Rootkits, Trojans, Backdoors and New Developments

Hugo E. Decloedt, Renier van Heerden
University of Pretoria, Pretoria, South Africa
hugoD@renier.co.za
rvanheerden@csir.co.za

Abstract: This paper gives an overview of the history and new developments with Rootkits, Trojans and backdoors. The paper also looks at the different types of rootkits that exist, how to use a rootkit, and methods for detecting rootkits. Backdoors, Trojans and detection methods are investigated. Methods for detecting and removing malicious software are also looked at. Current and future developments in rootkits, Trojans and backdoors are evaluated.

Keywords: Rootkit, Trojan, Backdoor

1. Introduction

Rootkits, Trojans and backdoors are software programs that grant a user access to a computer with or without the prior knowledge and consent of the computer's owner. Rootkits as we know them now came into being sometime during the mid 1990's [1]. There are three main classes of rootkits available today. These are binary, kernel and library kits. A backdoor is a secret entry into a program that allows someone that is aware of the backdoor to gain access without going through the usual security procedures [2]. The first backdoor program which enabled any user backdoor access to a victim's computer was Sub7 created in 1990 [3]. These backdoors have been used legitimately for many years by programmers to debug and test programs. This is usually preferred when a programmer is developing a software application that requires authentication using a lengthy procedure in order to run and test the software. These backdoors become a threat once dishonest programmers discover and use them to gain unauthorised access to the software.

A Trojan horse is useful or apparently useful program or command procedure containing hidden code that, when invoked, performs some unwanted or harmful function [2]. The first Trojan horse called ANIMAL was written by John Walker in 1975 [3]. Trojan horse programs can be used to accomplish functions indirectly that would be impossible to accomplish directly for an unauthorised user. Another application of a Trojan horse could be data destruction. While a program appears to be performing a legitimate useful function, it might be performing malicious tasks in the background, such as deleting files.

In Section 2, the function, types, usage and current state of rootkits are discussed. In Section 3 the same topics with backdoors are discussed and Section 4 covers Trojans.

2. Rootkits

The functions of a rootkit are to hide processes, files, logins and logs. A rootkit may also include program code that intercepts data between the computer and a terminal or network connections. Trojans and backdoors are sometimes also included with a rootkit, thus enabling access to the computer.

The three types of rootkits are:

- Library rootkits.
- Application rootkits.
- Kernel rootkits.

Library rootkits cannot be located using a file search or by checking which applications are running in the Task Manager in Microsoft Windows [4]. Rootkits want to survive a boot, thus they are usually located in the following:

- Registry keys.
- Startup files.
- Add-on to an existing application.
- Patching binaries on hard drive.
- Using a custom master boot record (MBR).

All rootkits want to stay hidden and need to be executed [5].

2.1 Types of rootkits

2.1.1 Library rootkits

Library rootkits have similar goals to that of loadable kernel modules (LKM) [1]. These rootkits use different methods to elude detection. For example “T0rn 8” uses a special system library to relay the process information from the kernel space to user space utilities. It is reasonably easy to edit the main system library to switch the data before it is sent to the kernel.

2.1.2 Application rootkits

These rootkits usually operate by replacing normal application binaries with Trojan fakes. Another way used is to inject code or make use of hooks or patches [1]. The first rootkits used replaced critical system libraries such as /bin/login and network daemons. The executables were used to perform tasks for the attacker such as hiding malicious processes. After breaking in, attacker copies the rootkit on the compromised system. The copy is initiated using an included installation script. This script overwrites the current binaries with the new ones.

Common binaries that are overwritten include but are not limited to the following:

- The binary handling authentication in Linux operating systems (OS), login.
- Internet services daemon running on Linux OS that handles internet services, inetd.
- Secure shell (SSH) daemon that handles all SSH connections, sshd.

The binaries that are compromised can provide remote access via a magic password. It can also provide local access. Process hiding is also possible by compromising the /bin/ps binary. A modified syslogd daemon will prevent certain processes from logging messages to system logs and remote log servers. Netstat can be compromised and modified to hide certain network connections. It is also possible for rootkits to hide files as well as user activities.

