National Computer Board

Computer Emergency Response Team of Mauritius (CERT-MU)

Guide on Protection Against

National Computer Board |CERT-MU www.cert-mu.org.mu

Contents

1.0 An Introduction to Hacking

Hacking basically means gaining access to a computer file or network without authorisation or through illegal means.

1.1 A Little Bit of History

The concept of Hacking has been around since the development of the first electronic computers. The first computer hackers emerged at Massachusetts Institute of Technology (MIT). They borrowed their name from a term to describe members of a model train group at the school who "hack" the electric trains, tracks, and switches to make them perform faster and differently. A few of the members transferred their curiosity and rigging skills to the new mainframe computing systems being studied and developed on campus.

Phone hackers (phreaks) broke into regional and international phone networks to make free calls. One phreak, John Draper (also known as "Cap'n Crunch"), learnt that a toy whistle that came freely with Cap'n Crunch cereal generated a 2600-hertz signal, similar to AT&T's long-distance switching system. Draper built a "blue box" and made use of the whistle which sounded like a phone receiver that allowed phreaks to make free calls.

Prior to Usenet newsgroups and e-mail, the “Sherwood Forest” and “Catch-22” groups became the venue of choice for phreaks and hackers to gossip, trade tips, and share stolen computer passwords or credit card numbers. Hacking groups then began to form. Among the first were “Legion of Doom” in the United States, and “Chaos Computer Club” in Germany.

1.2 Who is a hacker?

A 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 derogatory meaning of cracker and is used to denote someone who gains unauthorised access to computers & networks. An intruder is an entity that gains or attempts to gain access to a system or system resource without having authorisation 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.

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 compromising the “Bazee.com” website for a day.

Hackers may bombard a website with innumerable visitors or queries so as to make it unavailable 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 Denial of Service (DoS) attack. If a large number of computers is involved, it is Distributed DoS (DDoS).

1.2.1 Script-kiddies

Scanning tools that look for well-known vulnerabilities are generally written by advanced hackers who make it available over the Internet. Less experienced hackers, commonly known as "script kiddies," then run the scanning tools 24 x 7, scanning large numbers of systems to find vulnerabilities. They typically run the tools against the namespaces 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 and to gain access to systems.

1.2.2 White Hats

White Hats are computer security experts who specialize in penetration testing and other methodologies to ensure that a company's information systems are secure. These IT security professionals rely on a constantly evolving arsenal of technology to counteract hackers.

1.2.3 Black Hats

These are the bad guys, who are typically referred to as just plain hackers. The term is often used specifically for hackers who break into networks or computers, or create computer viruses. Black hat hackers continue to technologically outpace white hats. They often manage to find the path of least resistance, whether due to human error or negligence, or with a new type of attack. Hacking purists often use the term “crackers” to refer to black hat hackers. Black hats’ motivation is generally to generate money.

1.2.4 Grey Hats

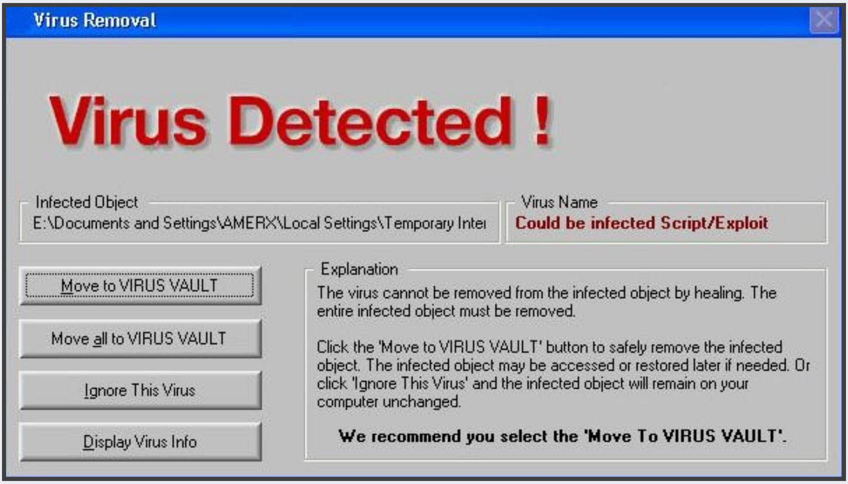
A grey hat refers to a skilled hacker whose activities fall somewhere between white and black hat hackers, on a variety of spectra. It may relate to illegal acts, though in goodwill. They usually do not hack for personal gain or have malicious intentions, but may be prepared to technically commit crimes during the course of their technological exploits in order to achieve better security.

2.0 Types 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.

2.1 Virus

A virus is a hidden, self-replicating section of computer software, usually malicious logic, that propagates by infecting itself; - 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.



Virus Removal

Virus Detected !

Infected Object
E:\Documents and Settings\AMERX\Local Settings\Temporary Inte

Virus Name
Could be infected Script/Exploit

Explanation
The virus cannot be removed from the infected object by healing. The entire infected object must be removed.

Click the 'Move to VIRUS VAULT' button to safely remove the infected object. The infected object may be accessed or restored later if needed. Or click 'Ignore This Virus' and the infected object will remain on your computer unchanged.

We recommend you select the 'Move To VIRUS VAULT'.

Figure 1: Virus detection

Example: 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.

2.2 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 bypass security mechanisms, sometimes by exploiting legitimate authorisations of a system entity that invokes the program. The idea of maliciously modifying a normal program in addition to its usual function and arranging for the victim to use the modified version is known as Trojan horse attack.

