Cybercrimes: A Multidisciplinary Analysis

A Multidisciplinary Analysis

Bearbeitet von
Sumit Ghosh, Elliot Turrini

schnell und portofrei erhältlich bei

beck-shop.de
DIE FACHBUCHHANDLUNG

Chapter 2
Unauthorized Intrusions and Denial of Service

Alan Boulanger

2.1 Unauthorized Intrusions

According to Carnegie Mellon University's Computer Emergency Response Team (CERT), the number of computer intrusion incidents reported have increased significantly every year. In 2001, for example, CERT reported over 52,000 computer security incidents; a 140% increase in the number of reported in the previous year. In 2009, F-Secure reported that over 9,000,000 computers worldwide had been compromised and infected with a single type of malicious software.

In collaboration with the Federal Bureau of Investigation's (FBI) Computer Crime Squad in San Francisco, the Computer Security Institute has conducted annual surveys of the computer security practitioners working for US corporations, medical institutions, universities and financial institutions. The results of the 2002–2005 surveys are compelling. Of the respondents, 90% reported a computer security incident within the last 12 months, while 80% reported financial losses related to the security breaches. Of the entities reporting losses, the respondents were able to quantify the amount of damages and they reported combined losses exceeding $450 million annually in 2002. Since 2005, the rate of direct monetary losses through security incidents appears to have stabilized in the $130 million to $200 million range of those reporting organizations. In 2006 the FBI conducted an extensive computer crime survey. After analyzing the results of over 2,000 participants, the FBI reported the total cost of all computer crime to U.S business exceeds $67 billion per year.

Individual information security incidents are too commonplace to be cataloged and reported effectively. This is the result of a major shift in the tools, techniques and targets of the hackers. Historically hackers would target the core systems of public and private sectors. This activity is ongoing and these organizations responded with significant investment in building electronic walls protecting their critical infrastructure. The hacker community has adapted to these defenses and are now focusing their attacks on endpoint computer systems using trojans, viruses, and other assorted malicious software.

Information security is a very serious issue. The media have reported a substantial number of recent attacks on high profile sites, and the number of reported

security-related incidents continues to rise. In 1996 the United States Department of Defense (DoD) reported an estimate of 250,000 attacks per year on its computer systems and stated that the rate of attacks is increasing exponentially. The trend has continued to this day.

A key contributing factor for this increase in attacks is the widespread availability of automated, malicious, software packages, or toolkits. Many of these packages are easy to acquire and easy to deploy. No longer are perpetrators required to possess an in-depth knowledge of the Internet or operating systems to successfully carry out attacks. These "point and click" attack tools provide a novice computer user the ability to attack and inflict damages on the victim computer's operations. Both hackers and computer security professionals have developed sophisticated software tools to either break into foreign computer systems or identify potential security breaches in computer networks. These tools are often found left behind in compromised systems and they are also present in the toolkits of legitimate "tiger" teams, authorized to attempt to break into computer systems with the full consent of the network owners.

Data recovered from post mortem analysis of compromised systems as well as from the computers exploited by perpetrators to launch attacks reveal strong similarities in how intruders seek out their targets and launch attacks on their victims. Many of the elements of the attack plan are observed to be automated and based on sophisticated software toolkits.

2.1.1 Tools to Exploit Unauthorized Intrusions

Available toolkits may be organized into six distinct categories, each of which comes with a set of tools and techniques [1] that had been developed to exploit a specific type of system vulnerability [2–4]. These include (1) scanners, (2) remote exploits, (3) local exploits, (4) monitoring tools, also known as sniffers, (5) stealth and backdoor tools, and (6) a new class of tool called the auto-rooter.

2.1.1.1 Scanners

A scanner extracts information about a host or network and comes in two basic categories. The first type of scanner, termed network auditing tools, are utilized to scan a remote host or a series of hosts on a network and report back security-related vulnerabilities. The second type of scanner, known as host-based static auditing tools, is used to report back the security vulnerabilities of a local host.

In 1992, Christopher Klaus developed and released in the public domain the Internet Security Scanner (ISS), which was one of one of the first network auditing tool set and included many of the common security tests. In 1994 and 1995, Dan Farmer of Sun Microsystems and Wietse Venema of Eindhoven University of Technology developed and released the Security Analysis Tool for Auditing Networks

(SATAN). SATAN expanded the functionality of ISS by including additional security tests and was designed to be portable, permitting it to be run on a larger variety of platforms. The popularity and ease of use of SATAN resulted in a large number of unauthorized scans of computer systems by hackers and merely inquisitive users. Today, the number of industrial-grade network security auditing packages available on the Internet has exploded. They are widely available on the World Wide Web (WWW), at anonymous FTP sites, and on underground bulletin boards. Many of the tools that have been used to successfully breach network security are easily found through searching the WWW. A few of the popular, freely available network scanners include:

- SATAN: Available from http://www.porcupine.org
- NESSUS: Available from http://www.tenablesecurity.com
- NMAP: Available from http://www.insecure.org

  A NESSUS user can access the following information for specified hosts:

