Global Information Assurance Certification Paper

Rootkit: Attacker undercover tools.

Saliman Manap

21/10/2001

1.0 Introduction

Today, hacker, attacker and intruder are common problem in security threat. Lots of vulnerability discover and publish, make it easier for them to develop new technique and method of hacking. The vulnerability that was purposely publishing for education was misused as a tool to do destruction and other malicious activity and lots of hacker tools specially written and publish to the nets and even script kiddies can used it for some destruction.

In this paper we will briefly take a look on rootkit, the famous tools used as undercover tool for hackers or intruder. I hope we will understand more on rootkit and what capability it has in the sense of hacker, administrator and user perceptive view.

This paper is intend to make a better and clear view about rootkit which was post by Sunnie Hawkins in the article "Understanding the Attackers Toolkit" January 13, 2001"s and Andrew R.Jones - "A review of loadable kernel modules" June 12, 2001.

1.1 What is Rootkit?

Rootkit name are combination from two words, "root" and "kit". "Root" was taken from "root" a name of UNIX administrator, which is the highest-access level in UNIX environments while "kit" can be refer as tools. From this word we can interpret rootkit as tools or collection of tools that enable attacker to keep the root power on the compromised system in order to keep the continuously power over the compromised server he/she should hide their presence from being detected by administrator. This is what actually rootkit do. So the best meaning we can describe rootkit is it is a tool or collection of tools that hide an attacker presence and at the same time give the attacker ability to keep full control the server or host continuously without being detected.

In UNIX environment the attacker installs a rootkit on a computer after first obtaining the access level, either by user-level access or administrator-level access. Administrator-level access is needed for most rootkit installation this can be done by exploiting known remote vulnerabilities to gain the root-level access. If the attackers only have user-level access, local exploit or cracking administrator password need to be done in order to get full access level before rootkit successfully installed.

As attacker undercover tools, rootkit program must have a capability to mask the intrusion and his presence. The rootkit may consist of several utilities such as:

- **Back door programs**

Backdoor is an unauthorized way of gaining access to a program, online service or an entire computer system. The backdoor is written and installed to let they log in to the hacked system without using an exploit again. A backdoor is a potential security risk and can be installing from various backdoor types such as:

- Login Backdoor – Modifying login.c to look backdoor password before stored password. Attacker can log into any account using backdoor password.
- Telnetd Backdoor – Trojaned the "in.telnetd" to allow attacker gain access with backdoor password.
- Services Backdoor – Replacing and manipulate services like "ftp", "rlogin", even "inetd" as backdoor to gain access.
- Cronjob backdoor – Backdoor could also be added in "crontjob" to run on specific time for example at 12 midnight to 1 am.
- Library backdoors – Almost every UNIX and Windows system have shared libraries. Shared libraries can be backdoor to do malicious activity including giving a root or administrator access.
- Kernel backdoors – This backdoor is basically exploiting the kernel, which is core of the operating system to handle and to hide backdoor effectively (will be explained in kernel rootkit).
- Network traffic backdoors which typically using TCP, UDP, and ICMP – Backdoor that exploiting network traffic protocol is widely used. In TCP protocol backdoor like ssh is popularly used because it communicate in encrypt, while crafting and tunneling packet in UDP and ICMP traffic will give a better chances escaping from firewall and "netstat".

All of these and any other backdoor are explained and documented by Christopher Klaus can be reach at http://secinf.net/info/unix/backdoors.txt

➢ **Packet sniffers**

Packet Sniffer is a program and/or device that monitor data traveling over a network, TCP/IP or other network protocol. The attackers used the sniffers to listen or to steal valuable information off a network. Many services such as "ftp" and "telnet" transfer their password in plain text and it is easily capture by sniffer. Jason Drury has a good paper on sniffer at http://www.sans.org/infosecFAQ/switchednet/sniffers.htm

➢ **Log-wiping utilities**

Log file are the lists actions that have occurred. For example in UNIX operating system, wtmp log are file that listing time and date user log in into the system. Log file are important to administrator to monitor, review system performance and detect any suspicious activities. By deleting intrusion log in the log file attacker can hide the intrusion from being detect easily.

➢ **Miscellaneous programs**

The rootkit also contained others programs which depending on type of rootkit packages, some rootkit packages do offer;

DDOS program – to install and to set at the compromised server or host to be a DDOS client such as, trinoo.

IRC program – IRC program and bot are common program that installed by attacker at the compromised server or host. This IRC bot will connect to the nets and log on some server waiting for the attacker to issue a command to them. By issuing some command control attacker

can trigger a distributed denial of service (DDoS) attack. The best paper to explain this attack is well written by Steve Gibson can be reach at http://grc.com/dos/grcdos.htm

Attacker utility – other utilities sometimes discover on compromised system with the rootkit tool and provided by rootkit for example,

System patch - attacker patch the system after they successfully compromised the system. Patching the system will prevent other attacker to gain access into the system again. Since the backdoor was installed there no need to attacker to exploit the vulnerability again.

Log editor – log editor is useful to edit the log file on compromised system.