Example: An attacker gets the source code of an editor program, modifies it to steal someone's files (but still work perfectly as an editor). This is done by compiling the stolen source code and converting it into anothet version which works fine. However, the new code is meant to steal the legitimate user's file.

2.3 Worm

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

Example: An attacker discovers or uses bugs in an Operating System or Application that makes it possible to gain unauthorised 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 has access to.

2.4 Vulnerability

Vulnerability is a flaw or a weakness in a system's design, implementation, or operation and management that can 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 shortcuts for gaining unauthorised access to systems.

They may use their access for illegal or destructive purposes, or may simply be testing their own skills. 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 in 2003 proved this, despite that fact that a patch for was available six months earlier.

2.4.1 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 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.

2.4.2 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 organisations 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. Security sites, such as "SecurityFocus", CERTs, the SANS Institute, and many others, provide information about how to determine the vulnerabilities an operating system 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.

2.4.3 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 does not stop at the operating system.

Applications such as online banking, e-commerce sites, 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.

2.5 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 are examples of buffer overflow.

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.

2.6 Denial-of-Service

DoS attack disrupts or completely denies service to legitimate users, 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 - Transmission Control Protocol/Internet Protocol. There are a variety of DoS attacks which can be categorised into bandwidth consumption attacks, resource starvation attacks, routing/DNS attacks.

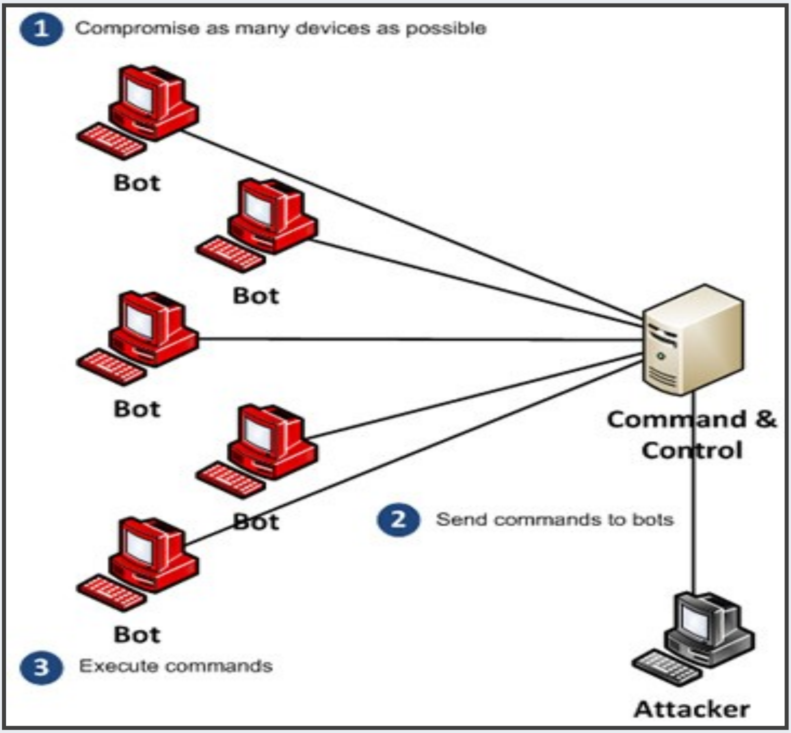


Figure 2: Distributed Denial-of-service (DDoS) attack

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.

3.0 Hacking Tools

Hackers make use of a variety of tools to attack systems, each having distinct capabilities. Most popular tools can be categorised into port scanners, vulnerability scanners, rootkits and sniffers, as shown in the diagram below:

