hierarchy of servers used to associate domain names, IP addresses, and mail servers. The hierarchy extends from the root DNS servers down to DNS servers for individual organizations and networks. These DNS servers contain information about other low-level DNS servers and IP addresses of individual hosts.

Scanning

Armed with information gained from footprinting, an attacker may know names and addresses for potential targets, and perhaps specific host system information. Footprinting is similar to looking up names and numbers in a telephone book. Scanning is a more active step to learn about potential targets from their responses. There are many different ways to conduct scans.

War Dialing. The most primitive though still useful type of scanning is war dialing. War dialers are simply automated machines for dialing a set of phone lines to find accessible modems. A telephone number within an organization is usually easy to find through the Internet or telephone books, then an attacker could dial a surrounding range of numbers. The results will reveal phone lines with modems. War dialers can include a nudging function that sends a predefined string of characters to a modem to see how it responds. The response may reveal the lack of a password, the type of platform, and perhaps a remote access program (e.g., the popular pcAnywhere).

P1: NPP
JWBS001B-74.tex WL041/Bidgoli WL041-Bidgoli.cls August 5, 2005 23:41 Char Count= 0

Although war dialers have been in use for decades, they can still be effective in attacks when a modem is not properly secured. Obviously, modems without password protection are completely vulnerable. In addition, modems can be attacked by guessing the password (see the section below on password attacks). A successful attack through an unsecure modem can lead to compromise of an entire organization's network, effectively bypassing firewalls and other sophisticated defenses.

Ping Sweeps. Internet Control Message Protocol (ICMP) is part of the Internet Protocol to enable notification of troubles and other control functions. Ping consisting of a pair of ICMP messages called Echo Request and Echo Reply are designed to verify that a specific host is operational. An IP-addressable host should reply to an ICMP Echo Request with an ICMP Echo Reply.

Ping is frequently used by attackers to scan a block of IP addresses for live hosts. Any number of tools can easily perform a ping sweep. However, since ping sweeps can be noticed, organizations will sometimes block ICMP messages. TCP packets to well known ports will also work, prompting a TCP SYN-ACK reply.

Network Mapping. Traceroute is a widely used utility for mapping a network topology. It cleverly takes advantage of the time-to-live (TTL) field in the IP packet header. The TTL field is set to the maximum time allowed for delivery of an IP packet. Each router decrements the TTL field by the time spent by the packet in that router, but routers typically forward packets quickly and are then forced to decrement the TTL by the minimum unit of one. Thus, the TTL field essentially serves as a hop count, where each router decrements the TTL field by one. If the TTL field reaches a value of zero, a router should discard the packet and send an ICMP Time Exceeded message back to the source IP address in the discarded packet.

Traceroute sends out a series of UDP packets, starting with a TTL field value of one and incrementing the value by one for each successive packet. When ICMP Time Exceeded messages are returned, they reveal the addresses of routers at various distances. An example of traceroute is shown in Figure 74-1. Similarly, ICMP messages could be used instead of UDP packets.

Port Scanning. TCP and UDP packets are sent to and received at specific ports indicated in the TCP and UDP headers. The headers allow a range of 65,535 TCP and 65,535 UDP ports. Certain "well known" port numbers are pre-assigned to common protocols. For example, Web servers listen for HTTP requests on TCP port 80. The other ports may be used dynamically as needed.

An attacker is very often interested to discover which ports are open on a potential target, i.e., which services are listening. However, probing every possible port manually would be very tedious. A port scanner is an automated tool for sending probes to a set of specific ports to see which ports are open. An example of a port scan is shown in Figure 74-2.

Operating System Detection. Knowledge of a host's operating system and its version is valuable to attackers because specific vulnerabilities are known for different operating systems. One technique used by attackers is TCP stack fingerprinting, implemented in the popular Nmap tool. While the TCP protocol is standardized in terms of its three-way connection establishment handshake, the standards do not cover responses to various illegal combinations of TCP flags. Operating systems can differ in their implementations of responses to illegal TCP packets. The idea of TCP stack fingerprinting is to probe for these differences with various illegal TCP packets until the operating system, even its particular version, can be identified (Fyodor, 2002).

Scanning Tools. Plenty of free and commercial scanning tools are available. Many of these are used for legitimate purposes by system administrators.

Sam Spade is a combination of useful reconnaissance tools, wrapped behind a Windows graphical user interface, including ping, whois, IP block whois (ARIN database query), nslookup, traceroute, and a function to verify email addresses on a specific mail server. A version of Sam Spade is available as a Web-based tool. Many other Web-based scanning tools can be found easily, such as a Web portal run by Mixter that includes ping, traceroute, whois, and port scans.

Other examples of free toolkits include CyberKit and Cheops. Cheops is a popular, easy-to-use tool for network mapping that automatically draws out a network topology based on discovered hosts and distances; it also discovers active services through port scanning and identifies operating systems by TCP stack fingerprinting.

An example of a commercial tool, NetScanTools Pro includes ping, port scans, traceroute, netscanner (ping sweep), custom ICMP packet generation, whois, nslookup, IP packet capturing, email address validation, and operating system identification. It uses an unusual method for operating system identification based on observing responses to four types of ICMP messages and variations of them. WildPackets' iNetTools is another commercial tool providing many of the functions as other scanners.

Finally, probably the most widely used tool for port scanning is the open-source Nmap shown in Figure 74-3. Nmap is perhaps the most capable port scanner, providing options for many different types of scans. Possible scans include TCP Connect, TCP SYN, TCP FIN, Xmas Tree, Null, TCP ACK, and UDP. Other interesting options in Nmap include: scanning for RPC (remote procedure calls) services on a target machine; sending decoy scans with fake source addresses; sending scans with different timing options to avoid detection; and identifying a computer's operating system via TCP stack fingerprinting.