2.1.3 Kernel rootkits

Most Linux systems differentiate between kernel mode and user mode [1]. The exact date of when LKM came into use is unknown. LKM first came into use as malicious kernel modules. Kernel level rootkits usually add code or replace a section of kernel code with modified code in order to hide its existence. Windows will tag specific sections of memory that are assigned to either kernel mode or user mode. The only problem with this is that memory tagged in kernel mode is not protected from other processes that are also running in kernel mode. This makes kernel level rootkits extremely dangerous. Kernel level rootkits are almost impossible to detect due to them not altering the operation and behavior of the computer significantly enough to alert the user of their existence. Current versions of Windows also support kernel mode as well as user mode [6].

2.2 Usage

The steps of using a rootkit [7] follow:

- Discover vulnerability on system
- Exploit vulnerability to gain access
- Collect information and install backdoors

The steps taken to attack a vulnerable system using a rootkit can be broken down into four phases. The first phase is where the attacker discovers a vulnerable system. This is usually accomplished using a port scanner. An example of a popular port scanner is Nmap, which is short for network mapper.

The second phase is to exploit the vulnerability in the victim’s computer in order to gain access. The third phase is to install backdoors and other malicious software programs on the victim’s computer in order to gain a stronger foothold on the victim’s computer. The fourth and final phase of the attack is to achieve the final goal of the attack. This may be to steal information, disseminate spam or to launch a denial of service (DoS) attack on another computer system over a network. Eventually the compromise will be detected and a suitable response will be initiated.

2.3 Detecting rootkits

Most early rootkits worked by modifying the system binaries on UNIX machines. This can easily be detected by doing integrity based checks. While this method was extremely effective in detecting early rootkits, eventually the rootkits started targeting process and kernel memory. Another method of detecting rootkits is to compare the binary files stored in memory to those that are stored on the hard disk drive (HDD).

Signature based detection has been the classic approach to detecting malware in computer systems. Whenever a new virus or worm is detected by antivirus companies such Symantec they scan the virus program code for a unique sequence of bits. This unique sequence of bits then becomes the signature for that specific malware. Signature based rootkit detection has disadvantages. Two of these disadvantages are that rootkits might disable the antivirus software package on the victim’s computer before installing itself on the computer. The second disadvantage might be that the rootkit is installed before the antivirus program and that the process is hidden from the antivirus program. Also another important disadvantage is that if the antivirus software package does not contain a unique signature for the rootkit, it will not be able to detect the rootkit.

Another popular and efficient way to detect rootkits is hook detection [7]. One of the most popular hook intrusion detection methods is to use virtual intruder capture engine (VICE).

Another method of detecting rootkits is to use cross view detection [7]. The idea of cross view detection is very simple. Information is requested using two different ways. If the information retrieved differs then one of the retrieval methods have been compromised. Cross view detection is usually done from a high level and then compared to a low level’s retrieved data. This method does not reveal the exact location of a rootkit, although it does reveal that there is a rootkit present. Rootkit revealer [8] is a popular software tool capable of detecting rootkits using this method.

The best approach in detecting rootkits is to combine the above mentioned methods into one single package in order to detect rootkits. For serious rootkit detection a hardware based approach might be a better option. Copilot [9] is a hardware peripheral component interconnect (PCI) module that has its own independent processor that scans for rootkits. Due to the fact that it is independent from the computer’s kernel and processes, it cannot be compromised.

2.4 Current state of rootkits

Modern rootkits today are injected into the privileged mode of the processor and thereby have full access to the operating system and hardware [10]. Common features of today’s rootkits include the hiding of processes and files, key logging and hidden data transfers via network. Detecting rootkits when they almost completely take over the system is an extremely difficult task. To prevent detection some rootkits hide themselves not only in the random access memory (RAM) or on the hard disk drive (HDD), but they also hide themselves in the Flash memory of peripheral devices like advanced configuration and power interface – basic input/output system (ACPI-BIOS) [11] or peripheral component interconnect (PCI) cards [12]. If the system is infected a simple clean-up of the system will not remove the rootkit since it is not residing on the HDD. While rootkits are currently mostly used on workstations or servers they can be used to subvert the Trusted Computing Base of an embedded system. An example of this is securing the media streaming in high definition television (HDTV) which is a hot topic [10]. Past experience has shown that securing Pay TV is an extremely hard task. Attackers have found many ways to circumvent current security measures which forced TV stations and providers to issue new smart cards which in turn resulted in a new round of attacks. Field programmable gate array (FPGA) microcontrollers will enable providers to update the security “hardware” in much the same way that they update the software today. These updates could then include new hashing algorithms, encryption algorithms or random number generators. However an attacker could change the configuration of the FPGA for his own purposes. For example, a changed FPGA could stream unencrypted media to an external device or circumvent the authentication system.