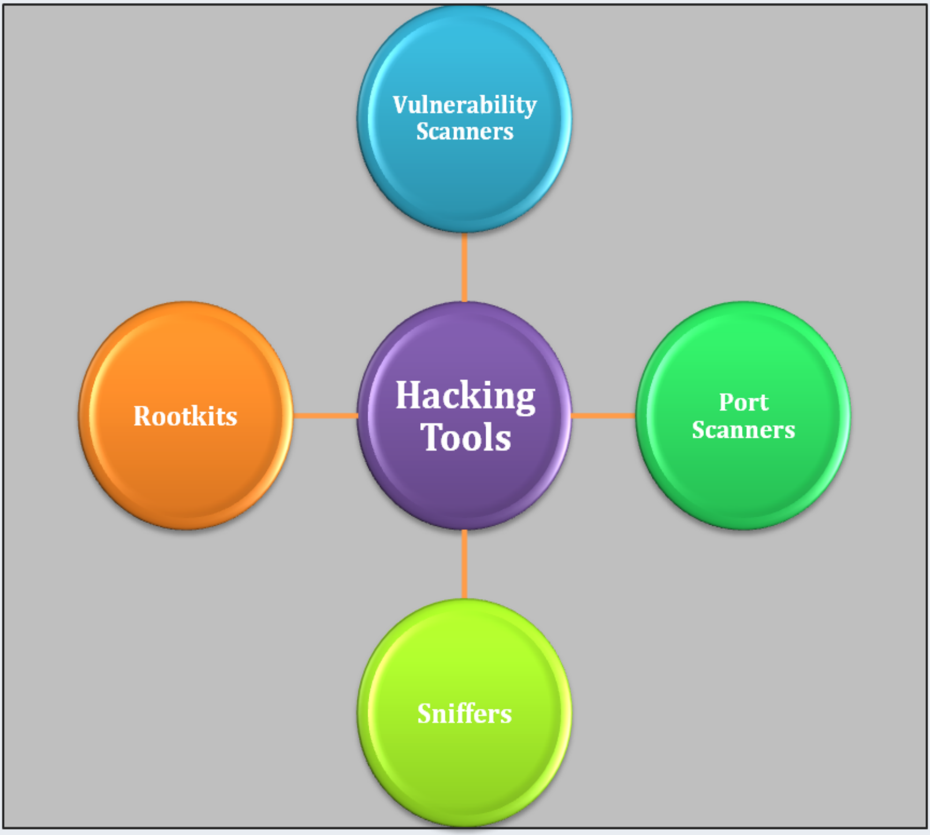


Figure 3: Hacking Tools

3.1 Port Scanners

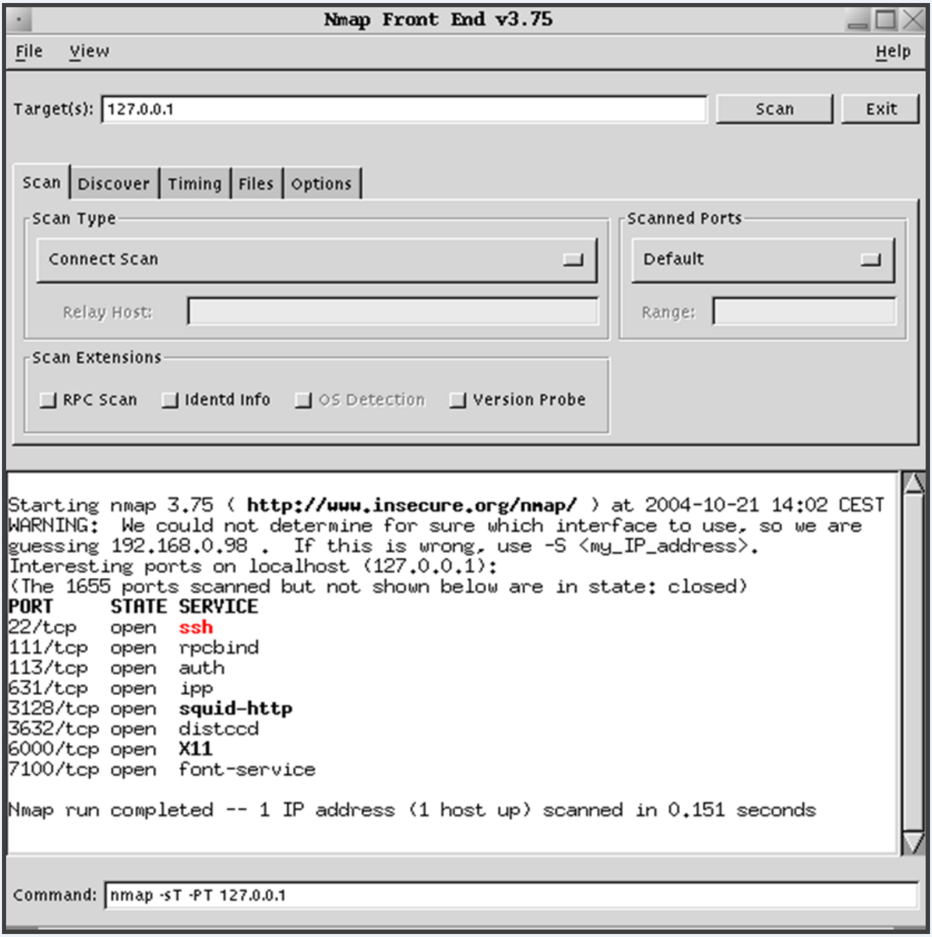
Port scanners are probably the most commonly used scanning tools on the Internet. These tools scan large IP spaces and report on the systems they encounter, the ports available, and other information, such as OS types. The most popular port scanner is Network Mapper (Nmap). The Nmap port scanner is described as follows on the Nmap web site. 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, the services (ports) that are offered, the operating system (and OS version) they are running, the type of packet filters/firewalls that are in use and other characteristics. Nmap runs on most types of computers, and both console and graphical versions are available. Nmap is a free software, available with full source code under the terms of the GNU General Public License.

Nmap is a helpful security tool because it allows us to determine which services are being offered by a system. Since Nmap is optimized to scan large IP ranges, it can be run against all IP addresses used by an organisation, or all cable modem IP addresses provided by an organisation. After using Nmap to find machines and identify their services, we 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.

An illustration of the Nmap port scanner is shown below. The IP address to be scanned is entered in the Target(s) field and the "Scan" button is normally used to begin a scan. Other options such as "Discover", "Timing", "Files" and the type of scan to be performed are also available to the user.



Target(s): 127.0.0.1

Scan Type: Connect Scan

Relay Host:

Scanned Ports: Default

Range:

Scan Extensions: ☐ RPC Scan ☐ Identd Info ☐ OS Detection ☐ Version Probe

```
Starting nmap 3.75 ( http://www.insecure.org/nmap/ ) at 2004-10-21 14:02 CEST
WARNING:  We could not determine for sure which interface to use, so we are
guessing 192.168.0.98 .  If this is wrong, use -S <my_IP_address>.
Interesting ports on localhost (127.0.0.1):
(The 1655 ports scanned but not shown below are in state: closed)
PORT      STATE SERVICE
22/tcp    open  ssh
111/tcp   open  rpcbind
113/tcp   open  auth
631/tcp   open  ipp
3128/tcp  open  squid-http
3632/tcp  open  distccd
6000/tcp  open  X11
7100/tcp  open  font-service

Nmap run completed -- 1 IP address (1 host up) scanned in 0.151 seconds
```