Vulnerability Scanning

Using general network scanning, an attacker can discover a broad range of information about a potential target, such as host addresses, network topology, open ports, and operating systems. The next step in reconnaissance is to scan for specific vulnerabilities that might be exploited for an attack. It is possible to manually scan for vulnerabilities, but would be obviously time consuming to check

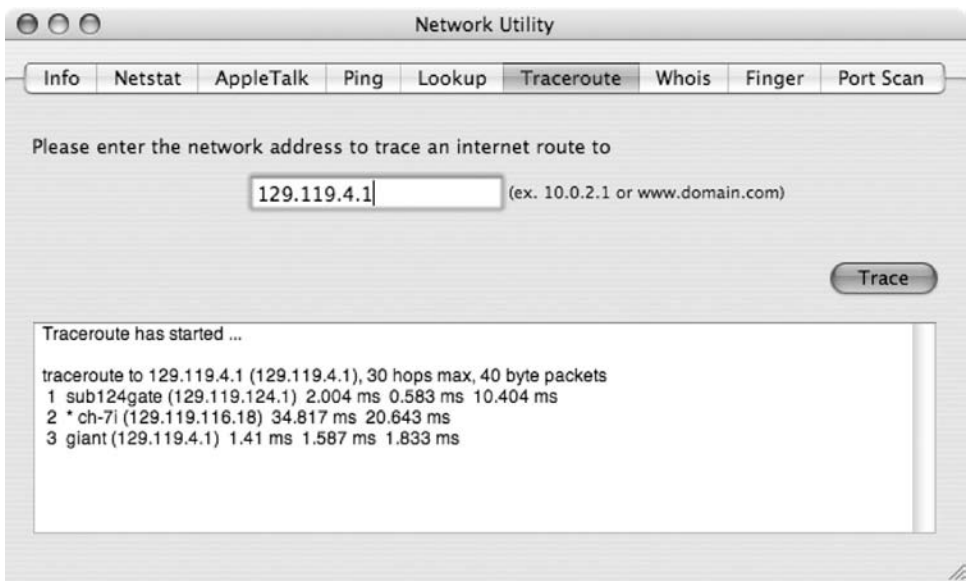

Figure 1: Example of traceroute output.

many machines for hundreds or thousands of vulnerabilities. Many automated vulnerability scanners are available and often used by system administrators to evaluate the security of an internal network.

Types of Vulnerabilities. Several types of vulnerabilities are usually sought by scanners:

- Default configuration weaknesses: Many operating systems and service applications ship with default accounts and passwords. These are intended to help ease the installation process, or simplify troubleshooting in case of lost passwords. Naturally, default passwords should be changed but are sometimes overlooked or ignored. Attackers look for the existence of default configurations

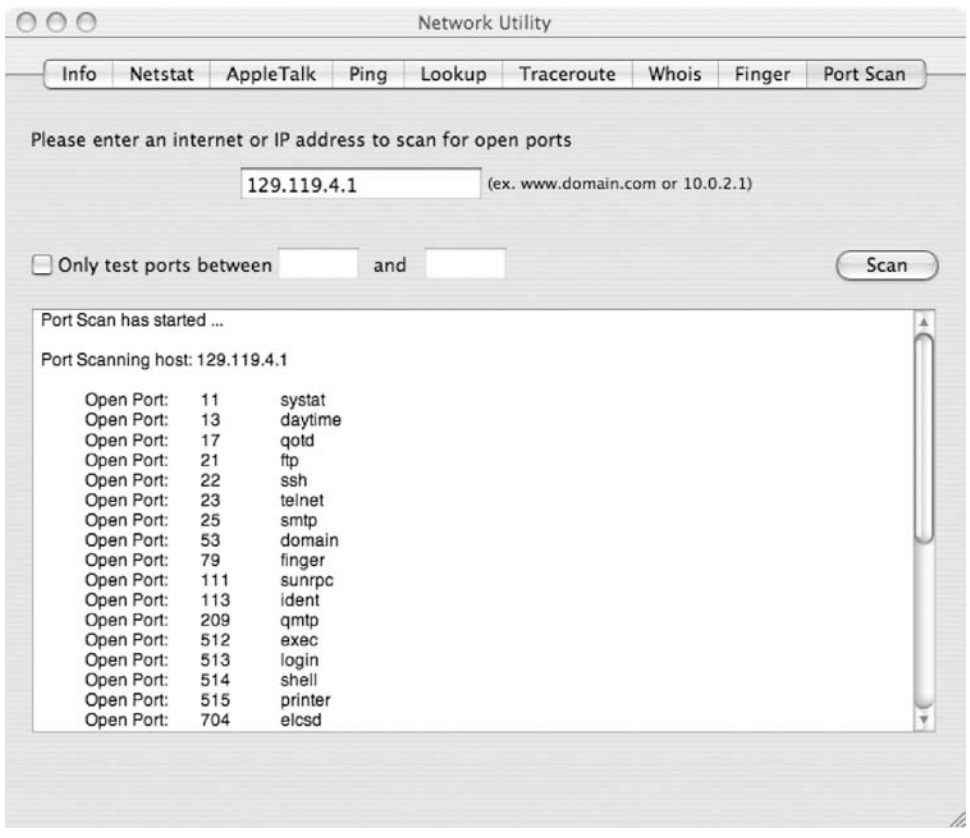

Figure 2: Example of a port scan.

