Global Information Assurance Certification Paper

Barbara J. Evans
Certification and Assignment: GSEC Version 1.4b
Title of Paper: RootKits

Abstract

Usually the first sign that a system has been compromised is simple anomalies in the behavior of the system. For instance the core utilities may be have differently, a command-line such as netstat or ps that has been used time and time again without a problem may now begin to generate error messages. A change in bandwith-usage patterns is also a huge red flag. These interruptions in your nicely running network are due to attacks by programmers commonly known as hackers. A hacker is a clever programmer who attempts to break into computer systems. Hacker use after they have already gained access to a network. A rootkit is a compilation of programs that hackers use to cover up intrusion and obtain administrator-level access to a computer or computer network. A hacker must first obtain entry level access to the system through a known vulnerability or cracking a password. The hacker then installs a rootkit on a computer. Users ids and other passwords are obtained by the rootkit giving the hacker access to privileged or root access. Rootkits consist of different utilities that accomplish many things including monitoring key strokes and traffic, creating a backdoor into the system for hacker use, altercation of file logs, avoidance of detection through altercation of existing system tools, and attacks on other computers in the network. A rootkit generally consists of network sniffers, log-cleaning scripts, trojaned replacements of core system utilities and back doors such as inetd insertions. A sniffer is an application that is used to gain passwords and listen for sensitive information. Sniffers arouse out of the need to debug networking problems. However, they are now mostly used by hackers during an attack on your network. The main purpose of a rootkit is to allow intruders access to the infected network at a later time without being detected. The documentation of rootkits dates back to the early 90s. The primary targets were first Sun and Linux operating systems. Today rootkits are becoming harder to detect daily and are readily available for almost all operating systems.

Definition

Because rootkits are designed to hide the presence of an attacker it is important to understand how a rootkit functions. When a rootkit is installed it is able to overwrite many commands that are used on a daily basis. This is known as “Trojanizing” a command. For example commands such as ls, login, or killall. By overwriting such commands, the intruders are hidden from the administrators. Because rootkits are designed to hide the presence of an intruder the command will still function as normal, however, it will also complete the hackers added application. Hackers may use a common command such as find to hide their actions. The hacker overrides the find command so that whenever this common

application is ran it will hide from view all files installed by the rootkit. The following are commands that are commonly used by hackers:

lsof: This command is used to open files. This command may be altered by a rootkit so that files or processes can be hidden.

login: This command is used when a user is signing onto the system. The login command can be altered by a rootkit, it will then record all users and passwords that are inputted into the system. This documentation can then be accessed by the hackers to gain access to privileged areas of the network.

netstat: This is a useful tool for displaying information about current network connections, interface statistics, and routing tables. Hackers use Netstat to hide the connections that are made by the intruder in the system.

Ifconfig: This tool is useful for displaying and configuring information about network interfaces. The network may be placed in promiscuous mode if a sniffer is installed and running. By placing an interface in promiscuous mode the network interface is enabled to intercept and read packets on the network. Ifconfig is commonly altered to conceal the evidence of an interface in promiscuous mode thus hiding the presence of a sniffer or password grabber.

killall: A command used to stop processes. Hackers use the altercation of this command to prevent administrators from halting processes that have been installed by the rootkits.

inetd(xinetd): Is a super server that is designed to start programs that provide Internet services. (x)inetd then generates the appropriate server to accept the connections. Many rootkits are able to add their applications onto the configurations file causing the rootkit services to be generated when a specific port is accessed.

find: This useful command is used to find files inside a directory hierarchy. Because of the altercation of the find command network administrators have difficulty searching for programs that are known to be installed by rootkits.

Rootkits do not always operate by overriding a command. Some rootkits are called LKMs or loadable kernel modules. These rootkits offer a very severe threat to the system. LKMs are a mechanism for adding functionality to an operating-system kernel on the fly without requiring a kernel recompilation. Another words these rootkits operate by deleting old programs or processes and replacing them with a changed behavior of the same program or processes. This does not modify the actual command but modifies the behavior of a certain command. Even if you reboot a system that is infected by an LKM, the LKM process will reload it during boot-up just like any other kernel module.