Command: nmap -sT -PT 127.0.0.1

Figure 4: An illustration of the Nmap Port Scanner

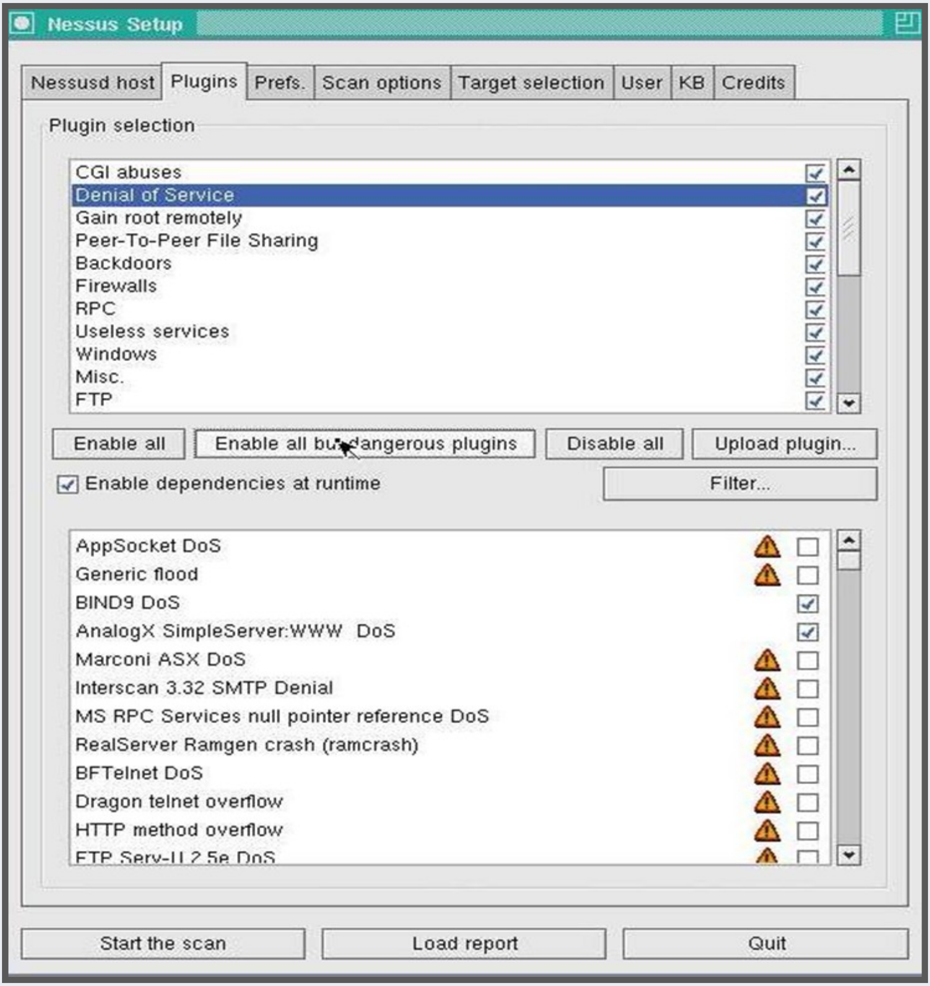
3.2 Vulnerability Scanners

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 scanners available is, Nessus. The Nessus vulnerability tool is described on the Nessus web as “a free, powerful, up-to-date and easy to use remote security scanner”. A security scanner is a software, which remotely audits a given network and determine whether crackers may break into it, or misuse it in some way.

Unlike many other security scanners, Nessus will not consider that a given service is running on a fixed port. For example, 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 attempt to exploit the vulnerability. Nessus is very fast, reliable and has a modular architecture that allows you to customise 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.

The diagram below illustrates the Nessus Vulnerability Scanner:



CGI abuses ☑
Denial of Service ☑
Gain root remotely ☑
Peer-To-Peer File Sharing ☑
Backdoors ☑
Firewalls ☑
RPC ☑
Useless services ☑
Windows ☑
Misc. ☑
FTP ☑

☑ Enable dependencies at runtime

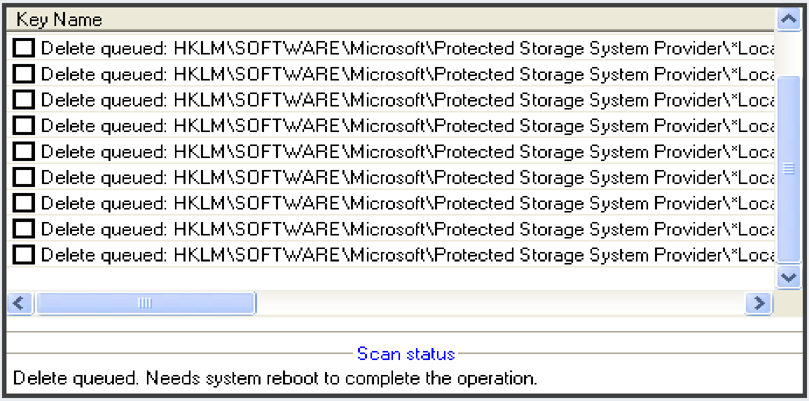
AppSocket DoS ⚠ □
Generic flood ⚠ □
BIND9 DoS ☑
AnalogX SimpleServer:WWW DoS ☑
Marconi ASX DoS ⚠ □
Interscan 3.32 SMTP Denial ⚠ □
MS RPC Services null pointer reference DoS ⚠ □
RealServer Ramgen crash (ramcrash) ⚠ □
BFTelnet DoS ⚠ □
Dragon telnet overflow ⚠ □
HTTP method overflow ⚠ □
FTP Serv-U 2.5e DoS ⚠ □

Figure 5: Nessus Vulnerability Scanner

3.3 Rootkits

The term rootkit describes a set of scripts and executables packaged together that allow intruders to hide any evidence about their 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 backdoor 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 well-known exploits to assist the attacker in the re-entry of a system.