P1: NPP
JWBS001B-74.tex WL041/Bidgoli WL041-Bidgoli.cls August 5, 2005 23:41 Char Count= 0

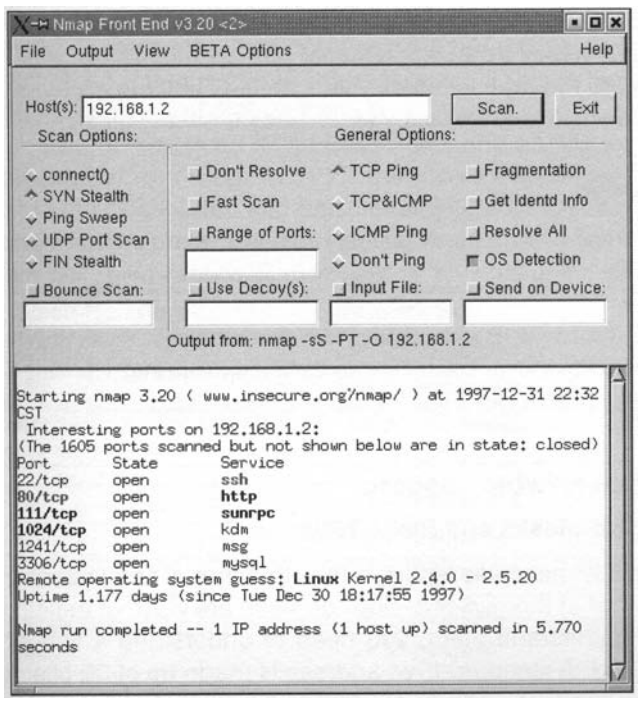

Figure 3: Nmap graphical user interface.

because they offer an easy way to compromise a system.

- Misconfiguration errors: Networking equipment requires expertise to configure properly. Obviously, incorrect configuration settings can defeat any security offered by networking equipment. An example is a misconfigured firewall that could be too permissive in allowing incoming packets.
- Well-known system vulnerabilities: New vulnerabilities are being constantly discovered in operating systems and applications. The most critical are often published by vendors with a patch. However, it requires a great deal of time and effort for organizations or individuals to keep up with security bulletins and patches. The time between the publication of a security vulnerability and the installation of patches leaves a window of opportunity for attackers to exploit that vulnerability.

Vulnerability Scanning Tools. A vulnerability scanner is an automated program generally consisting of: a vulnerability database to check; a user interface to allow control of the scanner; scanning engine to send and receive packets; knowledge base to track the current scan; and a recording and reporting tool (Skoudis, 2002). Many open-source and commercial vulnerability scanners can be found easily.

Most vulnerabilities scanner operate in a similar way. They first discover live hosts within a given address range using ping or similar utility. Then they run a basic set of scans to discover open ports and active services running on the hosts. Based on this information, they proceed to more customized probes for vulnerabilities. In the final step, they generate output in the form of a report. Some vulnerabilities scanners include a function for network mapping as a byproduct.

The Security Administrator's Tool for Analyzing Networks (SATAN) was an early well-known vulnerability scanner developed in 1995. While SATAN is still freely available, it has two updated descendents, the open-source Security Auditor's Research Assistant (SARA) and the commercial Security Administrator's Integrated Network Tool (SAINT). SARA enhances SATAN's security engine and program architecture by providing an improved user interface and up-to-date vulnerability tests. SARA can discover information about hosts by examining various network services. It can also find potential security flaws, such as misconfigured network services, well-known system vulnerabilities, or poorly chosen policies. It can generate a report of these results or execute a rule-based program to investigate any potential security problems.

Nessus is a popular open-source vulnerability scanner. It works in a client-server architecture, where the client and server may run on the same machine. The client consists of a tool for user configuration and a tool for recording and reporting results. The server consists of a vulnerability database, a knowledge base to keep track of the current scan, and a scanning engine. Nmap is included as the built-in port scanning tool. The vulnerability database is designed to be modular in the form of plug-ins, each plug-in to check for a specific vulnerability. Nessus contains over 500 plug-ins, and a large user base continually contributes new ones. Vulnerabilities are rated and classified into categories such as finger abuses, Windows-related vulnerabilities, backdoors, CGI (common gateway interface) abuses, RPC vulnerabilities, firewall misconfigurations, remote root access, FTP, and SMTP (mail server vulnerabilities).

Commercial vulnerability scanners include TigerTools' TigerSuite Pro, McAfee's CyberCop ASaP, ISS's Internet Scanner, eEye Digital Security's Retina Network Security Scanner, and Cisco Systems' Secure Scanner.

ATTACK PHASE

The actual attack phase can take many different forms and serve different purposes, such as stealing confidential data, tampering with the integrity of data, compromising the availability of a resource, or obtaining unauthorized access to a system. As mentioned previously, these specific attack types can be directed at either specific targets or the general network infrastructure. Quite often, large-scale, indiscriminate attacks have the effect of widespread disruption of computers and networks, even if that is not the real intent. They have widespread effects because they are carried out through a network towards a large number of targets.

The major types of attack covered here include sniffing, session hijacking, password attacks, exploits, social engineering attacks, Trojan horses, spyware and adware, viruses and worms, spam, and denial-of-service (DoS) attacks. The list is not meant to be exhaustive, but rather highlights of important attack types seen today.

These attack types are not mutually exclusive—in fact, many times they are combined in so-called blended threats. For example, social engineering can be a component of many e-mail worms. Some spyware is included

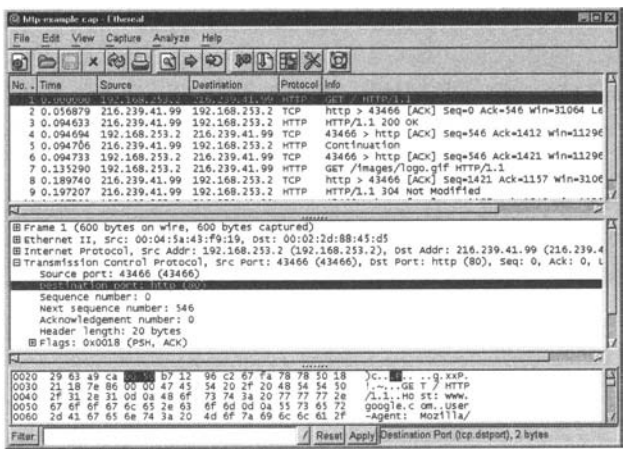

Figure 4: Example of Ethereal output.

as a Trojan horse in seemingly harmless software. Viruses can spread via spam, and so forth.