Basic Prevention Techniques

Any good security system begins with prevention. The basic security plan should include a virtual private network (VPN), firewalls, updated vendor patches, and updated applications. A secure network can be obtained by the implantation of basic security guidelines. A VPN is a way of using the internet or other public telecommunication infrastructure to provide remote user with secure access to the internet. A virtual private network provides the same capabilities as a virtual private network at a fraction of the cost. A VPN uses a public telecommunication infrastructure to setup private offices through a series of security procedures. The use of firewalls on all networks is a must when internet access is granted. A firewall is a set of related programs located at a gateway network server. A firewall protects the resources of a private network from users on other networks. Although a firewall is not the only way to prevent intruders it is a good and solid beginning. It is pertinent for a system administrator to know exactly what applications are running on all systems. Administrators should run systematic checks to ensure that any unneeded services are turned off and deleted. Unauthorized applications will be detected and removed during these systematic checks. System administrators should grant access on an as needed basis. Users should not be granted access to information on programs that are not needed for job performance. To ensure that all data transmitted across wide area network (WAN) is encrypted administrators must use VPNs. A secure shell is a unix based command interface and protocol for securely getting access to a remote computer. SSH commands are encrypted and secure in several ways. Both the client and the server connection are authenticated using a digital certificate and passwords are protected by being encrypted. Through the use of Secure Shell (SSH) all transmitted data will be encrypted including usernames and passwords. Administrators need to stay current on all vendor releases. To resolve security issues all applications must be kept up to date. Many vulnerabilities are published in hacker publications and websites. Hackers are able to use this information to attack your network. By ensuring that all patches are up to date administrators will resolve all known vulnerabilities making it more difficult for hackers to breach the system. Administrators should also make use of an intrusion based detection system (IDS). It is vital that administrators are notified when an unauthorized attempt to connect to the network is made by making use of one of these systems the IDS will inform an administrator immediately. By using an up to date version of a program such a Logwatch administrators can gain an upper hand in basic system monitoring. Because most system activities are logged it is vital for administrators to monitor system activity for unusual activity.

File security must be ensured to prevent rootkit attacks. By using an immutable file the system administrator ensures that the file may not be changed, deleted,

renamed, or altered in any way. To set the immutable flag on a file, use the "chattr" command found in most Linux distributions.

chattr +i <file>: sets the flag
chattr -i <file>: unsets flag
lsattr <file>: displays attributes set to a file

By setting the flag on a file to immutable many common rootkits are caused to fail. Because the rootkit attempts to change the file it fails on immutable files that cannot be changed. Setting a file flag to immutable, however, may not stop LKM rootkits, because the actual kernel is not able to be changed to immutable. The common commands given at the start of the paper are an effective beginning point when deciding which file commands to change to immutable. Because immutable flags are easily changed they should not be relied upon as the only defense against rootkits. Immutable flags can be changed by the root which means that if a hacker can gain access to a computer on the network they would be able to locate and change the status of an immutable file.

System administrators must implement basic security measures to ensure the health of the network. After sufficient security measures are in place it is vital for administrators to implement detection and monitoring practices on the system.

Detection and Monitoring

It is an extraordinarily difficult task to detect something that has been designed to be concealed. Applications have been created to assist system administrators in the monitoring of all system activities. These applications are also able to assist in the detection of rootkits.
An application that is useful to system administrators in the monitoring of systems is Chkrootkit. Chkrootkit is written and maintained by Neslon Murilo. Chkrootkit has been tested on Linux 2.0.x, 2.2.x, and 2.4.x. It is a shell script that checks the system for binaries for rootkit modification as well as having the ability to detect well-known LKM rootkits. Using the following command files, chkrootkit searches for common files and directories that rootkits place on the system. Awk, cut, echo, egrep, find, head, id, ls, netstat, ps, strings, sed, and uname. Chkrootkit can also checks to ensure that the network interfaces are not in promiscuous mode. Because promiscuous mode allows a network device to intercept and read each network packet that arrives this mode allows the sniffer program all packets for analysis. Chkrootkitis also able to check for hidden processes. By checking the output of ps with the /proc directory chkrootkit is able to check for hidden processes such as backdoors and sniffers. The use of infected files by chkrootkit would defeat the purpose of the entire application.
A superior option of chkrootkit is that it allows command files to be used from alternate places such as a compact disk or floppy disk. By setting these files as write protected and setting the immutable flag it ensures that uninfected files are being used by chkrootkit.