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 is 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, etc...



Key Name

☐ Delete queued: HKLM\SOFTWARE\Microsoft\Protected Storage System Provider*Loca
☐ Delete queued: HKLM\SOFTWARE\Microsoft\Protected Storage System Provider*Loca
☐ Delete queued: HKLM\SOFTWARE\Microsoft\Protected Storage System Provider*Loca
☐ Delete queued: HKLM\SOFTWARE\Microsoft\Protected Storage System Provider*Loca
☐ Delete queued: HKLM\SOFTWARE\Microsoft\Protected Storage System Provider*Loca
☐ Delete queued: HKLM\SOFTWARE\Microsoft\Protected Storage System Provider*Loca
☐ Delete queued: HKLM\SOFTWARE\Microsoft\Protected Storage System Provider*Loca
☐ Delete queued: HKLM\SOFTWARE\Microsoft\Protected Storage System Provider*Loca
☐ Delete queued: HKLM\SOFTWARE\Microsoft\Protected Storage System Provider*Loca

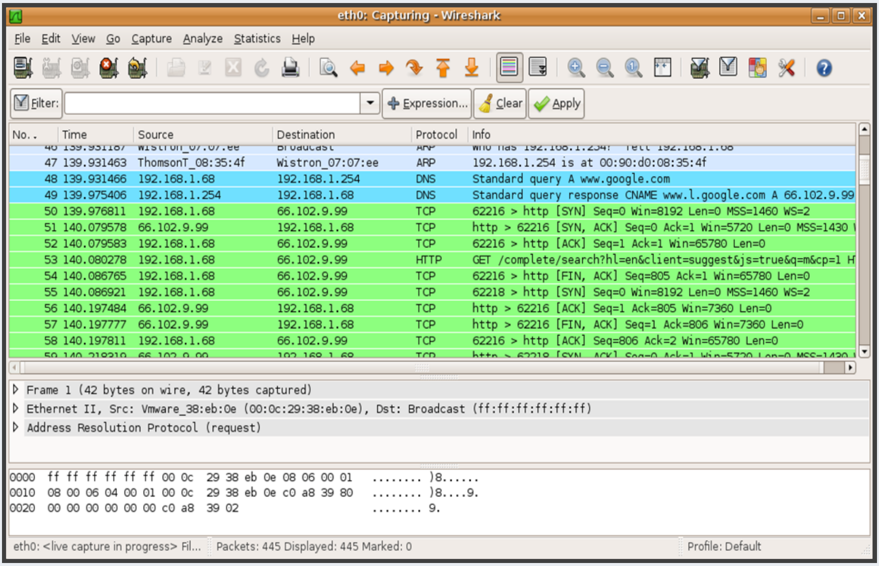
Scan status

Delete queued. Needs system reboot to complete the operation.

Figure 6: Example of Rootkit

3.4 Sniffers

Network sniffing, or simply "sniffing," means using a computer to read all network traffic, some of which may not be destined for that system. To perform sniffing, a network interface must support multiple layers so that it forwards to the application layer, all network traffic, irrespective of their destination. A good example of a sniffer is Wireshark, which is a network protocol analyzer. It can be used to capture and interactively browse the traffic running on a computer network. It has a rich and powerful feature and is the world's most popular tool of its kind. It runs on most computing platforms including Windows, OS X, Linux and UNIX. It is freely available as open source, and is released under the GNU General Public License Version 2.


eth0: Capturing - Wireshark
File Edit View Go Capture Analyze Statistics Help
Filter:
Expression...
Clear
Apply

No. .	Time	Source	Destination	Protocol	Info
46	139.931187	Wistron_07:07:ee	Broadcast	ARP	Who has 192.168.1.254? Tell 192.168.1.68
47	139.931463	ThomsonT_08:35:4f	Wistron_07:07:ee	ARP	192.168.1.254 is at 00:90:d0:08:35:4f
48	139.931466	192.168.1.68	192.168.1.254	DNS	Standard query A www.google.com
49	139.975406	192.168.1.254	192.168.1.68	DNS	Standard query response CNAME www.l.google.com A 66.102.9.99
50	139.976811	192.168.1.68	66.102.9.99	TCP	62216 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=2
51	140.079578	66.102.9.99	192.168.1.68	TCP	http > 62216 [SYN, ACK] Seq=0 Ack=1 Win=5720 Len=0 MSS=1430
52	140.079583	192.168.1.68	66.102.9.99	TCP	62216 > http [ACK] Seq=1 Ack=1 Win=65780 Len=0
53	140.080278	192.168.1.68	66.102.9.99	HTTP	GET /complete/search?hl=en&client=suggest&js=true&q=m&cp=1 H
54	140.086765	192.168.1.68	66.102.9.99	TCP	62216 > http [FIN, ACK] Seq=805 Ack=1 Win=65780 Len=0
55	140.086921	192.168.1.68	66.102.9.99	TCP	62218 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=2
56	140.197484	66.102.9.99	192.168.1.68	TCP	http > 62216 [ACK] Seq=1 Ack=805 Win=7360 Len=0
57	140.197777	66.102.9.99	192.168.1.68	TCP	http > 62216 [FIN, ACK] Seq=1 Ack=806 Win=7360 Len=0
58	140.197811	192.168.1.68	66.102.9.99	TCP	62216 > http [ACK] Seq=806 Ack=2 Win=65780 Len=0
59	140.218319	66.102.9.99	192.168.1.68	TCP	http > 62218 [SYN, ACK] Seq=0 Ack=1 Win=5720 Len=0 MSS=1430

```
▷ Frame 1 (42 bytes on wire, 42 bytes captured)
▷ Ethernet II, Src: Vmware_38:eb:0e (00:0c:29:38:eb:0e), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
▷ Address Resolution Protocol (request)
```

```
0000  ff ff ff ff ff ff 00 0c  29 38 eb 0e 08 06 00 01   ........ )8......
0010  08 00 06 04 00 01 00 0c  29 38 eb 0e c0 a8 39 80   ........ )8....9.
0020  00 00 00 00 00 00 c0 a8  39 02                     ........ 9.
```

