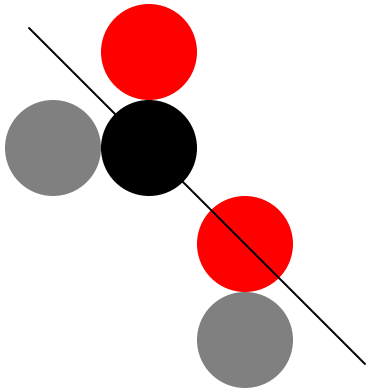

Rootkits

The new wave of invisible malware is here

by **David Sancho, TrendLabs Av-EMEA**

SUMMARY

Lately there has been a lot of discussion about rootkits and the type of threats they present. This article aims to provide a basic explanation of rootkits and how this low level technology can be used by malware developers to infiltrate computers in a way that is very difficult to detect and remove.

The name 'rootkit' is derived from the term "root", which is the name given to the superuser in the UNIX family of operating systems. [In the 1980s], hackers were known to infiltrate UNIX machines and install a program that provided a backdoor, enabling the hacker to return any time with full "root" privilages. The term 'rootkit' is now used in a similar way by modern-day researchers for Windows programs.

The operating system provides programmers with a set of basic functions that they can use to perform their everyday duties: from opening files to establishing network connections. This set of functions is called the API (Application Programmer's Interface). Rootkits intercept specific API functions in such a way that the information returned by them is untrue. So, imagine what a rootkit could do if the function it hijacks is a file function – it could easily deny the listing of any specific file or folder on the disk. It can hide itself or any other file from any program – from Windows Explorer to a simple 'dir' from the command line. Sound powerful? There's more. It can also hijack, using the same techniques, any and all access to the registry database, as well as the process list. This means that a rootkit can hide the presence of malicious programs running in the system, in addition to any registry keys it may have modified for its dubious deeds. This is why rootkits are becoming so popular among malware writers – at minimum, it provides a cloak of invisibility.

Ok, But how do they do it?

So rootkits can hijack function calls and return phony information. Ok, but how do they do that?

There are two main strategies they can use, which correspond to their broader classification: *user-mode* and *kernel-mode* rootkits.

In modern processors, programs can run in either kernel-mode or user-mode. The main difference resides in the access level they have to other programs in the memory. Kernel-mode programs can access all memory (yes, they could overwrite any other program or data and do virtually anything imaginable), while user-mode ones are confined in their own

Copyright 2005 by Trend Micro Incorporated. All rights reserved. Trend Micro and the t-ball logo are trademarks or registered trademarks of Trend Micro Incorporated. TrendLabs is a service mark of Trend Micro Incorporated. All other company and/or product names may be trademarks or registered trademarks of their owners. Information contained in this document is subject to change without notice.

memory cage and can't affect any other. Therefore the segregation between these two modes provides a far more secure computing environment such as in Windows XP compared to that which we had in Win9x.

Though there are methods to alter the system behavior in user-mode and modern spyware threats have often used this to their own advantage, kernel-mode is the ultimate goal for a malicious attacker. A malicious program will be able to mangle with any other data structure in the memory, even the operating system code, if it is able to install itself as a kernel-mode driver. Pointers to the API functions, will point intead to the rootkit code.

Note that the key word here is "driver": only device drivers can obtain such a high access level. In this matter, permission levels are vital for the security of the network: if all users have "administrator" privileges, they will all be able to load drivers – even the malicious kind. Though this has already been discussed in the past, these new kernel-mode threats highlight the need for companies to build this into their consideration set, as they design their security system. If there were fewer users with administrator privileges, rootkits would be much less of a threat.

To be sure, it's difficult to develop rootkits. However, the ongoing development of these threats is of an open source nature, which means that the source code to create them is freely downloadable from the Internet. Just like other types of malware we've seen recently, more and more code is freely available – and the modular format makes it relatively easy for even a *script kiddie* to take this complex code and add it to their own programs.

The bot and spyware connection

Essentially, rootkits are only a technology, neither good nor evil in themselves. However, they are increasingly being used as a cloaking technology for spyware or other malware. And since rootkits are readily available, bot worm creators have begun hiding their creations behind rootkit screens in order to remain unnoticed for a longer period. This means that in the near future, we expect to see rootkit detection numbers rise. From a malicious attacker's point of view, jumping on the rootkit wagon has several advantages: freely downloadable source code and a separate detection, which, if caught, will not uncover the main program.

The Antivirus challenge: detection and elimination

The problem most antivirus solutions face at the moment is rootkit detection. This is due to the fact that rootkit-shielded malware is installed in the system, so traditional antivirus scanning would not see the malicious program. Thus it would remain undetected.

The three phases that companies can focus on in rootkit detection are the following:

1. **Intervention**. Detecting and stopping the rootkit file before it infects the system, by use of signature matching on the installer program.

2. **Behavioral Detection**. Detecting the rootkit as it is being installed in the system. Theoretically it is possible to analyze the behavior of programs as they are being executed in order to detect rootkit-like behavior. One problem is such techniques are prone to false positives, as legitimate programs may present similar behavior patterns (including, but not limited to, legitimate device drivers). This is a risky way of detecting rootkits.

3. **Cleaning**. Detecting the rootkit once it has been installed. The objective is to uncover the rogue driver while it's active by spotting some known unhidden part.

As with other malware, rootkit techniques are continuously evolving and rootkit authors are finding new ways to hide processes/files/registries more effectively, therefore thwarting AV vendors' attempts to provide detection. This is an ongoing battle that may never end, but it seems clear that rootkits are here to stay and will continue to pose a serious challenge. Antivirus vendors must focus on providing interception, detection and cleaning capabilities to combat this growing threat.

About Trend Micro

Trend Micro Inc. provides centrally controlled server-based virus protection and content filtering products and services. By protecting information that flows through Internet gateways, email servers, and file servers, Trend Micro allows companies worldwide to stop viruses and other malicious codes from a central point before they reach the desktop.