By using the LSMOD utility the administrator is able to show information about all loaded modules. As an administrator you should become familiar with the output of this command. The output includes name, size, use count, and list of referring modules. To guard against LKM rootkit's a system administrator should keep track of all modules that are loaded into the kernel. System administrators should also remember that LSMOD can be altered by the rootkit to hide modules or output generated by the rootkit. Another command that is able to list what modules are loaded into the kernel is Cat. Using cat the same information that is displayed in LSMOD will be displayed.

Another application that is vital to system administrators is one that monitors the integrity of files. Tripwire is a utility used to monitor the integrity of files. Basically Tripwire is a tool that checks to see what has changed on your system. Tripwire is known as a detection tool. Using this program or others similar to it the database files and directory attributes are protected by password. MD5 is used to check for changes and to analyze the integrity of the files against the current files and directories. MD5 sums are a hash function that transforms a string of data of any length into a shorter fixed-length value. It is claimed that no two strings of data will have the same MD5 value.

For example,

the MD5 value for the command ls is dc1961b6ce3ff6dfe2c898603f4985.

For the command netstat the value is 30286974e55bb9f9e82f93cc44c39492.

Notice that the values are completely different for the two commands. If files are being monitored any modification would be recognized by Tripwire and the system administrator would be notified. Tripwire is not only able to tell the administrator that a change has been made. It can also tell who made the change, as well as what, when, and where the change was made.
This will assist the administrator in the researching of this incident.
Tripwire and other programs like it are highly used and very much approved of in the security industry.

Very similar to Tripwire is Redhat Package Manager (rpm). RPM can be used to verify the checksum of installed applications. Every time an RPM package is installed the package manager puts information about the package and the individuals files the package contains into a central RPM database.

Although not foolproof, this information can be used for performing quick and easy system audits. This should not be relied upon if the system is given a clean bill of health, however, if a problem is found there is no doubt that an error is present. In addition rpm can also check for other file discrepancies. This could include anything from permissions to file size. By using the command -V in rpm

the signature is checked on any installed rpm package. Using the -Va option will check all installed packages. Remember that RPM can only be trusted when a negative results. This is another reason that I recommend using all of the applications in this paper. RPM is very effective when used in combination with chkrootkit.

Running the command “rpm -V util-Linux” on a test system returned the following results:

…….T c /etc/fdprm
…….T c /etc/pam.d/ chfn
…….T c /etc/pam.d/ chsh
s.5....T c /etc/pam.d/login

By using the excerpt below, you can interpret the above results. It was determined that the mTime (T) was changed on all files. The mTime is the file modification date and time. The MD5 value of login and the file size were also changed. Using this information the system administrator would be able to further investigate the incident to determine why the files had been changed.

From the RPM man page:

S file Size differs
M Mode differs (includes permissions and file type)
5 MD5 sum differs
D Device major/minor number mis-match
L read Link (2) patch mis-match
U User ownership differs
G Group ownership differs
T mTime differs

A program was written by a soft project group called Kernel Security Therapy Anti Trolls (KSTAT). This program assists in the detection of LKM rootkits. KSTAT has been written for Linux 2.2.x and 2.4.x kernels. KSTAT works by checking the memory for information about the host (including LKMS). KSTAT provides many of the same options for detecing rootkits as the other applications, however, KSTAT also has the ability to run option -s. This will display information about the system call table.

All of the above applications are necessary for a system administrator. Using a rookit hackers are able to reek havoc upon your system. It is very important that the necessary steps be taken in order to detect rootkits in your system. The above applications should be run automatically at least every night. The advantage to using these automated programs are numerous. Not only are companies able to save man hours because they do not have to pay IT techs,

but the programs save time and effort by automatically identifying the problem and giving a good start on how to fix it. With over $118 million spent on computer forensics and other incident response services it is obvious that companies are beginning to see the value of system security.