eth0: <live capture in progress> Fil... Packets: 445 Displayed: 445 Marked: 0 Profile: Default

Figure 7: Capturing page of the Wireshark protocol analyzer

4.0 Protecting Against Hacking

The various ways of preventing attacks from hacking are described in the following sub-sections. You can go through these to have an understanding of how you can protect yourselves from the bad world.

4.1 Preventing vulnerabilities by educating the Security Administrator

The education of Security Administrators goes beyond learning about security software and applying the software to one's network. Without understanding what they are protecting their network from and what the attacker can do to their network, the network they protect is vulnerable. Since the majority of attacks come from the script kiddie, that should be an area of high concern for Security Administrators.

The common black hat has access to hundreds of scripts online that just requires an IP address to initiate some form of attack against a computer. Most of these scripts are attacks on vulnerabilities that have been documented for months. Patches are usually available within a few days after the vulnerability is released. However, if the security administrator does not implement these patches, his network is vulnerable to such attacks.

A honeypot is a computer system that is purposely set up on the web to be attacked. Security professionals commonly use Honeypots as an educational tool to show how black hats probe and exploit a system. All aspects of network defence lead back to education.

Understanding how these attacks take place and knowing what the common attacks are, aid Security Administrators in defending their networks. The fight against hacking effort can benefit enormously just by educating Security Administrators as to counteract the threats that are out in the open.

4.2 Preventing Incidents and Disasters through Policies and Procedures

A security-event policy is a must have for all organisations. It greatly minimises the cropping of incidents and disasters within the organisation. All levels of the organisation need to be aware of the policies. The higher the severity of the incident, the more likely upper management will get involved in some way. Having set procedures for handling incidents will greatly assist without top management support. It also gives a level of cover for the technicians directly involved in the incident response, in the form of procedures for middle management to interface with the rest of the organisation.

Ideally, your Disaster Recovery policy should define how long certain services may be unavailable before the DR policy kicks in. This will help incident response, as these kinds of events are disasters.

If the event is of a type where the recovery window will not be met (example: an onsite-backup DR site gets a realtime feed of changed data, and the intruders deleted a bunch of data that got replicated to the DR site before they were noticed. Therefore, offsite recovery procedures will need to be used) then upper management will have to get involved for the risk assessment decisions.

Some components of any incident response plan:

⇒ **Identify the compromised systems and exposed data.**

⇒ **Determine early on whether or not legal evidence will need to be retained for eventual prosecution.**

♦ If evidence is to be retained, do not touch anything about that system unless absolutely required. Do not log into it. Avoid browsing through log files.

♦ If evidence is to be retained, the compromised systems need to be left online but disconnected until such time as a certified computer forensics expert can dissect the system in a way compatible with evidence handling rules:

    ∗ Powering off a compromised system can taint the data.

    ∗ If your storage system allows you to do this, take a snapshot of the affected Logical Unit Numbers (LUNs)[1] before disconnection and flag them as read-only.

♦ Evidence handling rules are complex. They are not recommended unless you have received proper training on them. Most general System Administrators do not have this kind of training.

[1] *A logical unit number or LUN is a number used to identify a logical unit, which is a device addressed by the SCSI*

♦ If evidence is being retained, treat the loss of service as a hardware-loss disaster and start recovery procedures with new hardware.

⇒ **Pre-set rules for what kinds of disasters require what kinds of notice. Laws and regulation vary by locality.**

♦ Rules pertaining to ‘exposure’ and ‘proven compromise’ often vary

♦ Notification rules will require the Communications department to get involved.

♦ If the required notice is big enough, top-level management will have to be involved.

⇒ **Using DR data, determine how much time can be spent before getting the service back on line becomes a higher priority.**

♦ Service-recovery times may require the work of figuring out what happened, to be then subordinated. After that a drive image of the affected device will need to be taken for dissection after services are restored (this is not an evidentiary copy, it is for the technical engineers to reverse engineer).

♦ Service-recovery tasks should be planned so as to include a complete rebuild of the affected system, rather than cleaning only part of the affected system.

♦ In some cases service-recovery times are tight enough that disk images need to be taken immediately after identifying a compromise has occurred and legal evidence is not to be retained. Once the service is rebuilt, the work of figuring out what happened can start.

⇒ **Sift through log files for information relating to how the attacker got in and what they may have done once in.**

⇒ **Sift through changed files for information relating to how they got in, and what they did once they got in.**

⇒ **Sift through firewall logs for information about where they came from, where they might have sent data to, and how much of it may have been sent.**

Policies and procedures should already be in place before a compromise, and well communicated to the people who will be implementing them in the event of a compromise. It provides everyone with a response framework at a time when people will not be thinking straight. Upper management can think about lawsuits and criminal charges, but actually bringing a case together is an expensive process and knowing that beforehand can help minimise the damage.

