CmpE 209: Network Security

Students:
Locsin, Mark
Do, Minh

Instructor:
Dr. Sinn, Richard

Department of Computer Engineering, San Jose State University

Team Presentation Report: Rootkit Overview

Mark David Locsin, Minh Do
Computer Engineering Department, San Jose State University, San Jose, CA 95192
E-mail: mdlocsin@hotmail.com, minhdo20@gmail.com

Abstract

Rootkits are a serious threat to today's computing. Rotkits are malware that's designed to be covert and conceal themselves in the system using various ways. Rootkits are very difficult to remove, and the only sure way to fix a compromised computer system is to reinstall the OS. If a rootkit successfully infiltrates a private network, it can result to a serious breach of security for the network as a whole.

1. Introduction

Malware is a serious concern to our daily computing. Nowadays, it is normal to use antivirus and anti-spyware scanning software every day. Aggressive malware is easy to detect by making it obvious that a computer system has been compromised. However, rootkits are a different type of malware. Rootkits are designed to be hidden and passive. Due to its nature, rootkits have a huge potential and can be a very serious threat to a computer system.

2. Rootkit Definition

Rootkits are a collection of tools/programs designed to hide the fact that a given system has been compromised. If the user and the set of antivirus software being used fail to detect that a computer system has been compromised by a rootkit, the rootkit will be able to continually operate in that system. An infected system is referred to as "a rooted system".

The rootkit can a number of things to that system. It may install other rootkits and malware into the system or other computers in that private network. It may potentially steal private information (such as credit card numbers) stored in the local system. It may record user's keystrokes, allowing the rootkit to steal password information. It may even allow a hacker to gain access to the compromised system[2].

2.1. Rootkit Origins

Rootkits originated in the UNIX world[1], taking its name from the UNIX "root" account. The first rootkits concealed themselves by replacing the binaries in the UNIX system. They also provided a backdoor to the system, allowing hackers to infiltrate the system[2].

One of the most simple and classic UNIX rootkit method is to replace the "/bin/login" binary[7]. This allows a hacker a backdoor login to the system, and when a hacker uses the backdoor, the system will not generate log files for the hacker's login. Other binaries traditionally replaced in a rooted system include the following: netstat, ifconfig, ps, ld, du, in.telnetd, chfn, chsh, inetd, passwd, top, rshd, and syslogd[7].

In Windows, Rootkits have found many ways to conceal themselves. Early rootkits used rundll32.exe to conceal their process names in windows task manager. A rootkit can create a file with an arbitrary name and a ".dll" extension. It will then use the windows system tool "rundll32.exe" to execute the ".dll" file. Even if the system detects the malware and successfully removes the ".dll" file, the rootkit may recreate the ".dll" file with a new name after some period of time or after a windows reboot. This makes the detection and removal of the malware very problematic.

2.2. Rootkit Operations

Rootkits usually work together with other malware[2]. The rootkit itself is probably installed by some other software, such as a Trojan horse.

Many rootkits are initially installed by the exploitation of security holes in internet browsers. It is therefore important to keep all of the browsers up-to-date with security patches.

Rootkits in windows have found advanced ways to conceal their presence. One way is to use direct kernel object manipulation to fool the Windows Event Viewer program[14]. It becomes very difficult for users to detect the malware's processes. Windows rootkits have also resorted to creating various instances of itself under different names. This was done to conceal itself from users, antivirus and anti-spyware scanners (and, to some extent, to fool companies who create antivirus and anti-spyware software)[4].

The rootkit is likely to install other rootkits and malware into the infected system in order to keep the system infected.

2.2. Examples of known rootkits

The following is a list of examples of a few well known rootkits and a short description of what they do:

1. HackerDefender: *"It is a user-mode rootkit that modifies several Windows and Native API functions, which allows it to hide information (files, processes, etc.) from other applications. In addition, Hacker Defender implements a backdoor and port redirector that operates through TCP ports opened by existing services. This means it is not possible to find the hidden backdoor through traditional means, like launching a remote port scan against the compromised machine."*[8]
2. Shadow Walker: *"[Allows] an attacker to hide both known and unknown malicious code from a security scanner by controlling its memory reads at the hardware level"*[6]
3. BackOrifice: Usually installed by a Trojan. It provides backdoor access to a computer system. It also conceals the fact that another user has logged in to the system or that surveillance is taking place.