Sniffing

Sniffing is a form of passive attack that enables the compromise of confidential information. Sniffers, traditionally used by network administrators for traffic monitoring and LAN troubleshooting, have become one of the most commonly used attack tools. An example from the Ethereal sniffer is shown in Figure 74-4. On a LAN, hosts see all traffic broadcast on the LAN medium but normally ignore the packets that are addressed to other hosts. A sniffer program puts the network interface of a host into promiscuous mode to capture all packets seen at the interface. Thus, the sniffer can eavesdrop on everything transmitted on the LAN including user names, passwords, DNS queries, e-mail messages, and all types of personal data.

Many free and commercial sniffers are available, including tcpdump, windump, Snort, Ethereal, Sniffit, and dsniff.

Session Hijacking

Session hijacking is a combination of sniffing and address spoofing that enables the compromise of a user's remote login session, thus providing an attacker unauthorized access to a machine with the privileges of the legitimate user. Address spoofing in IP is quite simple because the sender of an IP packet writes in the IP source address in the packet header. Attackers can send packets with any fake IP source address.

If a user is currently engaged in an interactive login session (e.g., through telnet, rlogin, FTP), a session hijacking tool allows an attacker to steal the session. When most hijack victims see their login session disappear, they usually just assume that the cause is network trouble and try to login again, unaware of the hijacking attack.

Popular session hijacking tools include Juggernaut and Hunt. The hijacking attack begins with the attacker sniffing packets of an interactive session between two hosts, carefully noting the TCP sequence numbers of all packets. To hijack the session, the attacker injects packets with a source address spoofing one of the hosts. The proper TCP sequence numbers must be used for the attack to work, because the receiving host must be convinced to accept the faked packets from the attacker.

Password Attacks

Password attacks enable unauthorized access to a machine or other resource with the privileges of the user associated with the compromised password. Passwords continue to be very frequently used for access control, despite their major weakness: if a password is guessed, an attacker could gain complete access. The most well protected systems could be compromised by a single weak password. Understandably, many attacks are often directed at guessing or bypassing passwords.

The easiest passwords to guess are the default passwords installed by many operating systems and service applications. Extensive lists of default accounts and passwords are not hard to find by searching on the Web, and sometimes they are overlooked or ignored by system administrators.

Another easy password attack is a dictionary attack that, as the name suggests, takes advantage of the natural human instinct to choose passwords that are common words or names. The chance of finding passwords that are common words may not be as likely as in the past though, because modern systems are usually programmed with rules to prevent users from choosing easily guessable passwords. More sophisticated hybrid password guessing tools combine dictionary attacks with limited brute-force attacks. They begin with guesses of common words but then methodically add characters to words to form new guesses.

The most powerful password attacks, called password cracking, can be performed if the attacker can obtain the password file (Shimonski, 2002). Computer systems store a list of user accounts and passwords in a password file, but the information is encrypted or hashed for protection against attackers. If an attacker can obtain the password file, the attacker has the advantage of time (translating into more CPU cycles) to crack the passwords by brute force (i.e., attempting all possible combinations of legal characters). A few examples of password cracking tools include John the Ripper, Cain and Abel, Crack, Lincrack, L0phtcrack, Nutcracker, PalmCrack, and RainbowCrack.

A variation and extension of password attacks involves guessing usernames as well as passwords. Even if an attacker does not know a username associated with a given resource prior to beginning a password attack, many systems include commonly named accounts such as "Administrator" or "Guest". Automated attacks in particular, such as self-propagating worms, can incorporate password attacks that will guess both usernames and passwords when attempting to compromise a resource.

Exploits

Exploits of vulnerabilities are a means of attack that enable unauthorized access to a system. Vulnerabilities are continuously discovered in operating systems and application software. A vulnerability is a description of a problem, which is not dangerous in and of itself. The danger comes when an exploit is written that takes advantage

of a vulnerability to compromise the security of the operating system or application. Usually vulnerabilities are announced at the same time with a patch for fixing the vulnerability. A vendor has knowledge of the vulnerability but holds the information from the public at large until there is a fix for the problem. However, the vulnerability is sometimes announced prior to a patch, in which case there is an important race to devise and distribute a patch prior to the creation of an exploit taking advantage of that vulnerability. In general, the time between the announcement of a vulnerability and the appearance of a corresponding exploit is shrinking (Turner, 2004).

One of the most common types of exploit, used particularly often by worms, is a buffer overflow attack. Attackers are drawn to this exploit because many applications and operating systems do not perform proper bounds checking and are thus vulnerable to a buffer overflow. Furthermore, if successful, a buffer overflow attack could lead to complete control of a target machine.

A well-known instance is a stack-based buffer overflow, or "smashing the stack" (AlephOne, 1996). During a function call, various pieces of data are pushed onto the program stack: function-call arguments, return pointer, frame pointer, and local variables. Normally, at the end of the function call, the pieces of data are popped off the stack, and the return pointer is used to resume execution of the main program. A stack-based buffer overflow depends on inputting more data than expected into the local variables. The excess data is written into the allocated buffer space and then overwritten onto the frame pointer and return pointer. If the excess data can be crafted carefully enough, the overwritten return pointer can be made to point back into the stack somewhere in the data input by the attacker. Hence, when the main program resumes execution, the attacker's data (malicious code) will be run.

It might be observed that a buffer overflow attack requires careful coding and significant technical knowledge about the target processor architecture. Hence, buffer overflow attacks are not easy to craft from scratch, but pre-written exploits can be found and used even by novice attackers.