- Host machines on the network that respond and, therefore, will permit subsequent communication
- Servers available on the responding hosts
- Shared disks available through Network File System (NFS) support
- File access through Network Information Service (NIS), a distributed database for shared information
- Remote execution capability
- Sendmail vulnerabilities, namely, versions that may be tricked into running bad commands
- Trivial File Transfer Protocol (TFTP) access and configuration, which can be used to download password files
- Remote shell access, which provides the ability to execute commands on a different system without explicitly logging in with a password
- Unrestricted X Window System server, which allows the hacker to connect to the server, spy on the users on the server, obtain passwords, and "freak out" users through painting "roaches" or "smiley faces" on their screens
- Readable or writeable File Transfer Protocol (FTP) directory, which allows the hacker to upload commercial software or pornographic material onto corporate computing systems

A specialized scanner, called host-based static auditing tools, is deployed to acquire unauthorized privileged access. It was originally developed to enable system administrators uncover common security weaknesses in a local system and thereafter "harden" it before hackers could intrude. In the hands of a clever hacker, the tool allows a perpetrator with an unprivileged account on the system to decipher the vulnerabilities and gain unauthorized privileged access. In 1989, Dan Farmer released one of the first static auditing tools, labeled the COPS package. COPS was a collection of scripts that scanned the local system, seeking out and reporting security vulnerabilities. In 1992, researchers at Texas A&M University developed and released the TIGER toolkit, which greatly expanded on the original ideas in

COPS. Both COPS and TIGER carry out extensive system checks and report on the following vulnerabilities:

- Permission problems in files, directories, and devices, which subsequently allows intruder access.
- Poor, easy-to-guess passwords.
- Poor security for password and group-definition files.
- Known vulnerable services, including anonymous FTP configuration and improperly configured services.
- Signs of past intrusions, particularly in key binary files.

2.1.1.2 Remote Exploits

Remote exploits include all software programs, methods, and techniques through which a foreign user, i.e., with no prior account on a given computer system, can penetrate into the system. The vulnerabilities associated with remote exploits stem from the services provided by computer systems in the network. In general, most services initiate or "open" a communication channel and monitor, i.e., "listen" for any incoming connection requests. For example, under sendmail, which processes electronic mail, the corresponding program will open a port and listen for incoming requests from other sendmail servers. When it detects a request, a sendmail server will "accept" the connection and communicate with the transmitting system or "client" on the network through a Simple Mail Transfer Protocol (SMTP). Where the sendmail server is vulnerable and the weakness may be exploited through data sent over by the client, the sendmail server's host is vulnerable to attack from unprivileged users on any connected system. Remote exploits represent one of the most feared and dangerous vulnerabilities and are, therefore, most closely guarded.

A subcategory of remote exploits is the protocol-based attack. In it, a software program is deployed to acquire unauthorized access by manipulating the Transmission Control Protocol/Internet Protocol network protocol suite, commonly referred to as TCP/IP. Vulnerabilities in TCP/IP have been well known for many years. As far back as 1985, R.T. Morris demonstrated a vulnerability through which a hostile system may hide its true identity and impersonate a different host's IP address. Where the victim computer system relies on address-based authentication, i.e., distinguishing between friend and foe through the IP address, a hostile attacker has a clear advantage in that it can completely circumvent the authentication process and gain access into the target system as a trusted peer. A generalization of this devastating attack coupled with other security-related vulnerabilities in TCP/IP were reported by Steve Bellovin in 1989. Additional vulnerabilities reported included session hijacking and IP spoofing, both at the level of User Datagram Protocol (UDP) and that of TCP; Routing Information Protocol (RIP) attacks; Internet Common Message Protocol (ICMP) attacks; and Border Gateway Protocol (BGP) attacks. The last reported, high-profile incident involving a protocol-based attack was Kevin Mitnick's TCP/IP-spoofing attack reportedly launched in December 1994 and the subject matter of the book, "Takedown," by John Markoff and Tsutomu Shimomura.

The vulnerabilities exposed by remote exploits served to motivate the development and deployment of firewalls and network auditing tools. As in an automobile, where the firewall separates the engine from the passenger compartment, a firewall in a networked computer system, say N, controls access to the services of N from the outside. In essence, the firewall hides information relative to the internal structure of the services and, often, strategically removes key sub-services, thereby minimizing the undue exposure of N.

2.1.1.3   Local Exploits

A local exploit resembles as insider attack in that a user with an existing account on a computer system exploits tools and services to acquire unauthorized privileges. This attack is commonly referred to as unauthorized user-to-root transition. The existing account may either be a prior legitimate one; acquired through a remote exploit; or obtained through trading information with other hackers, intercepting logon information from network traffic, or social engineering. Most local exploits stem from errors in a privileged program's software design and implementation that inadvertently allow an unprivileged user to execute hostile commands at a privileged level or access and modify privileged data. The instant privileged access is acquired, the hacker, in essence, is in complete control of the system. Exceptions notwithstanding, on most operating systems, the intruder is able to successfully modify the system logs to hide illicit activities, install a "backdoor" entrance that allows continuing privileged and unregistered access to the system. On average, new local exploits are reported at over three times the rate of new remote vulnerabilities and are widely available to anyone through security-related newsgroups, mailing lists, and sites on the WWW. It is considered good practice for system administrators to periodically utilize host based auditing tool and help ensure that their systems can withstand such attacks.