It is not uncommon for programmers to give their rootkit (or malware in general) names that suggest their software is used contrary to what it is designed for.

2.3. Examples of rootkit hunters

The following is a list of examples of a few rootkit hunters:

1. Microsoft Strider GhostBuster
2. SysInternals' Rootkit Revealer
3. F-Secure's Blacklight
4. Rootkit Hunter (aka rkhunter) is an Open source GPL tool available for UNIX, Linux and OSX

3. User's concerns

When a computer system is compromised by a rootkit, there are many possible security risks:

1. The infection can spread within a private computer network.
2. The rootkit may install other malware, such as viruses and Trojans into the system.
3. The rootkit (or the malware it installs) may steal personal data, such as credit cards or passwords, from the computer system.
4. The rootkit may allow a hacker to gain access and control a computer system.

3.1. Prevention

When it comes to rootkits, prevention is truly better than cure. The best way to prevent malware is for users to be vigilant to malware. It is best to keep up with the latest security patches. This helps prevent malware from being installed into the system. Antivirus and anti-spyware software should be used regularly in order to alert the user of any possible security risks and to alert the user if the system has been compromised. Antivirus software themselves may block certain websites and, to some extent, prevent malware from being installed into the system.

3.2. Use of a sandbox

Sandbox software, such as "Sandboxie" can be used to run suspected programs or to run web browsers. A sandbox provides an environment to a program where all reads/writes to both memory and file system is intercepted. This appears transparent to the program, and any data written by the program is only available within the sandbox. When a program within the sandbox reads data, the sandbox will first check its own cache. The sandbox will only read data from the local disk if the data being accessed does not exist within the sandbox.

It is important to note that a sandbox will not prevent data from being stolen from the system, but it prevents data from being written outside the sandbox environment, therefore containing any installed malware or Trojan.

The user can clear the sandbox and should do so regularly. After the sandbox is cleaned, all data saved within the sandbox environment is gone. Any files saved or any malware installed are simply gone. A sandbox can, therefore, be a useful tool to prevent the installation of malware into a system.

3.3. Treatment of compromised systems

In many cases, complete removal of rootkits is impractical. *"Even if the nature and composition of a rootkit is known, the time and effort of a system administrator with the necessary skills or experience would be better spent re-installing the operating system from scratch."*[2]

Since rootkits are designed to be covert in the first place, there is no way to be sure that a system has been completely cleaned. It seems that re-installation of the operating system is really the best way to treat a rooted system.

4. Conclusion

Rootkits are a serious threat to our daily computing. Rootkit programmers and anti-rootkit programmers are playing a cat-and-mouse game, continually trying to find ways to outsmart each other. It is important for users to keep up with security updates and to be vigilant to prevent and detect malware.

5. References

[1] Larry Stevenson, Nancy Altholz, "Rootkits for Dummies," Wiley Publishing Inc., Indianapolis, Indiana, 2007, pp. 7-9, 101-105, 149-180

[2] Wikipedia, "Rootkit," [Online Document], Present, [cited 2009 March 12], Available HTTP: http://en.wikipedia.org/wiki/Rootkit

[3] Ric Vieler, "Professional Rootkits," Wiley Publishing Inc., Indianapolis, Indiana, 2007, pp. 3, 37-63, 139-130, 275-290

[4] CNET, "Security Watch: Root Kit 101," [Online Document] 2005 October 21, [cited 2009 March 12], Available HTTP: http://reviews.cnet.com/4520-3513_7-6361348-1.html

[5] CA Inc., "CnsMin Spyware Detail," [Online Document] 2004 August 16, [cited 2009 March 12], Available HTTP: http://www.ca.com/us/securityadvisor/pest/pest.aspx?id=453072511

[6] SecurityFocus, "Windows rootkits," [Online Document] 2006 January 5, [cited 2009 March 12], Available HTTP: http://www.securityfocus.com/infocus/1854

[7] techFAQ, "What is a Rootkit?" [Online Document], [cited 2009 March 12], Available HTTP: http://www.tech-faq.com/rootkit.shtml

[8] F-Secure, "F-Secure Virus Descriptions: HacDef," [Online Document] 22 November 2005, [cited 2009 March 12], Available HTTP: http://www.f-secure.com/v-descs/hacdef.shtml