Social Engineering

Social engineering is a time-tested, low-tech approach that continues to be effective for both the reconnaissance and the actual attack phases. A social engineering attack refers to a human interaction where social skills are used to trick the victim into a compromising action, such as revealing personal information or opening an infected e-mail message. Social engineering can be combined with many of the other attack types to compromise security for just about any purpose. Although social engineering attacks are simple in concept, they can be surprisingly effective if executed well.

In the past, the telephone was a favorite avenue for social engineering attacks. Attackers would call an organization posing to be an employee, customer, supplier, or auditor, trying to obtain proprietary information. Today, many social engineering attacks are carried out through e-mail, due to the low risk and low cost of mass e-mailing. Also, e-mail works across different computing platforms and various types of devices (including handheld mobile devices). E-mail became the preferred medium after the success demonstrated by mass e-mailing viruses, such as the 2000 Love Letter and 2001 Anna Kournikova viruses. E-mail viruses typically offer a provocative reason to entice the recipient into opening (executing) an e-mail attachment, which results in a virus infection. More recently, e-mails might pretend to be security bulletins, bounced e-mail, notifications from an ISP or system administrator, or other official-looking messages.

Recently, a type of social engineering attack called phishing (password harvesting fishing) has escalated in frequency. Phishing attacks begin with e-mail seemingly from a reputable credit card company or financial institution that requests account information, often suggesting that there is a problem with an account or a transaction. These e-mails are carefully crafted to appear official and often include corporate graphics. The e-mails typically include a link directing the victim to a Website that appears to be genuine, but is actually a facsimile. The purpose of the Website is to capture any account or personal information submitted by the victim. An example of a phishing e-mail appearing to be sent from eBay is shown in Figure 74-5.

Trojan Horses

Trojan horses are malicious programs that appear to be benign (analogous to the Greek wooden horse in the Trojan War). The purpose of the disguise is to entice a user into installing and executing the program. If executed, Trojan horses are capable of doing anything that other programs can do, running with the privileges of the associated user. Similar and related to social engineering attacks, Trojan horses can be combined with many of the other attack types to compromise security for just about any purpose. Today, Trojan horses are distributed by any number of stealthy ways including virus and worm payloads, peer-to-peer file sharing, and Website downloads. Victims are often unaware of their installation.

The most worrisome Trojan horse may be backdoor programs, sometimes called remote access Trojans (RATs) because backdoors allow an attacker to remotely access a victim's machine. Backdoors circumvent the usual access control security (e.g., login with password). Many backdoor Trojans are known and some are promoted for legitimate administrative uses, including Sub7, Back Orifice 2000, and Virtual Network Computer (VNC).

Adware and Spyware

Adware is software to monitor and profile a user's online behavior, typically for the purposes of targeted marketing. Adware is often installed at the same time as other software programs; when this occurs without the user's knowledge, the adware (and the software with which it is bundled) is an instance of a Trojan horse. Even when the user is alerted to the presence of the adware (often buried in the ignored licensing agreement), adware can represent an attack on the privacy of the user and the confidentiality of the user's data when information about the user is communicated back to a marketing organization. Adware is primarily an annoyance,


From: eBay <eBay@eBay.com>
Subject: Account Violate The User Policy Second Notice
Date: December 1, 2004 5:25:40 AM CST
To:

Welcome to eBay

Dear valued customer

Need Help?

We regret to inform you that your eBay account could be suspended if you don't re-update your account information. To resolve this problems please click here and re-enter your account information. If your problems could not be resolved your account will be suspended for a period of 24 hours, after this period your account will be terminated.

For the User Agreement, Section 9, we may immediately issue a warning, temporarily suspend, indefinitely suspend or terminate your membership and refuse to provide our services to you if we believe that your actions may cause financial loss or legal liability for you, our users or us. We may also take these actions if we are unable to verify or authenticate any information you provide to us.

Due to the suspension of this account, please be advised you are prohibited from using eBay in any way. This includes the registering of a new account. Please note that this suspension does not relieve you of your agreed-upon obligation to pay any fees you may owe to eBay.

Regards, Safeharbor Department eBay, Inc
The eBay team.
This is an automatic message. Please do not reply.

Announcements | Register | Shop eBay-o-rama | Security Center | Policies | PayPal
Feedback Forum | About eBay | Jobs | Affiliates Program | Developers | eBay Downloads | eBay Gift Certificates
My eBay | Site Map
Browse | Sell | Services | Search | Help | Community

Copyright © 1995-2004 eBay Inc. All Rights Reserved.
Designated trademarks and brands are the property of their respective owners.
Use of this Web site constitutes acceptance of the eBay User Agreement and Privacy Policy.

TRUSTe site privacy statement

Figure 5: A fraudulent phishing e-mail pretending to be from eBay.

sometimes causing pop-up marketing windows during Web surfing.

A more serious and growing concern is another type of software that profiles and records a user's activities, called spyware. Similar to adware, spyware can sometimes be installed with a user's or system administrator's knowledge. For example, commercial versions of spyware are sold as means to monitor and regulate the online actions of children or an organization's employees. Often though, spyware can be installed stealthily on a machine as a Trojan horse or as part of a virus or worm compromise. Spyware can record keystrokes (also known as keystroke loggers), Websites visited, passwords, screenshots, and virtually anything done on a computer. After capturing data, spyware can communicate the stolen data by various channels (e.g., e-mail, FTP, upload to the Web, or Internet Relay Chat) to an attacker. Spyware, like adware, is an attack on user privacy, but spyware is also more likely to compromise confidential data for identity theft.