2.0 Chronology of Rootkit

In 80's UNIX was dominated as networking operating system. UNIX was chosen since it is the strong operating system in that times which can be handle multitasking and network environment process successfully. Unix system have some system tools to monitor the process and access such as ls- to show and listing file, who- to determine who is logging in the system, ifconfig- to check the status of Ethernet interface whether it is in promiscuous mode.

Since UNIX have system and log utilities that will check process running on the system, a brilliant attackers continuously try to find a way to bypass this mechanism in order to hide their presence. The way to bypass and to fool UNIX system then was publish such Hiding Out Under Unix by Black Tie Affair (Phrack Volume Three, Issue 25, File 6, March 25, 1989)

The earliest Trojan horse programs were bundled together in the form of "Root Kits" is original written for Sun's Berkeley flavor of Unix (SunOS 4) and later for Linux.

Rootkit intrusion was famous in 90's due to lots of finding compromised server was installed by rootkit. CERT and CIAC was publish and advisory due of increasing rootkit compromised this was CERT advisory CA-1994-01 "Ongoing network monitoring attack" February 3 1994, and CERT Advisory CA-95:18 - Widespread Attacks December 18, 1995 and it was reported in CERT* Coordination Center 1995 Annual Report. At the same time CIAC also publish their advisory which is E-09: Network Monitoring Attacks February 3, 1994 and E-12: Network Monitoring Attacks Update March 18, 1994

The famous rootkit was SunOS 4 rootkit and Linux Root Kit (lrk3) these rootkit can be downloaded from http://209.100.212.5/cgi-bin/search/search.cgi?searchvalue=rootkit Linux Root Kit version 3 (lrk3), was released in December of 1996, further added tcp wrapper Trojans and enhanced the programs in the kit. That was the most common method of concealing activity and stealing passwords by sniffing on the new favorite target of attackers, x86 compatible PC's running Linux.

The introduction of Linux was brought a new technique for attacker since the Linux kernel was freely available to download from Internet. The Rootkit LKM (loadable kernel module) "heroin.c" was the first malicious kernel that has been published to bugtrag. The first rootkit LKM was published on the net was knark are modified from "heroin.c" script.

3.0 Type of Rootkit

We can categories the rootkit into two types.

1. Application rootkit - established at the application layer.
2. Kernel rootkit - establish more deep into kernel layer.

This rootkit differences is not basically where they establish, but also the way they hide in the system.

3.1 Application Rootkit

Application rootkit was the conventional rootkit and widely used in loosely environment. The method using by application rootkit is replacing the good system application with trojaned system file. The trojaned system file will provide backdoor, hiding the attackers presence and it also will not log any connection and activity done by the attacker. Below is listing file usually replace by attacker.

- **Programs replace to hide attacker presence.**
  - "ls", "find", "du" — Trojaned system file will be able to hide attackers file, directory and stuff that have been brought into the system from being listing.
  - "ps", "top", "pidof" — All these programs are process monitor program. Trojaned program will hide attacker process from being listing.
  - "netstat" — netstat is used to check network activity such as open port, network connections establish and listening. Trojaned netstat will hide processes installed by attacker such as ssh daemon or other services.
  - "killall" — Trojaned "killall" will not be able to kill attacker process.
  - "ifconfig" — When sniffer is running PROMISC flag is set to the nic. "ifconfig" is a handy utility to set and to view setting of ethernet nic. Trojaned "ifconfig" will not display the PROMISC flag when sniffer is running. This is useful to hide sniffer from being detected.
  - "crontab" — Trojaned "crontab" will hide the attacker's crontab entry.
  - "tcpd", "syslogd" — Trojanised "tcpd" and "syslog" will not log any connection made by attacker. "tcpd" also capable to bypass tcp wrapper enforcement.

- **Program with backdoor**
  - "chfn" — root shell can be gain if backdoor password is entering as new full name.
  - "chsh" — root shell can be gain if backdoor password is entered as new shell.
  - "passwd" — root shell can be gain if rootkit password is entered as current password.
  - "login" — can log into any username including root if rootkit password is enter in password prompt.
  - "bd2" — Trojaned rpcbind program will allow the attacker to run arbitrary commands on the target system.

- **Network Daemons with backdoor**
  - "inetd" — Trojaned "inetd" will open port for attacker to log in. The password must be entered in the first line to gain root access.

- "rshd" — trojaned so that if the username is the rootkit password, a root shell is bound to the port (i.e. rsh [hostname] -l [rootkit password]).
- "rsh" – Trojaned "rsh" can give attacker root access by issue "rsh [hostname] -l [rootkit password]".
- "sshd" – Sometime a ssh daemon is installed to give the attacker secure channel from being capture by authorized sniffer.

➢ **Sniffer Program**
- "linsniffer" — a small network sniffer for Linux.
- "sniffchk" — program to check and to make sure a sniffer is still running.
- "le" — Solaris Ethernet packet sniffer.
- "snif" — another packet sniffer for linux.
- "sniff-10mb" — a sniffer designed to work on a 10mbps Ethernet connection.
- "sniff-100mb" — a sniffer designed to work on a 100mbps Ethernet connection.