2.1.1.4   Monitoring Tools

On the surface, a monitoring tool is a program that simply captures or logs information available to itself. In expert hands, it is analyzed later to uncover weaknesses and vulnerabilities in the system. A monitoring tool resembles the technical read-out information that was stored in R2D2 robot in the well-known movie, Star Wars: Part I, analyzing which scientists considered rebels by the Empire were able to uncover a weakness in the Death Star superweapon and successfully destroy it. Monitoring tools come in two forms, sniffers and snoopers.

- A "sniffer" program focuses on the information flowing back and forth between computer systems on a network, local or otherwise. Commonly referred to as network traffic, the information contains user name-password pairs, authentication related data, and other system details that may be exploited by an intruder. For performance reasons, most local systems choose not to encrypt the data flowing

on a local computer network. A hacker with physical access to a network can "plug in" a sniffer and log any length of the network traffic. To counteract the deployment and abuse of sniffers, all transmissible data are first encrypted at the host-network interface and then launched on the network. The most commonly utilized technique is the Secure Socket Layer (SSL), designed and developed by Netscape in 1994 to achieve secure transactions through mutual authentication and data encryption. SSL may be deployed within a local network as well as the Internet. Of course, there is no guarantee that SSL will forever prevent the intruder from accessing the information, given that any encryption, in theory, can be broken. When making commercial purchases on the Internet on a web browser, the lock icon on the bottom of the browser frame generally indicates the use of SSL. Any Internet transaction that is transmitted in clear text and not protected through SSL may be immediately vulnerable.

• Unlike a sniffer, a "snooper" focuses on the information confined within a given computer system, namely a user's activities including terminal or terminal emulator sessions, process memory usage, and keystrokes. By installing a trojan or keylogger snooper on a given computer system, a victim's keystrokes and mouse clicks are captured in their native form and stored within the snooper. Normally, the information is retrieved by the attacker over the communications channel and analyzed to uncover vulnerabilities. Even where the communications channel is encrypted, the snooper-logged information is merely encrypted at the host-network interface, i.e., at the boundary of the computer system and the outgoing network link. The encrypted packet is then transmitted to the attacker's computer system via the network link, where the corresponding host-network interface automatically decrypts the encrypted packet and presents it to the intruder.

2.1.1.5 Stealth and Backdoor Tools

Stealth tools comprise a collection of programs and techniques that permit an unauthorized user to alter system logs and eliminate all records of unauthorized entry and activities prior to exiting the system. A stealth tool can also deliberately preempt the system from recording any of the user's activities, while in operation, thus rendering the attacker invisible. Stealth toolkits often include "backdoor" programs, which consist of modified, drop-in, binary code replacements of critical sections of the system that provide authentication and system reporting services. Backdoor programs offer the following capabilities:

• Provide continued, unlogged use of the system when activated; the activation mechanism is often an encrypted password compiled into the program.
• Hide suspicious processes and files from users and system administrators.
• Report false system status to users and system administrators.
• Report false checksums for the modified programs, thereby defeating any alarms and watchdog devices.

A few of the well known backdoor/trojan packages include the following:

- The Back Orifice (BO/BO2K) server package was designed and developed as a remote administration tool by the hacker group, known as the cult of the Dead cow (cDc). When installed on a compromised Microsoft Windows NT/2000 machine, the BO/BO2K software package provides complete control over the host, including the ability to monitor and record keystrokes and mouse clicks and the exclusion and removal of installed application programs. While unauthorized application programs may be installed, new data files may be created and prior files modified or deleted, and hard drives can be reconfigured and reformatted. The presence of a BO/BO2K package on a machine must be viewed as a certain sign of compromise and warrants investigation. The software package is available at the hacker group's website: http://www.cultdeadcow.com. Available anti-virus software products may be utilized to detect and remove BO/BO2K packages. For further details, the reader is referred to CERT Vulnerability Note VN–98.07.
- Freely available on the WWW, the Netbus software package was developed by Carl-Frederik Neikter. It is a trojan, similar to BO/BO2K. Current versions of anti-virus software packages are able to detect NetBus installations.
- SubSeven: The SubSeven software package was designed as an improvement over NetBus. It is designed to serve as a slave to the remote master-attacker. When the software package is installed on a system, it quietly listens for connection requests from the remote master-attacker. Analysis of firewall logs generally reveal numerous attempts of automated scans to locate the presence of SubSeven software packages with default configurations in the target computer system. Most versions of anti-viral packages can detect SubSeven installations.
- As a key operating system of choice of the Computer Science community and given that its design reflects systematic protection mechanisms, Unix has been the focus of the hacker community for a long time. As a response to the challenge offered by Unix, the hacker community has develop a custom rootkit for every flavor of Unix. The term rootkit is derived from the ultimate user account on a Unix system, called the root, with the highest privileges, also referred to as root privileges. The rootkit is a set of tools and trojan devices that are configured and installed on compromised Unix systems. While the tools are designed to sanitize the audit logs, i.e., remove all evidence of intrusion, the trojan devices are altered systems utility programs, which permit the intruder a backdoor entry into the system through a special password. All conventional logging and authentication checks are relaxed and the intruder acquires the highest level system administrator privileges. Many freely available software packages can detect and report the presence of rootkits.