Viruses and Worms

Viruses and worms are software with the key characteristic of self-replication (Grimes, 2001; Harley, Slade, and Gattiker, 2001). While there is some debate and blurring of distinctions between viruses and worms, common traditional definitions are the following:

- Viruses are program code that replicate by modifying (infecting) a normal program or file with a copy of itself.
- Worms are stand-alone programs that replicate by spreading copies of themselves to other systems through a network.

Traditional viruses are not complete programs themselves. When the host program or file is executed, the virus code is executed and takes over control to copy itself to other files. Usually human action is needed to execute the host program, so viruses are sometimes characterized as requiring human action to replicate. Although viruses were far more common than worms around ten years ago, worms have become predominant in the past few years. The increase in worms has coincided with the growth of computer networks. Today virtually all computers are connected to private networks or the Internet, which is an environment naturally friendly to worms. In particular, the widespread popularity of e-mail has made it easier for worms to spread across different computing platforms. E-mail continues to be the most popular vector for worm propagation today, though typically e-mail worms require user intervention to propagate.

Viruses have evolved in their complexity over the years, often in response to countermeasures put in place by antivirus vendors. The first viruses often simply added their code to either the beginning or the end of the host file. In order to evade simple detection, viruses later began to intersperse their code throughout the host file. Another technique that viruses have adopted to evade detection is to encrypt their code within each host file instance, thus

P1: NPP
JWBS001B-74.tex WL041/Bidgoli WL041-Bidgoli.cls August 5, 2005 23:41 Char Count= 0

making it more difficult for a signature of the virus to be developed. When anti-virus programs began keying on the decryption algorithm as the signature, viruses became polymorphic, changing their decryption algorithm with each copy. Taking it one step further, some viruses have become metamorphic, i.e., they change their logic (not just the decryption algorithm) with each infection instance.

Worms that are standalone files have not had to evolve in the same way as file-infecting viruses. Functionally, a worm program must carry out a few specific steps to spread to another target after infection of a victim host.

First, an algorithm chooses candidates for the next targets. The simplest algorithm, which is used by quite a few worms, is to choose an IP address (32-bit number) at random. This is not efficient because the IP address space is not populated uniformly. More sophisticated target selection algorithms choose addresses within the same networks as the victim because local networks have shorter propagation delays to allow faster spreading. Other target selection algorithms may choose targets discovered from a victim's e-mail address book, mail server, DNS server, or countless other ways.

Second, many (but not all) worms will perform scanning of selected targets, for the same purpose as scanning done by human attackers. Scanning prompts responses from the potential targets that indicate whether the worm's programmed exploits can be successful. This process identifies suitable targets among the selected candidates.

The third step is the actual exploit or attack to compromise a suitable target. A common attack is to send e-mail to the target, usually carrying an infected attachment that has to be executed. More sophisticated e-mail worms are activated when their message is just previewed or read. Other worms might attack via file sharing, password guessing, or any number of exploits. It is also common for worms to combine multiple exploits or propagation vectors (blended threats) to increase the likelihood of success and rate of spreading.

The fourth step after successfully gaining access is to transfer a copy of the worm to the target. Depending on the exploit, a copy of the worm might have been transferred during the exploit (e.g., by e-mail). However, some exploits only create a means of access, such as a backdoor or shell. The worm takes advantage of the access to transfer a copy of itself via any number of protocols including FTP, TFTP, or HTTP.

An optional last step is execution of the worm's payload, if there is one. The payload is the part of the worm's program that is directed at an infected victim and not related to its propagation. The payload could be virtually anything, and not necessarily destructive. In recent cases, payloads have included: opening backdoors allowing remote access; installing spyware; downloading worm code updates from the Internet; or disabling anti-virus software.

Both viruses and worms are becoming easier to generate with the introduction of virus and worm toolkits. For example, the VBSWG (Visual Basic Script Worm Generator) toolkit simplified the process of creating e-mail worms for attackers – the Anna Kournikova worm in 2001 was produced using this toolkit. Other toolkits are easily found searching the Internet. Toolkits enable advances in malicious code technology to become commodities, easily reused by less experienced attackers for their own purposes.

Finally, there is a convergence occurring between viruses, worms, and other forms of malicious code. For example, there are instances of malicious code that both infect files like a virus and drop standalone copies of itself like a worm. Viruses and worms often possess characteristics of a Trojan horse, especially when they use social engineering to trick a user into aiding propagation. And viruses and worms can be used to enable the other forms of attack discussed next, both spam and denial-of-service attacks.

Spam

Spam, the e-mail equivalent of unsolicited junk mail, has been a growing problem over the past few years. E-mail addresses are harvested from the Internet or generated randomly. They typically advertise a product, service, or investment scheme (which may well turn out to be fraudulent). E-mail is an attractive advertising medium in economic terms. Spammers can send enormous volumes of e-mail at much less cost than postal mail. The necessary equipment is modest: a PC, software, and an Internet connection. Even if the response rate is very small, a sizable profit can be made easily.

At the very least, spam wastes network resources (bandwidth, memory, server processing) and necessitates spam filtering at ISPs and organizations. It also wastes users' time to read and delete. The seriousness of the problem has steadily grown as the volume of spam has escalated.

A growing concern with spam is evidence of collaboration between spammers, virus/worm writers, and organized crime. A substantial number of worms have been used as a delivery vehicle for Trojan horses that set up "bots." Bots listen for instructions from a remote attacker or allow backdoor access. A number of bots under coordinated control is a bot net. Bot nets are being used for distributed Dos attacks or spamming. Moreover, spam is increasingly being used for phishing (as described earlier). Phishing attacks attempting identity theft with increasing sophistication suggests the involvement of organized crime.

Denial of Service

Most people tend to think of denial of service (DoS) attacks as flooding, but at least four types of DoS attacks can be identified:

- starvation of resources (e.g., CPU cycles, memory) on a particular machine
- causing failure of applications or operating systems to handle exceptional conditions, due to programming flaws
- attacks on routing and DNS
- blocking of network access by consuming bandwidth with flooding traffic.