Although rootkits have been around for a while, experts say that hackers are using them considerably more than they used to in the past [13].

3. Backdoors

Backdoors are a method of bypassing authentication and gaining user or administrator rights on a computer without going through the regular authentication protocol. The backdoor could be via an installed program such as Back Orifice or through backdoors that were left by the software's developers. Popular backdoor programs from the 90's used for mischief were Back Orifice, Netbus and Sub7. Some of the latest popular backdoors are: DsBot, Aimot, Egg Drop, Hupigon, Mo Sucker and VanBot.

3.1 Back Orifice

Back Orifice is the most common of the three backdoor methods and is considered the most lethal. Back Orifice was created by the Cult of the Dead Cow computer hacker organisation. The program debuted on the 1st of August at DEFCON 6. According to the hacker group, the purpose of the software was to demonstrate to Microsoft the vulnerabilities that their operating system still had [14]. Back Orifice is marketed as a legitimate network administrator tool even though it is possible to hide it from the system process view. Back Orifice has a very user friendly graphical user interface (GUI) and was used by a lot of script kiddies. Two sequel versions have succeeded the original version, Back Orifice 2000 and Deep Back Orifice.

3.2 Netbus

Netbus was created at around the same time as Back Orifice. The software was originally designed to prank family or friends without being too malicious [15]. Operations that could be executed on the victim's computer remotely from another computer were:

- Opening the compact disc read only memory (CD-ROM).
- Starting the CD timer.
- Show an image on the remote computer's screen.
- Swapping the mouse buttons from left to right and vice versa.
- Execute a program.
- Send a message to the remote computer.
- Dumping a screenshot of the remote computer's desktop to the Netbus client.

3.3 Current state of rootkits

The problem of backdoor intrusion into any corporate network using a local area network (LAN) or wireless local area network (WLAN) is a real threat [16]. Securing a wired network only requires securing the physical network infrastructure, but securing a wireless network requires more security measures in order to obtain a high level of security. Wireless networks are common in enterprise networks today. Common wireless threats include the following:

3.3.1 Rogue access points

This threat is the most common, as well as the most dangerous of all the wireless threats [16]. The access point is usually low cost and brought in by an employee requiring wireless access. The access usually operates using default security settings and default passwords.

3.3.2 Misconfigured access points

If the access point is incorrectly configured, it might go unnoticed [16]. For example the security settings are disabled but employees and users are still able to connect to the network.

3.3.3 Client mis-associations

Most laptops are shipped with an embedded wireless networking device [16]. Employees might not disable these or they might attempt to connect to previously connected wireless networks creating a backdoor security risk. Software backdoors can also be obtained when downloading and installing shareware. These backdoors might attempt to change a computer's desktop, hijack an internet browser, change system files and can do this without your prior knowledge.

3.4 Prevention

There are numerous ways that a computer can get infected with a backdoor. Backdoors can be bundled with shareware or other downloadable software. Backdoor access to a computer system can be prevented by using a firewall. A good anti-spyware software package is required to scan for spyware that may contain backdoors. Some other useful security practices on top of installing an anti-spyware software package are:

- Obtaining operating system security updates and keeping your computer's operating system up to date.
- Updating anti-spyware software package at regular intervals to ensure up to date definitions required to detect spyware.
- Scan for spyware using an anti-spyware software package.

The general symptoms of a computer that has a backdoor(s) installed are system setting changes, excessive popup windows and slow performance.

4. Trojans

Trojan horses are one of the easiest weapons or tools that hackers in particular script kiddies can use to cause havoc on the internet [17]. A computer Trojan horse is a malicious software program that runs disguised as a software valuable program. A Trojan is typically disguised as an appealing message or software program similarly to the Trojan horses that concealed Greek warriors and resulted in the downfall of Troy. Once the Trojan horse is installed on a computer, anyone can connect to it using a client software program [17]. Port scanners are available that can locate computers who have already got Trojan horses installed on them. Anyone can utilise the Trojan horse on a victim's computer even if he or she did not install it.

