Global Information Assurance Certification Paper

Understanding the Attackers Toolkit

Sunnie Hawkins
13 January 2001

A rootkit is defined by the NSA Glossary of Terms Used in Security and Intrusion Detection as an "A hacker security tool that captures passwords and message traffic to and from a computer. A collection of tools that allows a hacker to provide a backdoor into a system, collect information on other systems on the network, mask the fact that the system is compromised, and much more. Rootkit is a classic example of Trojan Horse software. Rootkit is available for a wide range of operating systems."

A common misconception in network administration/security is that a rootkit is a magical program used to gain root access to a certain computer system. However, a rootkit, is used to hide an intruder in the system and to assist the intruder in keeping the privileged access the intruder have already attained. Intruders want to gain limited user-level access by the easiest means possible, usually socially engineering a password or cracking an easily guessed password. Once inside the network, with non-privileged access, the intruder will then exploit a known vulnerability to gain root access. Some of the more recent vulnerabilities include *rpc.statd* and *wu-ftpd*. The intruder unpacks the rootkit and installs it on the victim's computer system. Once a rootkit has been installed on the victim's network, the intruder will usually check to see if anyone else is on the network at that time. When the intruder sees that the intruder is the only active user on the network, The intruder begins to clean the log and replace or modify files. After the intruder feels safely hidden the intruder either captures/sniffs out more usernames or passwords or the intruder will use the compromised system to launch more attacks.

What's in a rootkit?

Rootkits got their start in the early 90's one of the first advisories came out in Feb 1994. This advisory from CERT-CC addressed "Ongoing Network Monitoring Attacks" CA-1994-01 revised on September 19, 1997. Rootkits have grown more popular and have increased in detection difficulty. The most common rootkits are used for SunOS and Linux operating systems. All rootkits consist of several different programs. A typical rootkit will include Ethernet Sniffer, which is designed to sniff out passwords. Rootkits can also include Trojan programs used as backdoors such as *inetd* or *login*. Support programs such as *ps, netstat, rshd,* and *ls to* hide the attacker directories or processes. Last but not least, log cleaners, such as *zap, zap2,* or *z2*, are used to remove login entries from the *wtmp, utmp,* and *lastlog* files. Some rootkits also enable services such as TELNET, SHELL, and FINGER. The rootkit also includes scripts that will clean up other files in the /var/log and var/adm directories. Using the modified programs of *ls, ps,* and *df* installed on the box, the intruder can "hide" his/her files and programs from the legitimate system administrator. The intruder next uses programs within the rootkit to clean up the extensive log files generated from the initial vulnerability exploitation. The intruder then uses the installed backdoor program for future access to the compromised

system in order to retrieve sniffer logs or launch another attack. If a rootkit is properly installed and the log-files are cleaned correctly, a normal system administrator is unaware that the intrusion has even occurred until another site contacts him or the disks fill because of the sniffer logs. However, most intruders are not careful enough to clean out the entire log files system or they may zero out a log file. That would be most indicative that something has gone wrong on the system, (lucky break for most system administrators). Besides cleaning and trojanizing, most of the system administrator's friends such as *ps*, *df*, and *ls*, many rootkits contain a program named FIX. FIX will take a snapshot of the original systems binary and after the intruder has installed their rootkit, it will move the modified binary into place. FIX mimics all three timestamps: atime, ctime, mtime; as well as date, permissions, user, and group of the original program. With the combination of these wonderful programs, detection of an unknown successfully and carefully installed rootkit could become not just time-consuming and difficult, but almost impossible.

LINUX ROOTKIT IV

During my research, I utilized the extensive information and source code available in Linux Rootkit IV (*lrk4*) written by Lord Somer released in November 1998 and available on www.lordsomer.com. However, lrk4 is not the first Linux rootkit other examples of Linux rootkits are lrk, lnrk, lrk2, and lrk3. Most versions include the normal rootkit components such as sniffers (*linsniffer* or *sniffit*) log editors/erasers (*z2*, *uted*, *lled*), and Trojan horse/backdoor replacement programs to allow remote access, user access to gain root privileges, hide files, process, and connections. Linux Rootkit IV is a very easy rootkit to use, and install. Installation of lrk4 included nothing more that executing the 'make install'. To install a shadow kit you execute the 'make shadow install'. Lrk4 will only work on Linux 2.X kernels. All utilities within lrk4 are all described in the README file for the lr4. Below is a short description of the utilities within lrk4.