2.1.1.6 Auto-Rooters

Auto-Rooters are attack toolkits with extremely high levels of automation and significant attack ferocity against specific weaknesses of computer systems.

Auto-Rooters are designed by expert hackers for the current generation of point-and-click computer users so that maximal damage is caused, worldwide. Auto-Rooters are an emerging class of attack toolkits and proliferating rapidly. When a vulnerability is first discovered and reported, even in a mainstream scientific forum, a window of opportunity arises for the hacker community to attack a very large number of computer systems that have been just rendered vulnerable. Fully aware that the defenders will soon create a program fragment to temporarily patch the defect, the hacker community races to create and disseminate packages, called Auto-Rooters that represent the ultimate in point-and-click hacking. By design, Auto-Rooters can be utilized by the most unsophisticated users, with little to no knowledge of systems and networks, to attack both local and remote computer systems across the world. Thus, these toolkits are potentially very dangerous. When a basic Auto-Rooter attack is launched against a network, every address within the specified network range is attacked with a specific remote exploit and the results of the attacks are logged. After the package completes phase I execution, the attacker has a report of the machines that are vulnerable. In phase II, automated attacks are launched on the machines identified as vulnerable, thereby enabling the perpetrator to compromise the largest number of machines with the least effort. Each of the compromised machines can then be altered and converted into elements of a distributed denial of service flood-net attack, patched with backdoors, or used as platforms to launch attacks against other internal or external computer systems. Auto-rooter activity is detectable and betrays a behavior profile similar to that of a network worm. Many hosts are scanned and each of the hosts will have the same service accessed with the same data parameters, usually consisting of the remote-exploit payload that will attempt to breach the security of the host. Properly configured firewalls, both internal and external, and intrusion detection systems can detect and mitigate the impact of an Auto-Rooter attack.

2.1.2 Deployment of Toolkits for Unauthorized Intrusions

Armed with a collection of "exploit scripts" that they may have developed, hackers generally attack computer systems on the network, driven primarily by intellectual challenge. At first, the attacks are tested on easy targets but then expert hackers move onto other computer systems that are difficult to break into and therefore offer greater challenge. There is mounting evidence, however, of increasingly focused attacks on the networks of specific organizations for the purpose of fraud, espionage, and monetary gain. As in many scenarios, the initial attacks stem from intellectual challenge and are refreshingly clever; however, they are quickly followed by mundane attacks launched by mediocre individuals whose goals are wanton damage and self-serving exploitation.

A detailed study of attacks launched against thousands of machines reveal two important insights. First, attacks seem to be launched at three levels, namely, (A) blind remote attack, (B) user-level attack, and (C) physical attack. Second,

unbeknownst to the community at large, hackers appear to follow a systematic methodology in conceiving and planning their attacks. The methodology is presented in Sect. 2.1.2.1.

(A) Under a blind remote attack, the perpetrator initiates an attempt to remotely penetrate into a computer system or network, armed solely with the network address in either numeric or text form. The attacker is blind in that he neither possesses valid account information nor access to the target. Blind remote attack represents the "classic" attack scenario, where an unknown attacker attempts to access a computer network illegally. Most penetration tests carried out by security consultants include, at the very least, a blind remote attack. The intruder will first deploy scanners and other methods to acquire security-related information on the target system. Following analysis of the data returned by the scanners, the intruder will choose the most appropriate remote exploit from the toolkit arsenal and launch it at the target to gain access to the system.

(B) A user-level attack represents a penetration attempt into a computer system on which the intruder already has a user-level account with unprivileged access, for the purpose of acquiring privileged access. The account exploited may have been legitimately acquired as a customer or employee of the organization; or otherwise acquired through "sniffed" passwords, traded accounts, "shoulder surfing," blind remote attack, cracked passwords, social engineering, or default user accounts. A majority of the losses in the financial industry attributed to breaches in network security stem from insider attacks, where a legitimate user attacks the network from within. In phase I of the attack, the perpetrator launches a COPS or TIGER scanner locally to detect and report common security vulnerabilities in the computer system and users. In phase II, the intruder will analyze the scanner data, identify the most effective local exploit from the toolkit, and launch the attack. If and where successful, the intruder will have gained privileged access to the computer system. In phase III, the intruder will intercept sensitive system data and network traffic to acquire unauthorized access into other machines on the network.

(C) Under physical attack, the individual with physical access to computers and the network equipment circumvents the traditional authentication, namely, login username and password, and plugs in attack computers and hardware scanners directly into the appropriate ports of the computer server and network equipment, thereby intercepting network traffic. As with the other paradigms, analysis of the traffic is likely to yield knowledge of the vulnerabilities which may be subsequently exploited to gain unauthorized access. Physical access greatly facilitates intrusions. It is common practice for most computer users to leave their desks with the computers running and active logged in sessions. An intruder can extract valuable information that will be vital to break into the network. Where the target computer system has no active sessions, the intruder can shut down and reboot the system, thereby gaining administrative privileges on the system for certain system configurations. This renders other systems on the network vulnerable to attack.