➢ **Other Utilities**
- fix — installs a trojaned program (e.g., ls) with the same timestamp and checksum information.
- wted — wtmp editor. You can modify the wtmp.
- z2 — erases entries from wtmp/utmp/lastlog.
- bindshell — binds a root shell to a port (port 31337 by default).
- zap3- erased their tracks from wtmp, utmp, lastlog, wtmpx, and utmpx. zap3 looks for log files in commonly used log directories such as/var/log, /var/adm, /usr/adm, and /var/run.

➢ **Other Method Hiding Stuff**

Method to hide the presence this type of rookit, attacker usually keep it in hidden directory or file. File or directory begin with dot "." or invisible " " are easiest method to hide stuff from administrator eyes. Directory or file begins with dot "." Will not be listed by "ls" command unless flag "–a" is used. Invisible " " directory or file are easily to create and usually not notified by administrator.

Place used by attacker to hide his rootkit or his staff is placing on the directory which is not usually checked by administrator several favorite place such as "/var", "/dev" or "/lib".

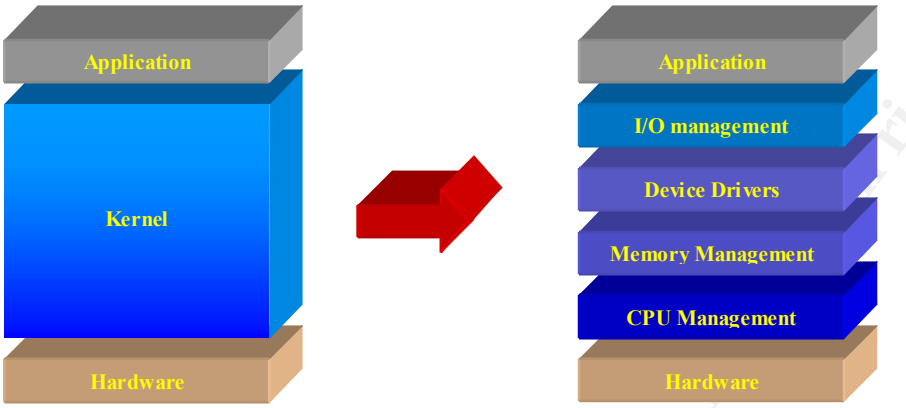
3.2 Kernel Rootkit

Kernels rootkit are powerful rootkit which less detectable than application rootkit. By manipulating and exploiting kernel capability it's become hardest rootkit to detect because it can bypass conventional system integrity checker at application layer.

Although first release of kernels rootkit was mainly written for Linux but it can be modified to be port to other operating system as well. Several document was written for other operating system, For FreeBSD; Attacking FreeBSD with Kernel Modules was written by pragmatic / THC on Jun 1999. For Solaris; Solaris Loadable Kernel Modules written by Plasmoid / THC in 1999 and for windows some development on rootkit can be access at http://www.rootkit.com

Before we move on next topic lets have a look on what is kernels.

3.2.1 What is kernel?



OS Layer source: http://www.howstuffworks.com/operating-system.htm

Kernels are the central modules of the operating system. It is the part of the operating system that loads first, and it remains in main memory. Because it stays in memory, it is important for the kernel to be as small as possible while still providing all the essential services required by other parts of the operating system and applications. Typically, the kernel is responsible for I/O management, Device drivers, CPU management, process and task management, and disk management.

3.2.2 How Kernels Rootkit works?

Kernel rootkit is basically exploiting the useable of LKM feature to do malicious activity. LKM are a very useful feature in Linux and any other system, which support LKM such as freeBSD and Solaris. It is used to load device drivers on as-needed basis.

As we know every OS has some functions build into its kernel, which are used for every operation. The functions uses are known as system calls. They represent a transition from user to kernel space. Opening a file in user space is represented by the sys_open system call in kernel space. By manipulating system call capability intruder can be gain the tremendous power to do malicious activity.

- **Hiding files.**

By manipulating system call function, command like "ls", "du" can be exploit to hide file or directory from being listing by the issuing command and hidden from everybody. In linux this can be done by manipulating sys_getdents() system call function.

- **Hiding processes.**

In UNIX implementation, process information is mapped to directory in "/proc" file system.
Hiding processes can be done by manipulating sys_getdents() system call function with additional effort of making this invisible in the task structure.

- **Hiding network connections.**

Similar to process hiding, hiding network connection can be done by preventing it to be log inside "/proc/net/tcp" and "/proc/net/udp" files. The idea for kernel rootkit is trojaned the sys_read(). Whenever reading these two files and a line matching certain string, the system call will hide it from user.

- **Hiding the sniffer.**

To hide the sniffer is basically hiding the promiscuous flag of the network interface. The system call to Trojan in this case is sys_ioctl().