Removal of a Rootkit

When removing a rootkit it is vital to first evaluate the situation. Before a rootkit is removed from the system the rootkit or LKM kernel should be identified to the best ability of the system administrator or IT tech in charge of removal. The effects of removing the rootkit should be thoroughly thought through before removing the rootkit so that no farther damage is done to the network. The computer needs to de disconnected from the network so that outside access is no longer possible and to prevent further damage to the network. After the situation has been assessed a plan of action should be confirmed. If possible the hard drive should be taken out and replaced. Having either the original hard drive or a copy of manipulated files is important for many reasons. The infected system will be a vital part of any legal proceedings as well as an important training tool to protect against future attacks. By utilizing all of the applications mentioned in this paper it should be possible to locate the rootkit and all files which were infected. However, for the applications to work properly the system administrator must have already set up the applications properly. This ensures that chkrootkit and all other applications are running using uninfected files otherwise these applications will not perform correctly and the removal will fail.

Cleanup

After the initial attack a network may never be the same. A complete clean up and system recovery should be performed to prevent future attacks. It must be assumed that all information on the network during the time of the attack has been exposed. Locating all Trojan versions of the standard system can be difficult. A system administrator should not trust any system utilities until they have been restored from a safe source such as a floppy disk or distribution media. Unless the system administrator is positive that back ups were made before the security breach, backups should not be used. There would be too great of a risk of re-installing an infected file. After the system is cleaned up and there is no sign of the rootkit determine how access was gained. A re-evaluation of all security measures will be necessary. This will help to prevent another security problem in the future.

Conclusion

In conclusion it is a good idea for system administrators to secure their system

from attackers. However it is also important to realize that there is a viable threat at all times and the effects of an attack must be known. This requires pre-planning when completing and work within the system. If you are adding a new system completing a back up or just installing a new program there are many things to keep in mind to ensure the safety of the system. There are many resources for system administrators to utilize, including many programs with automated features. As long as regular audits are performed you will be able to maintain as healthy network free of intruders.

References

Altunergil, O. (2001, December14). Understanding Rootkits. Retrieved from the World Wide Web on March 21, 2003: http://linux.oreillynet.com/pub/a/linux/2001/12/14/rootkit.html

Altunergil, O. (2002, February 07). Scanning for Rootkits. Retrieved from the World Wide Web on March 21, 2003. http://linux.oreillynet.com/pub/a/linux/2002/02/07/rootkits.html

Brumley, D. (1999, September). Rootkits - How Intruders Hide. Retrieved from the World Wide Web on February 20, 2003. http://www.theorygroup.com/Theory/rootkits.html

Steps for Recovering from a Unix or NT System Compromise. (200, April 17). Retrieved from the World Wide Wed on February 15, 2003. http://www.cert.org/tech_tips/root_compromise.html

Daviel, Andrew (2002, August 18) Rkdet - Rootkit detector for Linux. Retrieved from the World Wide Web on March 01, 2003. http://www.vancouver-webpages.com/rkdet/

Drake, J. (2001, October 12). How to tell if your Linux box has been cracked. Retrieved from the World Wide Web on February 27, 2003. http://www.linuxworld.com/site-stories/2001/1012.cracked.html

Galitz, G. (2001). Rootkits: Hiding a Successful System Compromise. Retrieved from the World Wide Web on March 11, 2003. http://www.iwar.org.uk/comsec/resources/root-berkeley/rootkit.htm

Garfinkel, Simson & Spafford, Gene. Practical Unix & Internet Security. (1996, April) O'Reilly & Associates Inc.

Kleimola, Johannes. (1999, March 24) Linux Rootkit IV Retrieved from the World Wide Web on March 01, 2003. http://www.hut.fi/~jjkleimo/kurssit/tik110452/experiment_rootkit.html

McClure, Stuart, Scambray, Joel, & Kurtz, George. Hacking Exposed Network Security Secrets & Solutions. (1999) Osborne/McGraw-Hill.