Knowing your service recovery times helps set expectation for how long the security response team can have for pouring over the actual compromised system (if not keeping legal evidence) before it is needed in the service-recovery.

4.3 Preventing viruses, Trojan horses, worms and system (access, OS and application) vulnerabilities through security technologies

Harden your systems (also called “lock-down” or “security tightening”) by :

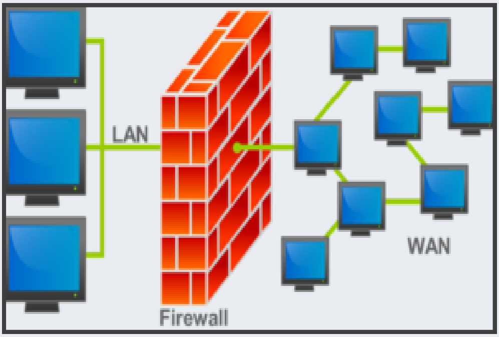
- Configuring necessary software for better security, for example anti virus, anti spyware and firewall
- Deactivating unnecessary software - disable any daemons that are not needed or seldom used, as they are the most vulnerable to attacks
- Configuring the base operating system for increased security
- Logs have to be kept, such as Webserver access logs, Database server authentication logs and Application-specific usage logs
- Proper access controls need to be enforced
- Ensure rights are correctly set on all systems
- Periodic audits of Access Control Lists (ACLs) to ensure that procedures are actually being followed, and temporary troubleshooting steps have been correctly removed after troubleshooting needs to be performed
- All firewall pass-through rules need to be configured, and audited periodically
- Webserver and filesystem Access Control Lists also need to be audited

- Enforce change management
- More than one person should centrally track and review any changes to the security environment
- Guest accounts need to be disabled
- Passwords are should not be set to defaults
- Change off-the-shelf applications that may setup users with predefined passwords have to be changed
- Least-privilege has to be practised. Users have to be given the access they actually need.
- Regular use of generic administrator or root accounts should be discouraged as they make it hard to track who was doing what and when.

Patch all your systems

Intruders can gain root access through the vulnerabilities in your programs, therefore, keep track of "patches" and/or new versions of all the programs that you use (once the security hole is found, manufacturers usually offer patches and fixes quickly before anyone can take advantage of the holes to any large extent), and avoid using new applications or those with previously documented vulnerabilities.

Install a firewall on the system or on the network



Firewalls refer to either software (e.g. ZoneAlarm) and/or hardware (e.g. Symantec-Axent's Firewall/ VPN 100 Appliance) that block network traffic coming to and leaving a system, and give permission to transmit and receive only to user-authorized software. They work at the packet level and do not only detect scan attempts but also block them. A packet-filtering firewall is required as it is the quickest way to enforce security at the border to the Internet.

The following suggestions/services for stopping unauthorized access, using firewalls are highly recommended:

- Tighten the routers at your border to the Internet in terms of packets that can enter or exit your network
- Deploy strong packet filtering firewalls in your network (either by bridge- or routing mode)
- Setup Proxy Servers for services you allow through your packet-filtering firewalls (can be client- or server-side/reverse proxy servers)
- Develop special custom made server or Internet services client and server software

Assess the security of your network:

- Portscan your own network from outside to see the exposed services (TCP/IP service that should not be exposed, such as FTP).
- Run a vulnerability scanner against your servers (commercial and free scanners are available) to monitor your network traffic (external and internal to your border firewalls).
- Refer to your system log - it will reveal (unauthorized) services running on the system and hacking attempts based on format string overflow that usually leave traces there.
- Check your firewall logs - all packets dropped or rejected and persistent attempts should be visible.

While creating passwords, do not use :

- Real words or combinations of words
- Numbers of significance such as date of birth
- Similar/same password for all your accounts
- Words from dictionary

Use encrypted connections

Encryption between client and server requires that both ends support the encryption method.

- If Telnet, POP or FTP is used, a proper authentication mechanism should be in place. For e.g. strongly encrypted passwords
- Use the Secure Shell (SSH) protocol instead of Telnet or FTP for secure data communication
- Never send sensitive information over unencrypted email

Do not install software from unreliable sites

These programs can hide “Trojans”. If you have to download a program, use a checksum, typically PGP or MD5 encoded, to verify its authenticity prior to installation.

Limit access to your server(s)

Restrict access to sensitive areas of your server(s)’ file systems or to the applications that they run. Only give access on a “need-to-know” basis or as per the principle of least privilege.

Use of compromised systems by hackers

Compromised systems can be used again only after formatting the hard disk(s) and re-installing the operating system to ensure that they are well protected.

Use Anti-Virus Software (eg. Norton Anti-Virus or McAfee)

Keep your virus definitions up-to-date and scan your system regularly for viruses.

4.4 Preventing attacks through deception techniques (honeypots or honeynets)

Honeypots or honeynets can be used to educate Administrators and also to catch black hats. A well-configured honeypot can be a warning of activity to come for black hats. Since honeypots are usually the least secure, they are likely to be the first ones attacked. Alerts spawned by the attacking of these boxes can keep an attacker from moving on to any of the real boxes in a network.