- **Hiding the LKM itself.**

A perfect LKM rootkit is must be able to hide it from the administrator. The LKM's in the system are kept in a single linked list, to hide the present of LKM rootkit, the LKM rootkit can remove it from the list so that command like "lsmod" won't show it.

- **Hiding symbols in the LKM.**

Normally functions defined in the LKM will be exported so that other LKM can use them. Hiding these symbols is necessary and macro can be used is "EXPORT_NO_SYMBOLS". This will prevent any symbol from being exported.

- **Communicating with LKM.**

After LKM rootkit was installed, now the attackers want to tell the kernel to hide another file. How can he do it? We know the normal way from the user land to talk to kernel land is through the system calls, so kernel rootkit have to modify some system calls. For example, kernel rootkit could replace sys_settimeofday(). When a special parameter is passed, trojaned system call will do appropriate things for attacker.

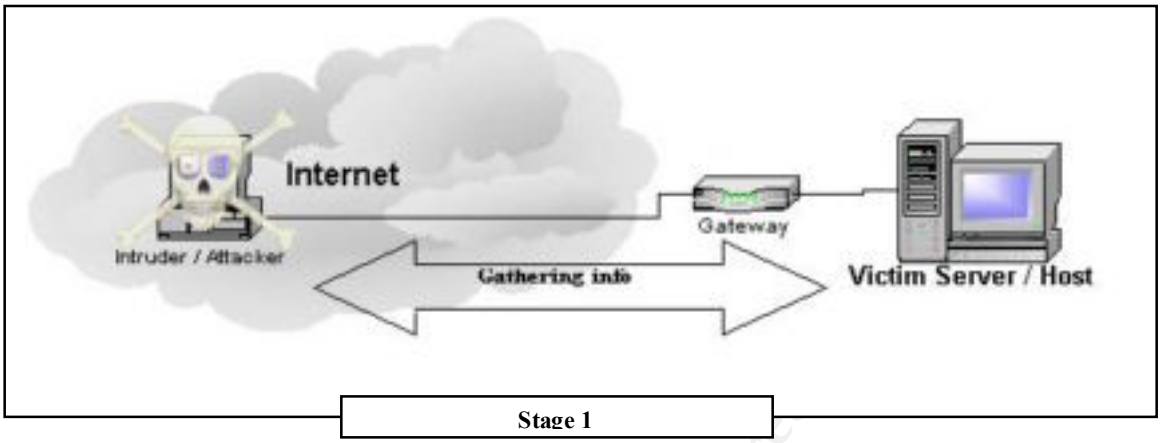
- **Redirecting File execution.**

Sometimes, the attacker may want to replace the system binaries, like "login", but doesn't want to change the file. Kernel rootkit can replace sys_execve(). Thus, whenever the system tries to execute the "login" program, it will be re-directed to execute the attacker's version of login program.

4.0 The Stages of Rootkit Compromised

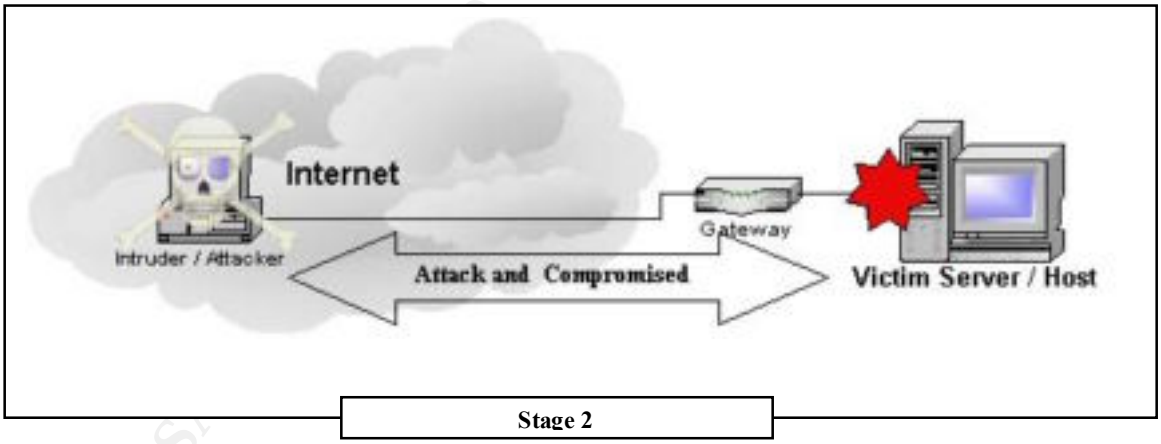
After we knew the type of rootkit lets look at the stages of rootkit compromised. There are several ways for a host to become compromised and rootkit was used to hide the attacker presence. For example, the user could be trick to open a hole or backdoor or sometimes the perimeter defense such as firewall went down. The following is one example of how hosts can be compromised and rootkit is installed.