What attackers look for:

- Credit card information.
- Any accounting information such as e-mail passwords,
- Dial-up passwords and web service passwords.
- Confidential documents containing sensitive data.
- E-mail addresses.
- Calendar information regarding user's whereabouts as well as schedule information.

The different types of Trojans are discussed below in detail [18].

4.1 Remote access Trojans

These are the most popular Trojans, since they provide the attacker with full control over the user's computer.

4.2 Data sending Trojans

The purpose of these Trojans is to send back data to the attacker of the user's usernames and passwords required for authentication in order to retrieve confidential information. Information includes but is not limited to credit card information as well as e-mail passwords. Some data sending Trojans log the victim's keystrokes and then send the logged data to the attacker at preset time intervals via email. These logs may contain sensitive data such as usernames and passwords of internet banking websites.

4.2.1 Destructive Trojans

The only function of these Trojans is to destroy and delete files. They can be activated remotely or locally using a time delay.

4.2.2 DoS Trojans
This type of Trojan gives the attacker the power to start a distributed denial of service (DDoS) attack if there are enough victims. WinTrinoo is an example of a popular tool for launching DDoS attacks using a Trojan.

4.2.3 Proxy Trojan
These Trojans turn a victim's computer into a proxy server. The proxy server running on the victim's computer is then open to the whole world for use. The proxy server is used to hide the attacker's internet protocol (IP) address when performing malicious tasks such as stealing credit card information.

4.3 Avoiding Trojan horses

Below are some pointers on how not to infect a computer with a Trojan horse [19]:

- Do not download files from locations which you are not 100% sure about the authenticity of the file's contents.
- Beware of hidden file extensions, the easiest way is to unhide file extensions in Windows.
- Never use the preview feature in any software package such as Microsoft Outlook.
- Never blindly type commands or visit unknown uniform resource locator (URL) addresses.

4.4 Current state of Trojans

Hardware manufacturers are increasingly outsourcing their integrated circuit (IC) fabrication work overseas due to their much lower cost structure [20]. This poses a serious security risk for critical business and military applications. Attackers can exploit this loss of control and to substitute authentic ICs and replace these with Trojan ICs. Trojans in ICs can be used to leak cryptography keys.

Trojans can also be injected into FPGAs in the hardware descriptor language (HDL) and then programmed into the FPGA [21]. These Trojans can be detected using techniques such as ECC detection.

4.5 Removing Trojans

Below is a list of options available to get rid of Trojans [19]:

- A clean re-installation of the operating system and all software packages.
- Using an antivirus software package to remove unwanted Trojan horses.
- Anti Trojan software packages can also be used, they are the ideal solution since they specialise in Trojan horse removal.

5. Conclusion

Rootkits were originally designed to gain access to a UNIX based operating system without going through the usual user authentication process. This then evolved into software used to gain access to UNIX, Microsoft and Macintosh based operating systems. Rootkits are now mostly used to gain access to a remote computer over a network for malicious activities. This gives the attacker the power to access sensitive data on the victim's computer as well as execute program code on the victim's computer. Operating systems now differentiate between kernel and user mode applications which decreases the chance of rootkits getting kernel level access to a computer. Since then user mode rootkits have become available and are also capable of doing harm to a computer.

Backdoors are either created by software loaded or patched onto the computer or it was created by software developers during the development phase. The latter is usually to bypass lengthy authentication procedures in order to test software code and program functions. Backdoors are installed onto a computer when a file is opened. These files can be anything from an e-mail attachment to a portable data file (PDF). Once the backdoor has been installed it allows remote access to the computer. Backdoors are usually included in spyware.

Trojans are files that look like normal files with an honest and useful purpose. Just like the Trojan horses fooled the people of Troy in believing that they were harmless, even though they were filled with Greek soldiers. Trojan horses have a similar goal to that of backdoors in the sense that they are used to gain unauthorised access to a computer. They have further functionality also such as being used for DDoS attacks or a proxy server to hide hackers' IP addresses from other victims and can also be used in conjunction with key loggers in order to capture sensitive confidential information such as credit card information or internet banking details. Other Trojans just delete files from the victim's computer. Rootkits, backdoors and Trojans all have one thing in common.