2.1.2.1   A Methodology of Attack

A comprehensive analysis of the nature of attacks and evidence from data left behind in compromised machines reveals the following insight. Virtually all intruders' attacks, exceptions notwithstanding, are well considered and systematically organized along seven stages. These include:

1. Reconnaissance: gather information about the target system or network.
2. Probe and attack: probe the system for weaknesses and deploy appropriate tools from the toolkit.
3. Toehold: exploit security weakness and gain entry into the system.
4. Advancement: advance from unprivileged account status to a privileged account.
5. Stealth: hide all tracks of intrusion and install a backdoor.
6. Listening post: establish a listening post to monitor if prior intrusion had been detected.
7. Takeover: expand control from a single host to other hosts on the network.

For the blind remote attack scenario (level A), the intruder would first attempt to gather information about the targeted system, as in stage 1. Utilizing this information, the intruder would apply the remote exploit tools and techniques in an attempt to gain a toehold into the network. This would represent stage 2. Where the penetration attempt is successful and the toehold is that of a privileged account, stage 3 is complete. Should the toehold be limited to an unprivileged account, the intruder would escalate to stage 4 and seek privileged access using an effective local exploit. Next, the intruder can immediately begin covering his or her tracks and establish a listening post, implying completion of stages 5 through 7.

For the user-level attack scenario (level B), the intruder has already achieved a toehold into the target network, implying stage 3. The toehold may have been attained either through username and password guessing or cracking the password file that had been retrieved from the remote system. Generally, once a password file has been stolen there is strong likelihood that the intruder will correctly guess 25% of the passwords. To escalate into stage 4, the intruder obtains information about the local system, as in stage 1, and then applies local exploits, as in stage 2. Eventually, privileged access is acquired, implying stage 4 has been achieved. Next, the intruder hides all visible evidence of intrusion, installs a series of backdoors to ensure future unauthorized access into the target, and begins the takeover process. Stages 5 through 7 are completed.

For the physical attack scenario (level C), the intruder may either reboot the system to gain administrator privileges; identify an active session in the unprotected physical computer and follow up with user-level attack scenario; or physically plug in computers and hardware scanners to download network traffic, analyze sensitive information, and achieve stages 1 through 7 with relative ease. Of great importance is that owners must provide adequate physical protection to their computer systems and network equipment.

2.1.2.2 An Illustrative Example of a Targeted Attack

In this section, we will illustrate the anatomy of an attack on a specific target. Consider a hypothetical company, XYZ Corporation, and that an unknown intruder has decided, for unknown reasons, to attack the computer systems and networks of XYZ. Clearly, the only information available to the intruder at the start of the episode is the public name of the corporation.

Under the reconnaissance step, the intruder begins to search the Internet and WWW for all references to the target corporation, including Internet connections, Web sites, FTP sites, and electronic mail service. Assume that the search yields a domain name, xyzcor.com, registered to XYZ Corporation, without any loss in generality. Armed with the domain name, the intruder then begins to search for more information through a number of different methods. One possibility is to exploit the domain information groper utility program, called "dig," developed by Steve Hotz. To learn more about other machines within the domain, the intruder attempts a "zone transfer" on the domain's name servers. Assuming that the effort is successful, the intruder extracts from the target system a list of host names and their network addresses. In the next step, the intruder begins to compile information about the users on the system. Two excellent sources include the newsgroups and news hierarchy in the domain and the WWW. Gradually, a list of users on the system is compiled. This list is very important to the intruder for it has the potential to reveal many username and password combinations and possibly the domain's policy of determining usernames. For example, if a search yields the line, "From: bobr@host.xyzcorp.com (Bob Reilly)," from a news posting, the intruder can now attempt to break into the account for username, bobr, by repeatedly guessing passwords. If a subsequent search yields the line, "From: sarahg@hostb.xyzcorp.com (Sarah Gregory)," there is a very good chance that the usernames for the entire system have been determined based on a uniform policy of concatenating the individual's first name and the first letter of his or her last name. Next, the intruder can either begin to guess additional usernames and passwords, or search for a given username on chat channels [Internet Relay Chat (IRC) Web Chat] seeking the user's personal information, including full name, address, phone number, etc. Armed with adequate personal information, the intruder might then contact the user either by phone, electronic mail, or chat and acquire account information through persuasion or social engineering. The intruder may even lure the user into inadvertently running a hostile or "Trojan horse" program, the intent being to capture account information and return it to the intruder. At the conclusion of the reconnaissance phase, the intruder may have acquired the following:

- Host name(s)
- Host address(es)
- Host owner
- Host machine type
- Host operating system
- Network owner
- Other hosts in the network

- Network configuration
- Other hosts trusted by the network
- Hosts outside the network
- List of users
- Username assignment policy