- **Stage 1 - Investigating victim host for vulnerability**



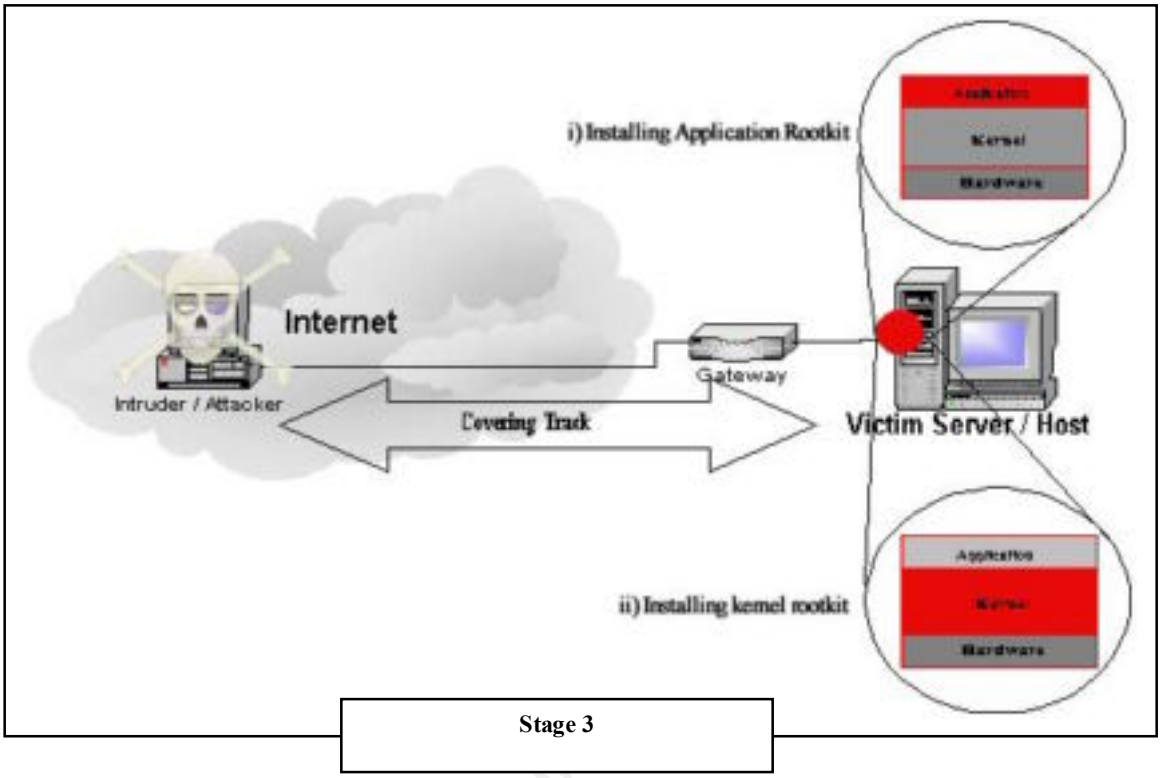
Before an attacker want to attack and compromised a system he/she should know what vulnerability of the system has. This stage involves foot printing, scanning and enumeration in the hack anatomy. The objective this stage for attacker is to gather as much info as he/she can for the targeted servers. Attacker will use some technique such as whois, dns querying, ping sweep, OS detection, list user accounts and other methods which is needed to gain and identified any weakness and vulnerable on the victim. Please bear in mind that in this stage any account even just a regular user account can be valuable things to intruder.

- **Stage 2 – Attack and Compromised The Server**



When enough data is gathered and system vulnerabilities are detected, it is time to crack and compromised the server. This stage will involve gaining access, escalating privilege and filtering in anatomy of hack. In some cases only low-level access is needed to gain access to the server such as normal users. After the attacker have gain access then he/she tried to get root access by exploiting local vulnerabilities. There is lots of local exploiting can give normal user root access such as local xwindows and local games exploit. Remote exploit like ftp exploit, telnetd exploit that could give the root access-level is more preferable by the attackers.