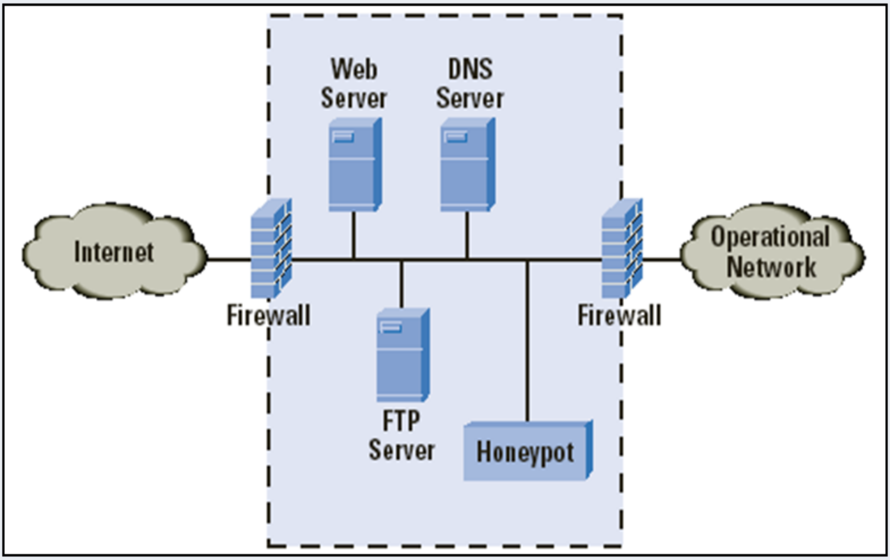


Figure 8: An illustration of a typical honeypot model within a network

Before setting up a honeypot, an administrator needs to protect his network from the box being compromised. The following rules should be applied:

- Confine the honeypot to its own network. If the box is compromised, ensure that the hacker does not have access to the rest of the production network
- In addition, do not block all outgoing access on the honeypot. If an attacker compromises the box and then cannot have access to anywhere else, he will get suspicious and leave. This may lead to not gaining enough information about the attacker.
- If a Network Intrusion Detection System (NIDS) is being utilised on the network, make sure there are special alerts sent out for any honeypot. These alerts can be early warning signs that an attacker is looking at your network.
- Store logs that could be used as evidence out of the honeypot. If a skilled attacker compromises the box, the first thing that he will do is look to delete or change the logs.
- Keep the honeypot up to date.

A very common port scan on a system is for port 27374, or the SubSeven port. SubSeven is a very powerful Trojan that can do any number of things to a victim’s computer. This deception tool looks like a SubSeven server listening on the standard port. It logs all activity that goes on between the client and the fake server. The user can send the attacker a message with the attacker’s IP saying that he is being logged on as well as some other useful information. This is more of a tool geared toward the home user instead of a corporate environment.

Honeypots turn the tables on the black-hat world. The administrator can collect IP addresses fairly easily and report these addresses to the concerned authorities. While honeypots are a good educating tool and a good way to catch some beginner black hats in the act, these boxes should be monitored very closely and should only be implemented on networks by experienced Security Administrators.

4.5 Preventing Mobile Phone Hacking (Mobile Attacks)

Smartphones are becoming increasingly prevalent in today's society. The smartphone is no longer a simple device that enables people to communicate. With emerging technologies, it now serves as portable media players and camera phones with high-resolution touchscreens and web browsers. Furthermore, it includes additional features with advanced computing capability and connectivity as compared to a simple feature phone. As your smartphone rises from gadget to a necessity tool, it is increasingly becoming the target of hacking. Mobile threats can range from simple (such as when someone finds your phone and reads all of your e-mail) to the highly complex (such as Trojan horses, viruses, or third-party apps that share your personal information). Therefore, it is crucial that you

The following sections show some of the attacks on mobile phones:

Password hacking

Be more cautious about protecting your password, irrespective of your phone model. For many services, PINs or passwords are available by default. If you do not change them, your phone becomes more prone to password hacking.

Voicemail Hacking

To prevent someone from accessing your phone’s voicemail, you have to make sure that you change your phone’s remote access PIN. This can be done by calling voicemail and listening to the security setting or even contacting the cell phone provider and ask for instructions. It is important to change your password regularly to minimise the chances of your mobile phone being hacked.

Data Gathering

Data Gathering means collecting data ranging from your name to passwords or credit card information. Preventing this type of hacking is more difficult. Thus, it is very important to create a strong password. Many people use information like their date of birth - easy to remember but also easy for other people to guess. You should think about making sure your passwords are unique and not related to you.

Many smartphones have an auto complete feature, hence ensure either to turn it off or at least deactivate the feature for programs and websites for which you log in. Otherwise, your phone will store the information, making it more accessible to a successful hacker.

Delete your browsing history regularly, and turn off Bluetooth, Wi-Fi, and GPS when you are not using them. Also, control your phone's Bluetooth visibility. Bluetooth is used to connect devices to each other, and when visibility is on it is possible for a hacker to reach through another device into your phone. To enhance your phone's protection even more, turn it off when you are not using it, or take out the battery.

5.0 Conclusion

Lately, exploit development has been catching up with security research. In many cases, exploits are being developed and released in less than a day after a 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 System Administrators and strict adherence to a well thought organisational information security policy can reap better results in the current scenario of multiple vulnerabilities.

CERT–MU

Computer Emergency Response Team of Mauritius (CERT-MU)

National Computer Board,

Tel: 210 5520

Fax: 208 0119

Website: www.cert-mu.org.mu

Incident Reporting

Hotline : 800 2378

Email: incident@cert-mu.gov.mu

Vulnerability Reporting

Email: vulnerability@cert-mu.gov.mu

General Queries

Email: info@cert-mu.gov.mu

Mailing List Subscription

Email: subscribe@cert-mu.gov.mu