P1: NPP
JWBS001B-74.tex WL041/Bidgoli WL041-Bidgoli.cls August 5, 2005 23:41 Char Count= 0

There are numerous examples of DoS attacks. A "land attack" is an example of starvation. On vulnerable machines with Windows NT before service pack 4, the land attack would cause the machine to loop, endlessly consuming CPU cycles. The "ping of death" is an ICMP Echo Request message exceeding the maximum allowable length of 65,536 bytes. It caused earlier operating systems to crash or freeze (that programming flaw has been remedied in later operating systems).

The "Smurf" attack is an example of an indirect flooding attack, where the ICMP protocol is abused to cause many response packets to be sent to a victim machine in response to a broadcast packet. It is indirect because real attacker's address is not seen in any packets. It is also interesting as an example of amplification: a single attacker's packet is multiplied into many packets by the recipients of the broadcast.

The most harmful flooding attacks take advantage of amplification through a distributed DoS (DDoS) network (Dittrich, 2004). A famous DDoS attack occurred in February 2000 against several Websites including Yahoo, eBay, e*Trade, and others. Examples of automated DDoS tools include Trin00, TFN (tribe flood network), TFN2K, and Stacheldraht. In addition, viruses and worms have been known to infect victims with DDoS agents.

DDoS attacks generally proceed in two phases. The first phase is stealthy preparation of the DDoS network. The attacker attempts to compromise a large number of computers, often home PCs with a broadband connection, by installing a DDoS agent (i.e., a Trojan horse). DDoS tools such as Trin00 and TFN set up a two-level DDoS network. A small fraction of compromised machines are designated as "masters", waiting for commands from the attacker. The remainder of compromised machines are "daemons" waiting for commands from masters. The daemons carry out the actual flooding attack to a specified target.

DETECTION AVOIDANCE PHASE

During reconnaissance or an attack, an attacker would naturally prefer to avoid detection, which could trigger defensive actions. After a successful attack gaining access or control of a target, an attacker would like to hide evidence of the attack.

Evading Intrusion Detection Systems

Intrusion detection systems (IDSs) are designed to alert system administrators about any signs of suspicious activities. They are analogous in function to burglar alarms, a way to react in case intruders are able to penetrate preventive defenses (e.g., firewalls). Network-based IDSs monitor the network traffic and might be a stand-alone device or integrated in firewalls or routers. Host-based IDSs are processes that run on hosts and monitor system activities. IDSs are now commonly used by organizations. Naturally, an intelligent attacker would want to avoid detection by IDSs.

Without special precautions, an attacker could be easily detected by an IDS during reconnaissance because scanning tools are noisy. A port scan might involve thousands of packets, while a vulnerability scan could involve hundreds of thousands of packets. These scans may have an obvious impact on normal traffic patterns in a network. Moreover, these scans are exactly the signs that IDSs are designed to look for.

Most commercial IDSs attempt to match observed traffic against a database of attack signatures (i.e., misuse detection). Hence, an attacker could try to evade a signature match by changing the packets or traffic pattern of an attack. One approach to changing the appearance of an attack is to take advantage of IP fragmentation. An IDS must be able to reassemble fragments in order to analyze an attack. An IDS without the capability for fragment reassembly could be evaded by simply fragmenting the attack packets. An IDS might also be overwhelmed by a flood of fragments or unusual fragmentation.

IDS evasion is also possible at the application layer. For example, an IDS may have a signature for attacks against known weak CGI scripts on a Web server. An attacker may try to evade this signature by sending an HTTP request for a CGI script, but the HTTP request is carefully modified to not match the signature but still run on the Web server.

Another strategy for evading detection by IDSs, or other monitoring products, is to simply overload them with common, unimportant events to mask the actual attack. "Flying under the radar" of an IDS is easy to do when thousands of meaningless port scans and ping sweeps are filling the operators' consoles and logs, while a more sophisticated attack is executed.

Covering Up

Covering up evidence after an attack is particularly important if an attacker wants to maintain control of the victims. One of the obvious methods is to change the system logs on the victim's computers. Unix machines keep a running system log about all system activities, which can be viewed by system administrators to detect signs of intrusions. Likewise, Windows NT/2000/XP systems maintain event logs including logins, file changes, communications, and so on.

An attacker needs to gain sufficient access privileges, such as root or administrator, to change the log files. It is unwise to simply delete the logs because their absence would be noticed by system administrators searching for unusual signs. Instead, a sophisticated attacker will try to carefully edit system logs to selectively remove suspicious events, such as failed login attempts, error conditions, and file accesses.

Rootkits

Rootkits are known to be one of the most dangerous means for attackers to cover their tracks. Rootkits are obviously named for the root, the most prized target on Unix systems because the root user has complete system access. If an attacker has gained root access, it is possible to install a rootkit designed to hide signs of a compromise by selectively changing key system components. The rootkit cannot be detected as an additional application or process; it is a change to the operating system itself. For example, Unix systems include a program *ifconfig* that can show the status of network interfaces, including interfaces in

P1: NPP
JWBS001B-74.tex WL041/Bidgoli WL041-Bidgoli.cls August 5, 2005 23:41 Char Count= 0

promiscuous mode (or a sniffer). A toolkit could modify *ifconfig* to never reveal promiscuous interfaces, effectively hiding the presence of a sniffer. Another program *find* is normally useful to locate files and directories. A toolkit could modify *find* to hide an attacker's files.

Kernel-level rootkits have evolved from traditional rootkits. In most operating systems, the kernel is the fundamental core that controls processes, system memory, disk access, and other essential system operations. As the term implies, kernel-level rootkits involve modification of the kernel itself. The deception is embedded at the deepest level of the system, such that no programs or utilities can be trusted any more.