In the probe and attack step, the intruder begins to examine the perimeter of the system's security for potential weaknesses. This step is the most heavily automated portion of the penetration cycle. Toolkits left behind and recovered from compromised sites always reveal the presence of some type of scanner that enables the intruder to conduct security surveys on the entire network. Well known scanners include SATAN in the public domain, discussed earlier in this chapter, and commercial scanners such as ISS. This step also represents the most risk for the intruder in that scans and probes are most likely to be detected and logged by intrusion detection systems, where installed, which will promptly alert security-conscious system administrators and users. To uncover vulnerabilities, probe programs determine the remote services provided by the hosts. A freely available, public domain tool, "strobe," allows an intruder to scan a host or range of hosts to generate a list of services offered by each one. Thus, by letting loose strobe on the "host.xyzcorp.com," a list of services is compiled. Assume that the services of interest include FTP, SMTP (for e-mail), finger, WWW, printer, and xterm, the X-Window System server. These generally come with well known vulnerabilities. The intruder selects from the toolkit the most effective remote exploits against these services, one by one, and launches them, until the vulnerabilities are discovered. The FTP server is first checked for known vulnerabilities and configuration errors. Second, the sendmail server, SMTP, is probed to yield the software name and version number, thereby assisting the intruder to select the most effective exploit. Should bogus or no information be retrieved from the server's banner, the intruder's task is complicated and, worse, the likelihood that the intruder may be discovered is increased. Assume that all of the services, except the WWW server, successfully resist the probe. The WWW server on host.xyzcorp.com offers the "phf" service, which has a known vulnerability, and the intruder possesses an effective remote exploit. A hostile command is executed on the server, yielding an X Window System terminal emulation on the intruder's display. A toehold into the target network has just been achieved.

In the toehold step, the intruder has already gained unauthorized entry into the system. Should the user identification (UID) of the X Window System terminal indicate "root," the intruder jumps directly onto the stealth step, skipping the advancement step. Where the UID is that for an unprivileged user, the intruder will attempt to migrate to a privileged or administrative account.

In the advancement step, the intruder uses the information about the host, the operating system, and the services provided to search the toolkit for the most effective local exploit. Assume that the intruder has obtained a local display running a shell on the remote server with the UID, "www." The intruder will deploy the local scanning tools – COPS or TIGER, to search and report configuration errors and other known vulnerabilities, and then apply local exploits from the toolkit. If the local scan using COPS reveals the host to be an AIX* 3.2 (Advanced Interactive

Executive) machine, vulnerable to the "tprof" exploit, the intruder can successfully advance from UID "www" to UID "root," the privileged account. At the highest privileged level, the intruder is in full control of the target computer system. On most systems, any local file may be accessed, modified, and deleted. A malicious intruder may look around for any interesting data and delete the entire file system. Most intruders, however, retain their access to the compromised system, and move to the subsequent step.

In the stealth step, the intruder is the root, with complete access to all of the files on the local system. To erase all evidence of unauthorized entry and preempt detection, the intruder will edit the files containing the log entries. Given that intrusion had been gained through an exploit on the WWW, the intruder will check the WWW server access log for records of previous intrusions and delete all traces of illicit activity. By replacing the system's not-so-easy-to-read binary code with modified versions that hide process, file, and network connection information, effectively, all incriminating traces are removed.

In the listening post step, the intruder ensures continued, unlogged, and undetected access to the compromised system at anytime. Using an appropriate "rootkit" package, the intruder "patches" the system's binary files to serve three key objectives. The first is to ensure that any future activity will be never logged. The patched binary files have been deliberately designed to report false information on files, processes, and network interface status in response to the administrator's queries. The second objective is to facilitate continued and unlogged access to the system through a number of backdoors. The third objective is to establish a listening post for the network, for which a sniffer program is installed in the target. In the event the target computer system's network interface supports "promiscuous mode," the sniffer program allows the intruder's privileged account to intercept and record all network traffic. Where the "promiscuous mode" is unsupported, the intruder is limited to intercepting traffic for users on the local system, one at a time. Network traffic carries sensitive information, including e-mails and username-password combinations for other systems and networks. By recording and subsequently analyzing them, the intruder can easily widen the scope of control.

In the final takeover step, utilizing sniffed username-password combinations and a toolkit of local and remote exploits, the intruder can successfully extend attacks onto other hosts of the increasingly encompassing network. Starting with a single weakness in a single machine within an ever increasing hierarchy of interconnected computer networks, the intruder exercises control over a vast array of computer systems and networks. As more and more hosts fall victim, the intruder's base of platforms from which to launch new attacks keeps growing unabated. For each new compromised host, the installed backdoor programs ensure detection preemption and continued, unlogged, privileged access to the hosts. The username-password combinations obtained from the listening posts provide ammunition to continue acquiring future toeholds and root compromises. In theory, takeover step may continue indefinitely, ad infinitum, across any and every computer system that is linked to any portion of the compromised network.

The most important lesson in securing computer systems and networks is the following. While no system is totally secure [5], the application of basic precautions, described in this chapter, can go a long way to substantially reduce the possibility of a successful and damaging attack on an organization's vital assets. It has been common practice for a long time to first develop a system and, if and when it begins to function successfully, incorporate security precautions. Thus, security has been an afterthought, a reactionary measure, which is never strong and robust. Security concerns need to be addressed throughout the development and maintenance phases of every project. Organizations, worldwide, have begun to address the security issue seriously.