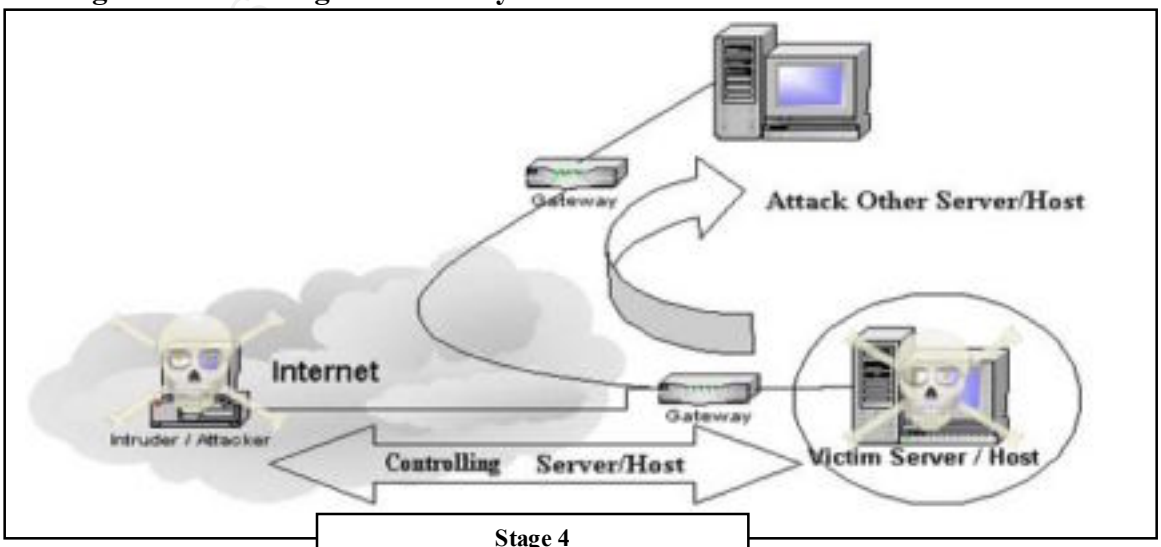
- **Stage 3 - Rootkit Installation**



Stage 3

Rootkit installation is in covering track and creating backdoor stage in hack anatomy. Installing rootkit is easiest things for example torn rootkit installation can be done by extracting the pack and run "/t0rn <password> <ssh-port> while some of the kernel rootkit module is just issue insmod <lkm rootkit>. Installing one of the type preferable or both can be done after root-access level is gained.
Since some of the rootkit include the log wiping utilities server intrusion and attacker log can be easily wipe out from the logs by using the tools. These tools also can be downloaded freely from the Internet.

- **Stage 4 - Controlling the victim system**



Stage 4

After the attacker has gaining access and rootkit was installed they have fully access to the system or server and the best part of it was their presence of activity was hide from the administrator. Compromised system or server is a good asset for attacker since it can be used as launch pad to compromise other system, it can be used to store his nasty stuff and the bad of all it can be used as DDOS agent o DDOS other system or local system.

5.0 Lists of Rootkit Analysis and dedicated info involving rootkit

For further knowledge below is the list of deep analysis and info made and written by security professional regarding cases involving rootkit.

- Analysis of the T0rn rootkit
- Analysis of N.F.O hacking- / rootkit
- Analysis of Rootkit/Smurf Payload Toolkit v 1.1
- Knark: Linux Kernel Subversion
- The Compromise, A Case Study by Mary M. Chaddock
- Linux rootkit from the mass-lpd autohacker (rh7.tar.gz)
- Automatic tool to compromise RedHat 7 using lpd
- Analysis of the Linux rootkit YoYo.tar.gz
- Analysis of the Linux rootkit TeLeKiT

Combination worms with rootkit that attack internet.

- The Lion Worm: King of the Jungle?
- The Ramen Worm and its use of rpc.statd, wu-ftpd and LPRng Vulnerabilities in Red Hat Linux
- Ramen Noodles
- RAMEN – A Linux Worm

6.0 Countermeasure

The best countermeasure can be done is eliminating or migrating vulnerability from approaching the threats. This will include;

- **Host Hardening, Systems patch and update**

Hardening the host or server is important. Hardening the host or server can save the host or server if network protection is failed in any condition. Beside that system patch and update is importance to avoid any vulnerability occur in the operating system. This will include:

Eliminating The Twenty Most Critical Internet Security Vulnerabilities

This Document was published by SANS (System Administration, Networking, and Security) Institute and the National Infrastructure Protection Center (NIPC) in the Federal Bureau of Investigation (FBI) are basically covering Top Twenty Most Critical Internet Security Vulnerabilities. It was compiled with assistance from computer security experts in academia, industry and government.

The good compilation for any new bug exploit on major operating system and application are:

Cybernotes: CyberNotes is compilation for various security expert groups and it is published every two weeks by the National Infrastructure Protection Center (NIPC). Its mission is to support security and information system professionals with timely information on cyber vulnerabilities, attacker exploit scripts, attacker trends, virus information, and other critical infrastructure-related best practices.

Safermag: SAFER is another security compilation from various security expert group and it has been published by eGlobal Technology's security team since June 1998. Its purpose is to provide a service to the security community by summarizing and commenting on all new security vulnerabilities and tools, on a monthly basis.

- **Detecting the Rootkit**

Several programmers were written especially to detect rootkit in system environment this include;

chkrootkit <http://www.chkrootkit.org>
Famous rootkit scanner was specially build for UNIX has been tested on: Linux 2.0.x, 2.2.x, FreeBSD 2.2.x, 3.x and 4.x, OpenBSD 2.6, 2.7, 2.8 and 2.9, Solaris 2.5.1, 2.6 and 8.0. IT capability can detect most rootkit including lrk3, lrk4, lrk5, lrk6 (and some variants),Solaris rootkit, FreeBSD rootkit ,t0rn (including some variants and t0rn v8), Ambient's Rootkit for Linux (ARK), Ramen Worm, rh[67]-shaper, RSHAm, Romanian rootkit, RK17, Lion Worm, Adore Worm, LPD Worm, kenny-rk, Adore LKM, ShitC Worm, Omega Worm, Wormkit Worm, Maniac-RK, dsc-rootkit, Ducoci rootkit, x.c Worm.