Covert Channels

Although logs and operating systems can be modified to escape detection, the presence of a system compromise might be given away by communications. For example, system administrators might recognize the packets from an attacker trying to access a backdoor. Clearly, an attacker would prefer to hide his communications through covert channels.

A common method used to hide communications is tunneling, which essentially means one packet encapsulated in the payload of another packet. The outer packet is the vehicle for delivery through a network; the receiver has to simply extract the inner packet. The outer packet is usually IP for routing through the Internet. Also, ICMP messages and HTTP messages have been used. Since the inner packet has no effect on network routing, any type of packet can be carried by tunneling.

CONCLUSIONS

Computer systems are common targets for a wide range of electronic attacks. Instead of an exhaustive catalog, this chapter has attempted a quick tour of the most pressing types of attacks in preparation for later chapters with more details.

An understanding of attacks is necessary in order to design strong electronic defenses. This chapter has not addressed electronic defenses, which will be covered by other chapters. We have seen that attacks can be viewed as a sequence of phases proceeding from reconnaissance to attack to covering up. An understanding of the methods and tools used in each attack phase can be helpful in fortifying cyber defenses.

GLOSSARY

Adware a type of software to monitor and profile a user's online behavior.

Backdoor a means for an attacker to remotely access a system.

Buffer overflow an attack on programs without bounds checking that allows arbitrary attack code to be executed remotely on a target system.

Denial of service a type of attack on the proper operation of a system or service through exhaustion of system resources, exhaustion of bandwidth, exploitation of programming bugs, or attacks on routing and DNS.

Firewall a security system intended to protect an organization's internal network against threats from an external network, using configurable filtering rules.

Footprinting the initial process of discovering and identifying potential targets.

Intrusion detection system a device to monitor network traffic or system activities to search for signs of intrusions.

Pfishing a social engineering attack luring e-mail victims to a fake Website for the purpose of stealing personal data, such as account passwords.

Port scan a probe to TCP or UDP ports to discover whether a service is listening.

Reconnaissance the process of collecting information about potential targets in preparation for an attack.

Rootkit tools to change system components to evade detection of an intrusion.

Session hijacking an attack to eavesdrop on an active session and take over control by impersonating one of the hosts.

Sniffer a program to passively intercept and copy network traffic typically on LANs.

Social engineering an attack attempting to persuade or trick victims into a compromising action, such as revealing

Spam unsolicited junk e-mail, usually sent in bulk to many addresses.

Spyware a type of software to attack privacy by stealing personal data.

Trojan horse a program appearing to be useful but actually containing malicious functions.

Virus program code that executes during the execution of an infected host program to copy itself to other programs.

Vulnerability a weakness or flaw that may be exploited by an attack to compromise a system or service.

Worm a self-replicating program that automatically attempts to copy itself to other systems across a network.

CROSS REFERENCES

See *Computer Viruses and Worms; Denial of Service Attacks; Hoax Viruses and Virus Alerts; Hostile Java Applets; Spam and the Legal Counter Attacks; Trojan Horse Programs*.

REFERENCES

Aleph One, "Smashing the stack for fun and profit," available at http://www.insecure.org/stf/smashstack.txt (date of access: Oct. 1, 2004).

CERT, "2004 E-crime watch survey shows significant increase in electronic crimes," available at http://www.cert.org/about/ecrime.html (date of access: Oct. 1, 2004).

Chirillo, J. (2002). *Hack Attacks Revealed*, 2nd ed. Indianapolis, IA: Wiley Publishing.

Dittrich, D., "Distributed denial of service (DDoS) attacks/tools," available at http://staff.washington.edu/dittrich/misc/ddos/ (date of access: Oct. 1, 2004).

Fyodor, "Remote OS detection via TCP/IP stack fingerprinting," available at http://www.insecure.org/nmap/nmap-fingerprinting-article.html (date of access: Oct. 1, 2004).

Grimes, R. (2001). *Malicious Mobile Code: Virus Protection for Windows*. Sebastopol, CA: O'Reilly.

Harley, D., Slade, D., and Gattiker, U. (2001). *Viruses Revealed*. New York: McGraw-Hill.

McClure, S., Scambray, J., and Kutz, G. (2001). *Hacking Exposed*. 3rd ed. New York: McGraw-Hill.

Skoudis, E. (2002). *Counter Hack: A Step-by-Step Guide to Computer Attacks and Effective Defenses*. Upper Saddle River, NJ: Prentice Hall PTR.

Shimonski, R., "Introduction to password cracking," available at http://www-106.ibm.com/developerworks/library/s-crack/ (date of access: Oct. 1, 2004).

Sophos, "Suspected Sasser worm author caught; could trigger more arrests,: available at http://www.sophos.com/virusinfo/articles/sasserarrest.html (date of access: Oct. 1, 2004).

Symantec Corp., "W32.Beagle.B@mm," available at http://securityresponse.symantec.com/avcenter/venc/data/w32.beagle.b@mm.html (date of access: Oct. 1, 2004).

Turner, D., et al. (2004). *Symantec Internet Security Threat Report: Trends for January 1, 2004–June 30, 2004*. available at http://www.symantec.com (date of access: Oct. 1, 2004).

FURTHER READING

A number of Websites contain information and software related to the attack tools mentioned in this chapter:

- Packetstorm, available at http://www.packetstormsecurity.org
- Operation:Security, available at http://www.operationsecurity.com
- Insecure.org, available at http://www.insecure.org/tools.html

Well-known Websites with literature and advisories about vulnerabilities include:

- CERT, available at http://www.cert.org
- SANS, available at http://www.sans.org

Anti-virus corporate Websites are a good source of information about malicious code:

- McAfee, available at http://www.mcafeee.com
- Sophos, available at http://www.sophos.com
- Symantec, available at http://www.symanec.com
- TrendMicro, available at http://www.trendmicro.com