1 - Modified programs that hide the intruder:

- ls, find, du – these programs will not count or display the intruder files the data file is ROOTKIT_FILES_FILE, defaults *to /dev/ptyr*. NOTE: all files can be listed with the '*ls-/*' if SHOWFLAG is enables. Will hide any files/directories with the names, *ptyr*, *hack.dir*, and *W4r3z*.
- ps, top, pidof – these programs will not display the intruders processes
- netstat -- will not display traffic from or to specified IP addresses, user-ids, or ports
- killall – will not kill the intruders hidden processes
- ifconfig – will not display the PROMISC flag when sniffer is running
- crontab – will hide the crackers entries- the hidden crontab entry is in the */dev/hda02* by default
- tcpd – will not log connections listed in the configuration file
- syslogd -- will not log connections listed in the configuration file

2 - Trojaned programs with backdoors:

- chfn – new full name enter password will drop rootshell
- chsh – new shell enter password will drop rootshell
- passwd – rootshell if is entered as current password
- login – will allow the cracker to log in under any username with the rootkit password (*satori*)—also if root is refused username (*rewt*) will work and will disable the history logging

3 - Trojaned network daemons:

- inetd – rootshell listening on port 5002. the rootkit password most be entered in as the first line (*satori*)
- rshd – the username is the rootkit password, a root shell is bound to the port [rsh (hostname) –l (rootkit password)]

4 - Utilities:

- FIX – replaces and fixes timestamp/checksum information on files
- linsniffer – a packet sniffer
- sniffchk – checks to make sure the sniffer alive
- wted – wtmp/utmp editor
- z2 – erases entries in the wtmp/utmp/lastlog entries for a username-will only null the entry
- bindshell – binds a rootshell to a port (31337) by default

What do I do now?

Ways to keep your network somewhat safe from rootkits is to use encryption and not transmitting reusable clear-text passwords over the network or by using what is know as one-time-passwords. Using encryption or one time passwords will not keep you entirely safe from rootkits however, encryption and one time passwords will help in ensuring that an intruder is unable to capture additional usernames and passwords for the network.

There are also several utilities such as *Tripwire* and other Tripwire-like programs that provide system integrity checks. *Tripwire* is a tool that checks to see what has changed on your system. Unlike other intrusion detection or security software, *Tripwire* like programs monitor all changes *Tripwire* does not just look for "attack signatures". *Tripwire* first creates a database that monitors the binary signature, size, expected change of size, etc. *Tripwire* includes four cryptographic checksums of the content of each file that *Tripwire* uses to create the original database. When the software performs a system check, it will compare the system with the baseline of original database. If a modification has occurred *Tripwire* will alert the System Manager Station by a violation alert and the System Administrator by an email, the violation alert will show what files/directories were modified, added, or deleted. If the questioned alert is an authorized install, upgrade, or patch you can re-baseline *Tripwire* to ensure that the false positive

does not happen again. *Tripwire* is also configurable enough to allow a user to define what files or folders that the user would like to be monitored. For example, a user could set up *Tripwire* to monitor the system binaries or files/directories that should have minimal changes but not monitor the system log since the log will be ever changing. However, a good copy of the database must be kept in a secure location to ensure that an intruder does not modify it. Also keeping your database on physically read-only material such as a disk or CD-ROM is a good idea the database will then become an authoritative reference for integrity of the system.

Sources

NSA Information Systems Security Organization. "NSA Glossary of Terms Used in Security and Intrusion Detection." April 1998. http://www.sans.org/newlook/resources/glossary.htm.

CERT Coordination Center. "Steps for Recovering from a UNIX or NT System Compromise." April 17, 2000. http://www.cert.org/tech_tips/root_compromise.html.

CERT Coordination Center. "Ongoing Network Monitoring Attacks." revised on September 19, 1997. http://www.cert.org/advisories/CA-1994-01.html.

David O'Brian. "Recognizing and Recovering from Rootkit Attacks." November 1996. http://www.cs.wright.edu/people/faculty/pmateti/Courses/499/Fortification/obrien.html

David Brumley. "Rootkits – How Intruders Hide" http://www.theorygroup.com/Theory/rootkits.html

Tyler. "My experience with being cracked" Jul 19, 2000 http://www.rootprompt.org/article.php3?article=678

Lord Somer. "lrk4.shad.tar.gz" http://packetstorm.securify.com/UNIX/penetration/rootkits/lrk4.shad.tar.gz

TRIPWIRE CORP. "The Tripwire HQ Connector Bundle: Tripwire's Integrity Assessment Software Can Now Communicate Across Your Network" http://www.tripwire.com/products/connector.cfml?