Rootkits gain access via the kernel. Backdoors gain access via a hardcoded username and password or a software patch enabling unauthorised access. Trojans gain access by fooling the victim's computer into executing a file that executes software code giving the attacker access to the victim's computer.

All three of these attacks are detectable by modern antivirus software packages, given that they have up to date definitions. Backdoors can be removed using backdoor scanners or anti spyware software packages. Trojans can also be removed by using specialised Trojan scanners and removing them.

References

[1] Anton Chuvakin. (2003, February) An Overview of Unix Rootkits. [Online].

www.thehackademy.net/madchat/vxdevl/avtech/An%20Overview%20of%20Unix%20Rootkits.pdf

[2] William Stallings, *Network security essentials*, 3rd ed.: Pearson Education Inc, 2009.

[3] Wikipedia. (2010, May) Timeline of computer viruses and worms. [Online].

http://en.wikipedia.org/wiki/Timeline_of_notable_computer_viruses_and_worms

[4] Dave. (2010) 5 Star Support. [Online]. http://www.5starsupport.com/tutorial/rootkits.htm

[5] Jesse D Kornblum, "Exploiting the Rootkit Paradox with, "*International Journal of Digital Evidence*" vol. 5, no. 1, pp. 1-5, 2006.

[6] Symantec. White Paper: Symantic Security Response. [Online]. http://www.symantec.com/avcenter/reference/windows.rootkit.overview.pdf

[7] Chris Ries, "Inside Windows Rootkits," VigilantMinds Inc, Pittsburg, Security Research 2006.

[8] Bryce Cogswell and Mark Russinovich. (2006) RootkitRevealer. [Online].

http://technet.microsoft.com/enus/sysinternals/bb897445.aspx

[9] Nick L Petroni, Timothy Fraser, Jesus Molina, and William A Arbaugh. (2002) Copilot - a Coprocessor-based Kernel Runtime Integrity Monitor. [Online]. http://www.usenix.org/events/sec04/tech/full_papers/petroni/petroni_html/main.html

[10] Markus Kuceral and Michael Vetter, "FPGA-Rootkits: Hiding Malicious Code inside the Hardware," University of Applied Science Regensburg, Regensburg, White paper.

[11] J Heasman. Implementing and Detecting an ACPI Rootkit. [Online]. http://www.blackhat.com/presentations/bheurope-06/bh-eu-06-Heasman.pdf

[12] J Heasman. Implementing and Detecting a PCI Rootkit. [Online].

http://www.ngssoftware.com/research/papers/Implementing_And_Detecting_A_PCI Rootkit.pdf

[13] David Geer, "Hackers Get to the Root of the Problem," *Technology News*, pp. 17-19, May 2006.

[14] Matt Richtel. (1998) Hacker Group Says Program Can Exploit Microsoft Security Hole. [Online].

http://www.nytimes.com/library/tech/98/08/cyber/articles/04hacker.html

[15] Learn Networking. (2008) Three Archaic Backdoor Trojan Programs That Still Serve Great Pranks. [Online]. http://learn-networking.com/network-security/threearchaic-backdoor-trojan-programs-that-still-serve-greatpranks

[16] Neel Diksha and Agarwal Shubham, "Backdoor Intrusion in Wireless Networks- problems and solutions," Indian Institute of Information Technology-Allahabad, Allahabad,.

[17] Jamie Crapanzano. (2003) Deconstructing SubSeven, the Trojan Horse of Choice. [Online].

http://www.sans.org/reading_room/whitepapers/malicious/deconstructing_subseven_the_trojan_horse_of_choice_953

[18] GFI Software, "The corporate threat posed by email Trojans," GFI Software, 2004.

[19] Joseph Lo. (2006) Trojan Horse Attacks. [Online]. http://irchelp.org/irchelp/security/trojan.html

[20] Dakshi Agrawal, Selcuk Baktır, Deniz Karakoyunlu, Pankaj Rohatgi, and Berk Sunar, "Trojan Detection using IC Fingerprinting," in *IEEE Symposium on Security and Privacy*, Yorktown Heights, 2007.

[21] Shant Anu Dutt and Li Li, "Trust-Based Design and Check of FPGA Circuits Using Two-Level Randomized ECC Structures," *ACM Transactions on Reconfigurable Technology and Systems*, vol. II, no. 1, March 2009.