2.2 Denial of Service

Denial of service (DoS) attacks are characterized by deliberate and carefully considered efforts to limit or prevent legitimate users from accessing network resources. Most practical DoS attacks involve multiple target machines and multiple machines from which attacks are launched, implying distributed denial of service attacks, labeled DDoS attacks. Given the ubiquity of computers and their role in critical areas including control of the power grid, e-commerce, etc., DDoS attacks are quickly becoming the most serious problem on the Internet. Sustained DDoS attacks on a corporation's computer systems and network can cause significant financial loss and other damages to the target. As an analogy, a DoS attack on a telephone unit may work as follows. Assume that a subscriber, X, wishes to deny another subscriber, say Y, the ability to receive calls from the outside. Assume also that Y does not have call waiting or other sophisticated services. X would dial the number for Y on his or her own telephone and hang-up just as Y's telephone is about to start ringing. X would repeat this action immediately and continue the process indefinitely. The local telephone switching station that is directly connected to Y would be busy oscillating between ringing Y's telephone and cutting it off, implying that it would be difficult for some other subscriber to get connected to Y.

Although DDoS attacks have been known for a while, in February 2000, the first of a series of large-scale, coordinated DoS attacks were launched against key popular websites on the Internet, including Buy.com, CNN, Datek online trader, E*trade trader, and eBay. Even the DOJ and FBI websites were attacked and rendered unreachable by users, worldwide. Hundreds, possibly thousands of compromised machines were directed to attack target systems in a well-coordinated manner. The massive and precise coordination rendered the attacks exceptionally successful in that legitimate users were precluded from accessing the systems within a wide geographical area. For the duration of the attacks, subscribers were unable to access new information; customers were unable to places orders or execute bids at auctions; and financial traders were unable to access their accounts and place securities orders. The attacks lasted several hours and the technical staff of the target organizations could not track the sources of the attacks for the simple reason that an isolated

IP packet cannot be traced effectively. The attackers had utilized a packet spoofing technique to obscure the true source(s) of the attack. With source addresses of the attacking IP packets spoofed, attempts to identify abusive packets and filter them out fails. The attacks were a major media event and most of the mainstream television and media covered the attacks and the impact on the victims, namely, the Internet users. Conceptually, under DDoS attacks, a single attack computer can target multiple hosts simultaneously or in accordance to a specific schedule; multiple attack computers may be directed to attack a single host; or multiple attack computers can attack multiple hosts in a coordinated manner, where the coordination may be based either on timing or causality. This provides perpetrators the ability to control the granularity of attack over a wide dynamic range, i.e., they can attack a single host with surgical precision or large group of hosts and cut off from the rest of the world.

Historically, DoS attacks were originally developed for use on Internet Relay Chat (IRC). They were the result of on-line squabbles within the chatrooms. Rival IRC users would launch DoS attacks against each other in attempts to knock other users out of the chatroom. A rival would attempt to either overload the network connection or the computer system of the target with bogus packets. The sheer volume of the packets would consume all of the available bandwidth and clog the network connection. No data would either come in or get out of the target system, effectively knocking the opponent offline. The IRC user with the most network bandwidth at his or her disposal would emerge as the winner of these online virtual skirmishes.

Another form of DDoS attack targets vulnerabilities in the client and server software. Server software is the program that provides service to clients on the network. On the WWW for example, a user will use a local web browser program, or http client, to connect to computers running the web server software. The user will access and download HTML formatted code and resources. The web browser will the interpret the downloaded code and resources and display the results to user's computer screen. Under attack, the perpetrator sends deliberately malformed data to the server software program, which exploits a vulnerability in the application, causing the server software program to lockup, consume large amounts of CPU, memory, or disk space, or simply fail and exit. In the IRC community, such DoS attacks on IRC servers would enable the attacker to acquire special operator privileges or 'ops' in the chatrooms, granting complete control over the channel.

2.2.1 Different Manifestations of DDoS Attacks

- A mail bomb DoS attack is a technique wherein an user attempts to overload the e-mail processing capabilities of a specific user or network. It is the oldest and most crude of all DoS attacks. The attacker constructs a specific e-mail and sends it repeatedly to the same user or multiple users at a particular site. A variation of the attack may involve attaching with the e-mail a large file containing image, audio, video, or random garbage data. Under normal operation, an e-mail consumes a limited amount of available resources, including bandwidth,

memory, and disk space. Under attack, all of the available resources are virtually depleted by the sheer number of bogus e-mails, thereby crashing the target system or impacting it severely.