Miller, T. (2000). Detecting LoadableKernel Modules. Retrieved from the World Wide Web on March 11, 2003. http://www.incident-response.orgLKM.htm

O'Brien, David. (1996, November) Recognizing and Recovering from Rootkit

Attacks. Retrieved from the World Wide Web on February 27, 2003. http://www.cs.wright.edu/people/faculty/pmateti/Courses/499/Fortification/obrien.html

Project: Tripwire: Summary. (2002, February 28). Retrieved from the World Wide Web on February 27, 2003. http://sourceforge.net/projects/tripwire

Prosise, c. and Shah, S, (2001, January 25). At the root of rootkits. Retrieved from the World Wide Web on March 28, 2003. http://www.builder.com/webbuilding/0-7532-8-4561014-1.html?tag=st.bl.7532-84561014

Whatis.com. (2001, April 25) Rootkit. Retrieved from the World Wide Web on March 14, 2003. http://www.whatis.com

Last Updated: May 22nd, 2017

Upcoming Training

SANS Stockholm 2017	Stockholm, Sweden	May 29, 2017 - Jun 03, 2017	Live Event
Community SANS Ottawa SEC401	Ottawa, ON	Jun 05, 2017 - Jun 10, 2017	Community SANS
SANS San Francisco Summer 2017	San Francisco, CA	Jun 05, 2017 - Jun 10, 2017	Live Event
Security Operations Center Summit & Training	Washington, DC	Jun 05, 2017 - Jun 12, 2017	Live Event
SANS Houston 2017	Houston, TX	Jun 05, 2017 - Jun 10, 2017	Live Event
SANS Charlotte 2017	Charlotte, NC	Jun 12, 2017 - Jun 17, 2017	Live Event
SANS Rocky Mountain 2017 - SEC401: Security Essentials Bootcamp Style	Denver, CO	Jun 12, 2017 - Jun 17, 2017	vLive
SANS Secure Europe 2017	Amsterdam, Netherlands	Jun 12, 2017 - Jun 20, 2017	Live Event
Community SANS Portland SEC401	Portland, OR	Jun 12, 2017 - Jun 17, 2017	Community SANS
SANS Rocky Mountain 2017	Denver, CO	Jun 12, 2017 - Jun 17, 2017	Live Event
SANS Minneapolis 2017	Minneapolis, MN	Jun 19, 2017 - Jun 24, 2017	Live Event
SANS Columbia, MD 2017	Columbia, MD	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS Cyber Defence Canberra 2017	Canberra, Australia	Jun 26, 2017 - Jul 08, 2017	Live Event
SANS Paris 2017	Paris, France	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS London July 2017	London, United Kingdom	Jul 03, 2017 - Jul 08, 2017	Live Event
Cyber Defence Japan 2017	Tokyo, Japan	Jul 05, 2017 - Jul 15, 2017	Live Event
SANS Cyber Defence Singapore 2017	Singapore, Singapore	Jul 10, 2017 - Jul 15, 2017	Live Event
Community SANS Minneapolis SEC401	Minneapolis, MN	Jul 10, 2017 - Jul 15, 2017	Community SANS
SANS Los Angeles - Long Beach 2017	Long Beach, CA	Jul 10, 2017 - Jul 15, 2017	Live Event
SANS Munich Summer 2017	Munich, Germany	Jul 10, 2017 - Jul 15, 2017	Live Event
Community SANS Phoenix SEC401	Phoenix, AZ	Jul 10, 2017 - Jul 15, 2017	Community SANS
Mentor Session - SEC401	Ventura, CA	Jul 12, 2017 - Sep 13, 2017	Mentor
Mentor Session - SEC401	Macon, GA	Jul 12, 2017 - Aug 23, 2017	Mentor
Community SANS Colorado Springs SEC401	Colorado Springs, CO	Jul 17, 2017 - Jul 22, 2017	Community SANS
Community SANS Atlanta SEC401	Atlanta, GA	Jul 17, 2017 - Jul 22, 2017	Community SANS
SANSFIRE 2017	Washington, DC	Jul 22, 2017 - Jul 29, 2017	Live Event
SANSFIRE 2017 - SEC401: Security Essentials Bootcamp Style	Washington, DC	Jul 24, 2017 - Jul 29, 2017	vLive
Community SANS Charleston SEC401	Charleston, SC	Jul 24, 2017 - Jul 29, 2017	Community SANS
Community SANS Fort Lauderdale SEC401	Fort Lauderdale, FL	Jul 31, 2017 - Aug 05, 2017	Community SANS
SANS San Antonio 2017	San Antonio, TX	Aug 06, 2017 - Aug 11, 2017	Live Event
SANS Prague 2017	Prague, Czech Republic	Aug 07, 2017 - Aug 12, 2017	Live Event