WHITE PAPER: SYMANTEC SECURITY RESPONSE

symantec.

Windows Rootkit Overview

Windows Rootkit Overview

Contents

Introduction

The term rootkit originally referred to a collection of tools used to gain administrative access on UNIX operating systems. The collection of tools often included well-known system monitoring tools that were modified to hide the actions of an unauthorized user. An unauthorized user would replace the existing tools on the system with the modified versions preventing authorized users from discovering the security breach.

However, under Microsoft Windows, the term rootkits has taken on a more narrow definition. Rootkits in Windows refers to programs that use system hooking or modification to hide files, processes, registry keys, and other objects in order to hide programs and behaviors. In particular, Windows rootkits do not necessarily include any functionality to gain administrative privileges. In fact, many Windows rootkits require administrative privileges to even function.

Two basic classes of Windows rootkits exist – kernel mode rootkits and user mode rootkits.

User Mode Rootkits

User mode rootkits involve system hooking in the user or application space. Whenever an application makes a system call, the execution of that system call follows a predetermined path and a Windows rootkit can hijack the system call at many points along that path.

One of the most common user mode techniques is the in memory modification of system DLLs. Windows programs utilize common code found in Microsoft provided DLLs. At runtime, these DLLs are loaded into the application's memory space allowing the application to call and execute code in the DLL.

For example, an application may wish to enumerate all the registry keys on the system. In order to do so, the application calls a Windows API (RegEnumKey) whose code resides in ADVAPI32.DLL. A user mode rootkit would find the location of the API in ADVAPI32.DLL and modify the API so when the API is called, execution is redirected to the rootkit's code instead. The rootkit code would generally call the API itself and then modify the return results (e.g., removing itself from the list of returned registry keys) before returning the results to the application.

However, since user mode applications all run in their own memory space, the rootkit needs to perform this patching in the memory space of every running application. In addition, the rootkit needs to monitor the system for any new applications that execute and patch those programs' memory space before they fully execute. While trivial, other system hook techniques do not require such active monitoring and continual patching. In particular, one can simply hook the system call further down the path where all paths converge in the kernel.

Kernel Mode Rootkits

Kernel mode rootkits involve system hooking or modification in kernel space. Kernel space is generally off-limits to standard authorized (or unauthorized) users. One must have the appropriate rights in order to view or modify kernel memory. The kernel is an ideal place for system hooking because it is at the lowest level and thus, is the most reliable and robust method of system hooking.

The system call's path through the kernel passes through a variety of hook points. A few of these points will be described below.

As a system call's execution path leaves user mode and enters kernel mode, it must pass through a gate. The purpose of the gate is to ensure user mode code does not have general access to kernel mode space protecting the kernel space. This gate must be able to recognize the purpose of the incoming system call and initiate the execution of code inside the kernel space and then return results back to the incoming user mode system call. The gate is effectively a proxy between user mode and kernel mode. In older versions of Windows, this proxy is invoked via interrupts and in newer versions of Windows via model specific registers (MSRs). Both mechanisms can be hooked causing the gate to direct execution to the rootkit rather than the original kernel mode code.

Another popular hook point is to modify the System Service Descriptor Table (SSDT). The SSDT is a function pointer table in kernel memory that holds all the addresses of the system call functions in kernel memory. By simply modifying this table, the rootkit can redirect execution to its code instead of the original system call. Similarly to the previously mentioned techniques, the rootkit would likely call the original system call and then remove itself from the results before passing back the results.

Finally, another kernel mode rootkit technique is to simply modify the data structures in kernel memory. For example, kernel memory must keep a list of all running processes and a rootkit can simply remove themselves and other malicious processes they wish to hide from this list. This technique is known as direct kernel object modification (DKOM).

Conclusion

Many of these techniques are actually well documented and utilized by a variety of applications. However, what differentiates general system hooking from a rootkit is the action of hiding programs and behaviors such that their actions are hidden from the user. For example, a desktop firewall program may utilize similar system hooking mechanisms to watch and alert the user to any outgoing network connections whereas a rootkit will use the system hooking mechanism to hide their malicious backdoor communication channel.

Rootkit code has the ability to potentially bypass security applications and tools used to discover the existence of an intruder and malicious programs and thus, the use of rootkits in malicious code is clearly on the rise. In addition, open source and ready-to-use rootkit applications are now widely available on the Internet and thus, malicious code writers who may not even understand how a rootkit works can now easily integrate this technology into their own code.

About Symantec

Symantec is the global leader in information security, providing a broad range of software, appliances, and services designed to help individuals, small and mid-sized businesses, and large enterprises secure and manage their IT infrastructure. Symantec's Norton™ brand of products is the worldwide leader in consumer security and problem-solving solutions. Headquartered in Cupertino, California, Symantec has operations in 35 countries. More information is available at www.symantec.com.

Symantec has worldwide operations in 35 countries. For specific country offices and contact numbers, please visit our Web site. For product information in the U.S., call toll-free 1 800 745 6054.

Symantec Corporation
World Headquarters
20330 Stevens Creek Boulevard
Cupertino, CA 95014 USA
408 517 8000
800 721 3934
www.symantec.com