rkscan <http://www.hsc.fr/ressources/outils/rkscan/>.
Another rootkit scanner that can detect LKM rootkit: Adore and knark

Carbonite<http://www.foundstone.com/rdlabs/termsofuse.php?filename=carbonite.tar.gz>
Carbonite is a Linux Kernel Module to aid in RootKit detection is another module that will help in detecting loadable kernel module. Carbonite is an lsof and ps at the kernel level. Carbonite "freezes" the status of every process in Linux's task_struct, which is the kernel structure that maintains information on every running process in Linux.

rkdet <http://www.vancouver-webpages.com/rkdet/>
Rkdet is a daemon intended to catch someone installing a rootkit or running a packet sniffer. It is designed to run continually with a small footprint under an innocuous name. When triggered, it sends email, appends to a logfile, and disables networking or halts the system. It was designed to install with the minimum of disruption to a normal multiuser system, and should not require rebuilding with each kernel change or system upgrade.

LSM (Loadable Security Module) <http://freshmeat.net/projects/lsm/>
LSM is a very simple but effective intrusion prevention loadable kernel module. Current functionality involves protecting extended file attributes on ext2 from being modified by the super user and the module from being removed and other modules from being loaded. The basic protection also prevents access to raw device; so debugfs cannot be used on a disk partition nor

cannot be changed to the boot process occur. You will not be able to reconfigure lilo, if the module gets activated.

- **Network and perimeter defense**

Network and perimeter defense such as firewall should be review and implement carefully. Continuously monitoring should be done to detect any unauthorized access to the protected system. An IDS also should be placed, monitor and the signature is update regularly. This is important to detect any new attack occur in the network.

On detecting backdoors, Ying Zhang and Vern Paxson has proposed Detecting backdoors on Proc. 9th USENIX Security Symposium, Denver, CO, August 2000.

7.0 Conclusion

Rootkit is a dangerous tool in the hand of intruders. The continuously development has made it a perfect undercover tools for attacker. Preventing system from being rootkit is not just simply over the network protection alone. Host and application are also important aspect to cover up in the security measure. We have to remember that, lots of technology information kept growing and flowing on the nets. Technology, combine with creative genius of attacker thinker around the world, is leading to the development and application of new form of warfare. Yet we have to be well prepared for it.

8.0 Reference

Sunnie Hawkins, Understanding the Attackers Toolkit**,** January 13, 2001,URL: http://www.sans.org/infosecFAQ/linux/toolkit.htm

Andrew R. Jones, A Review of Loadable Kernel Modules**,** June 12, 2001, URL: http://www.sans.org/infosecFAQ/linux/kernel_mods.htm

Jason Drury, Sniffers: What are they and How to Protect From Them, November 11, 2000, URL: http://www.sans.org/infosecFAQ/switchednet/sniffers.htm

DeokJo Jeon**,** Understanding DDOS Attack, Tools and Free Anti-tools with Recommendation**,** April 7, 2001,URL: http://www.sans.org/infosecFAQ/threats/understanding_ddos.htm

Steve Gibson, The Strange Tale of the Denial OF Service Attacks Against GRC.COM, Gibson Research Corporation, Aug 31, 2001, URL: http://grc.com/dos/grcdos.htm

Black Tie Affair, Hiding Out Under UNIX, Volume Three, Issue 25, File 6 of 11, March 25, 1989, URL: http://www.phrack.org/show.php?p=25&a=6

Christopher Klaus, Backdoors, August 4 1997, URL: http://secinf.net/info/unix/backdoors.txt

CERT, CERT® Advisory CA-1994-01 Ongoing Network Monitoring Attacks, February 3, 1994, URL: http://www.cert.org/advisories/CA-1994-01.html

CERT, CERT® Advisory CA-95:18 - Widespread Attacks, December 18, 1995, URL: http://www.cctec.com/maillists/nanog/historical/9512/msg00082.html

CERT, CERT® Coordination Center 1995 Annual Report (Summary), October 14, 1997, URL: http://www.cert.org/annual_rpts/cert_rpt_95.html

CIAC, E-09: Network Monitoring Attacks, February 3, 1994, URL: http://www.ciac.org/ciac/bulletins/e-09.shtml

CIAC, E-12: Network Monitoring Attacks Update, March 18, 1994, URL: http://www.ciac.org/ciac/bulletins/e-12.shtml

Pragmatic / THC, Attacking FreeBSD with Kernel Modules, June 1999, URl: http://packetstormsecurity.org/groups/thc/bsdkern.htm

Plasmoid / THC, Solaris Loadable Kernel Modules, Version 1.0 (c) 1999, URL: http://packetstormsecurity.org/groups/thc/slkm-1.0.html

howstuffworks, How Operating Systems Work, URL: http://www.howstuffworks.com/operating-system.htm/printable

UEN Security, Anatomy of a Hack, URL: http://www.uen.org/security/whitepapers/UEN-anatomy.pdf