- In 1995, the popular underground magazine, Phrack, documented the SYN flood vulnerability and even provided the source code for launching an attack. A SYN attack focuses on the TCP protocol, which governs the establishment of reliable and full-duplex connections between clients and servers across IP networks through a 3-way handshake. First, the client's computer initiating the connection sends a connection request message, SYN, to the server computer. The server responds by sending back to the client a SYN/ACK acknowledgment message. This signifies that the server is ready to communicate with the client. Upon receipt of the SYN/ACK, the client knows with certainty that it has gotten the server's attention. Of course, the server needs to know with certainty that the client has received its SYN/ACK, which is where the third and final step of the 3-way handshake becomes necessary. The client transmits a message to the server acknowledging the SYN/ACK it has received. Upon receipt, the server and the client are now both certain that they are in communication with each other. The connection is deemed to have been established. The SYN attack is conceived based on a known fact that the server has already allocated a part of its resources when it sends the SYN/ACK to the client and is waiting for the final acknowledgment from the client in step 3. The expectation is that the acknowledgment will arrive soon, after which the connection will be established and communication will ensue. Initial implementations of the server allowed for a limited number, namely 5, of outstanding, half-open connections, implying that further requests for connections from other clients would be ignored. Clearly, in attack mode, the perpetrator would strategically send five bogus requests to the server, intercept the SYN/ACK responses from the server, and deliberately refrain from sending the final acknowledgments to the server. As a result, the server is inhibited from accepting additional requests from clients and denying service. Where revised implementations of the server program attempt to time out and retire the half-open connections, the attacker would continuously send connection request messages, under flood mode, from the clients to the server, eventually locking it out. To render it robust, the server program was revised to detect flooding activity by uncovering the source address(es) of the connection requests and applying appropriate filters to block them. The Octopus toolkit provided the resources for launching the SYN flood attack.
- Under a smurf DoS attack, an attack host computer sends out a ICMP request targeted at the network's broadcast address but deliberately inserts the victim's IP address in the return address field. The machines on the network, possibly numbering in the hundreds, will respond with a ICMP message directed at the victim, who is quickly inundated. Smurf toolkits can even amplify the aggravation of the attack by sending the ICMP request to multiple broadcast addresses, triggering thousands of ICMP responses to the victim. For obvious reasons, smurf DoS attack is also known as traffic amplifier attack. Mitigation techniques focus on proper configuration of the border routers.

- The Teardrop or Bonk DoS attack was first introduced in 1997 and it targets the TCP/IP protocol. Some implementations of the TCP/IP driver lack the ability to properly handle overlapping IP datagram fragments, thereby causing the system to crash. The attack consists in transmitting malformed sequence of IP datagrams to the target host. Patches for the weaknesses are available from appropriate vendors.
- Under the Land DoS attack, a malformed TCP SYN packet is synthesized, where the source address and port are deliberately set identical to that of the victim machine. Upon receipt of this malformed packet, the target will fault and hang since it had not sent a connection request message.
- In the ping of death (POD) DoS attack, first reported in 1997, the attacker causes an oversized ICMP packet to be transmitted to the target machine. Unable to process the oversized packet, the victim locks up. The weakness has since been eliminated but older versions remain vulnerable.
- The Trin00/TFN/Shaft/Stacheldraht attack was one of the earliest DDoS attack toolkit introduced in 1999. The toolkit contains both server and client components. In the pre-attack phase, the client component is installed on the attacker's computer or a specific compromised client machine, while the server components are configured and installed on all compromised server systems. In the attack phase, the single compromised client machine compels all of the compromised servers in the flood-net to launch DoS attack at a single host or the network.

2.2.2 Toolkits for DDoS

A typical DDoS attack toolkit consists of two basic software packages, namely, master and slave component packages. The slave package is installed on each of the computers that have already been compromised, while the master package is installed on the attacking computer. Perpetrators are known to have employed different techniques to install the slave packages. Under approach 1, the attacker first scans and locates a vulnerable host, then compromises it, and subsequently installs the slave package. Hackers will often utilize auto-rooter toolkits to seek out and attack large numbers of networked computer systems, the goal being to create a large network of subverted computers, collectively known as a "bot-net". The larger the bot-net, the greater the resource base upon which the attack is launched. Under approach 2, the slave package is distributed as a payload of computer viruses and network worms. The compromised machines are termed "zombies." There have been reported instances where a large bot-net had generated over 500 Mb/s of network traffic, which is capable of saturating over 300 T-1 links, where a T-1 link is rated at 1.5 Mb/s. A typical countermeasure against a DDoS attack consists in first identifying, where possible, the source address of the computers that are launching and directing the bogus messages, and then filtering them out of the network. A limitation of this technique stems from the fact that the bogus network traffic being generated by the zombies contain false, spoofed source addresses which are

created through pseudo-random number sequence generators. If an attack involves UDP packets, the network administrator can reconfigure the border routers to drop all UDP traffic. If this fails, the network administrator may then contact the organization's Internet Service Provider (ISP), who can then attempt to locate the source of the attack and install appropriate filters. Where the ISP is unable to mitigate the attack, it may then contact the service provider at the next higher level, and so on until the source of the attack is located and blocked. Given that the process is manual, clearly, the effects of a DDoS attack can cripple a network for a very long duration, relative to the network speed.

References

[1] Boulanger, A. (1998). Catapults and grappling hooks: the tools and techniques of information warfare. *IBM Systems Journal, 37*(1), 106–114.
[2] Boulanger, A. (1997, Summer). Cyber-crackers: computer fraud and vulnerabilities, invited article. *The Journal of Public Inquiry: President's Council on Integrity and Efficiency*.
[3] Weaver, R., & Stroz, E. (1998). *Counterfeit chip detection*. Internal Report for US FBI/US Secret Service.
[4] Boulanger, A. (2006). *Electronic identification as economic commodities in the black market*. IBM Technical Report.
[5] Boulanger, A. (2005). Open-source versus proprietary software: Is one more reliable and secure than the other? *IBM Systems Journal, 44*(2), 239–248.