David, Brumley, "Invisible intruders: rootkits in practice", login:, Special Issue: Intrusion Detection (Sept.1999). URL: http://www.usenix.org/publications/login/1999-9/features/rootkits.html

David, O'Brien, "Recognizing and Recovering from Rootkit Attacks", published in Sys Admin 5(11) (November 1996), pp.8-20 URL: http://www.cs.wright.edu/people/faculty/pmateti/Courses/499/Fortification/obrien.html

Dave, Dittrich, ""Root Kits" and hiding files/directories/processes after a break-in", version 1.1, 21 January 2001, URL: http://staff.washington.edu/dittrich/misc/faqs/rootkits.faq

Lance, Spitzner, "Know Your Enemy: part III, "They Gain Root"", URL- http://project.honeynet.org/papers/enemy3/

Pragmatic , THC, "(nearly) Complete Linux Loadable Kernel Modules " version 1.0, Mac 1999 URL: http://packetstormsecurity.org/groups/thc/LKM_HACKING.html

Chris Prosise and Saumil Udayan Shah, At the root of rootkits,URL: http://quickenexcite.cnet.com/webbuilding/0-7532-8-4561014-1.html?tag=st.bl.7532-8-4561014-5.txt.7532-8-4561014-1

Last Updated: February 23rd, 2018

Upcoming Training

SANS New York City Winter 2018	New York, NY	Feb 26, 2018 - Mar 03, 2018	Live Event
Mentor Session - AW SEC401	Melbourne, FL	Mar 01, 2018 - May 10, 2018	Mentor
SANS London March 2018	London, United Kingdom	Mar 05, 2018 - Mar 10, 2018	Live Event
Mentor Session - SEC401	Vancouver, BC	Mar 06, 2018 - May 15, 2018	Mentor
Mentor Session - SEC401	Grand Rapids, MI	Mar 09, 2018 - Apr 13, 2018	Mentor
SANS Secure Singapore 2018	Singapore, Singapore	Mar 12, 2018 - Mar 24, 2018	Live Event
SANS Secure Osaka 2018	Osaka, Japan	Mar 12, 2018 - Mar 17, 2018	Live Event
SANS San Francisco Spring 2018	San Francisco, CA	Mar 12, 2018 - Mar 17, 2018	Live Event
SANS Paris March 2018	Paris, France	Mar 12, 2018 - Mar 17, 2018	Live Event
SANS Northern VA Spring - Tysons 2018	McLean, VA	Mar 17, 2018 - Mar 24, 2018	Live Event
SANS Pen Test Austin 2018	Austin, TX	Mar 19, 2018 - Mar 24, 2018	Live Event
SANS Munich March 2018	Munich, Germany	Mar 19, 2018 - Mar 24, 2018	Live Event
Mentor Session - SEC401	Studio City, CA	Mar 20, 2018 - May 01, 2018	Mentor
Mentor Session - AW SEC401	Mayfield Village, OH	Mar 21, 2018 - May 23, 2018	Mentor
SANS Boston Spring 2018	Boston, MA	Mar 25, 2018 - Mar 30, 2018	Live Event
SANS 2018 - SEC401: Security Essentials Bootcamp Style	Orlando, FL	Apr 03, 2018 - Apr 08, 2018	vLive
SANS 2018	Orlando, FL	Apr 03, 2018 - Apr 10, 2018	Live Event
Community SANS Charleston SEC401	Charleston, SC	Apr 09, 2018 - Apr 14, 2018	Community SANS
SANS vLive - SEC401: Security Essentials Bootcamp Style	SEC401 - 201804,	Apr 09, 2018 - May 16, 2018	vLive
Community SANS St. Louis SEC401	St Louis, MO	Apr 16, 2018 - Apr 21, 2018	Community SANS
SANS London April 2018	London, United Kingdom	Apr 16, 2018 - Apr 21, 2018	Live Event
SANS Zurich 2018	Zurich, Switzerland	Apr 16, 2018 - Apr 21, 2018	Live Event
Mentor Session - AW SEC401	Memphis, TN	Apr 17, 2018 - May 17, 2018	Mentor
SANS Baltimore Spring 2018	Baltimore, MD	Apr 21, 2018 - Apr 28, 2018	Live Event
Baltimore Spring 2018 - SEC401: Security Essentials Bootcamp Style	Baltimore, MD	Apr 23, 2018 - Apr 28, 2018	vLive
SANS Seattle Spring 2018	Seattle, WA	Apr 23, 2018 - Apr 28, 2018	Live Event
SANS Riyadh April 2018	Riyadh, Saudi Arabia	Apr 28, 2018 - May 03, 2018	Live Event
Automotive Cybersecurity Summit & Training 2018	Chicago, IL	May 01, 2018 - May 08, 2018	Live Event
Community SANS Houston SEC401	Houston, TX	May 07, 2018 - May 12, 2018	Community SANS
SANS Security West 2018	San Diego, CA	May 11, 2018 - May 18, 2018	Live Event
SANS Northern VA Reston Spring 2018	Reston, VA	May 20, 2018 - May 25, 2018	Live Event