Microsoft | Malware Protection Center

Threat Report: Rootkits

June 2012

Microsoft

Microsoft Malware Protection CenterThreat Report: Rootkits

Author

Heather Goudey – *Microsoft Malware Protection Center*

Contributors

Jason Conradt – *Microsoft Protection Technologies*
Peter Ferrie – *Microsoft Malware Protection Center*
Joe Johnson – *Microsoft Protection Technologies*
Scott Molenkamp - *Microsoft Malware Protection Center*
Hamish O'Dea – *Microsoft Malware Protection Center*
Oleg Petrovsky – *Microsoft Malware Protection Center*
Tim Rains – *Microsoft Trustworthy Computing Communications*
Jasmine Sesso – *Microsoft Malware Protection Center*
Jeff Williams – *Microsoft Malware Protection Center*

Product Marketing

Ken Malcolmson – *Microsoft Trustworthy Computing Communications*

Table of Contents

Introduction

This Microsoft Malware Protection Center (MMPC) Threat Report examines one of the more insidious types of malware threatening organizations and individuals today — the rootkit. The report examines how attackers use rootkits, and how rootkits function on affected computers. The report describes some of the more prevalent malware families that use rootkit functionality in the wild today, before presenting some recommendations that can help organizations mitigate the risk from rootkits.

The purpose of rootkits

A rootkit, or rootkit functionality, provides stealth capabilities to malware. Many modern malware families must persist on a compromised computer for an extended period in order to be considered successful by the attacker. The purpose of much malware involves the theft of sensitive data or other abuse of resources, such as using a computer to perform *click-fraud*[1]. The malware needs to stay hidden on the compromised computer in order to monitor, filter, capture, and exfiltrate valuable data or subvert resources under the attacker's control. Rootkit functionality provides the stealth required to keep the malware hidden while the malware executes its *payload*, or actions such as downloading files, changing computer settings, logging keystrokes, and so on.

Rootkit etymology

It is worth pausing a moment to consider the origin of the term "rootkit." Originally, a rootkit was considered a suite of tools that an attacker could use to obtain the "root," or the highest level of privilege that is usually reserved for system administration, on a UNIX system, and then mask any resultant changes. In recent years, the term *rootkit* or "rootkit functionality" has more generally referred to malware that uses stealth functionality to hide itself to avoid detection and removal.

[1] "Click Fraud: Cybercriminals want you to 'like' it:" Security Tips & Talk blog, http://blogs.msdn.com/b/securitytipstalk/archive/2010/07/08/click-fraud-cybercriminals-want-you-to-like-it.aspx

How attackers use rootkits

Malware creators can devote significant resources to compromise computers, networks, and organizations. By using a rootkit, an attacker hopes to protect and maintain the compromise for as long as possible. The real value of the data and resources that the attackers pursue makes their efforts not only a viable exercise, but a profitable one. The more valuable the data, the more capital attackers can afford to invest in the tools required for a successfully targeted compromise. The level of investment and the importance of remaining undetected make rootkits a threat that should not be underestimated.

After a compromise has been made, and the attacker has established a presence on the targeted system or systems, the symptoms of that compromise need to be masked, as does the ongoing presence of the malware and other tools that the attacker might use. One of the most effective ways for an attacker to avoid detection is to provide no hint of compromise. If the affected organization has no idea that it has been infiltrated, it is unlikely to take additional investigative or more stringent security measures to uncover the attack, or to undertake further remediation or hardening measures.

Undetected, a successful rootkit can potentially remain in place for years, stealing data and resources from the affected system. While antivirus technologies are generally very good at generically and proactively detecting many types of malware, in practice the ability to detect new malware relies on effectively gathering intelligence on it. The lengths to which rootkit authors will go to prevent their creations being discovered makes gathering the necessary intelligence difficult. This raises hard questions for organizations to address. For example, how can an organization meaningfully protect users from threats that, without intelligence, are hypothetical? And how can the organization accurately judge the scope of the problem without accurate intelligence regarding the prevalence of the threat? Gaining a better understanding about how rootkits work, and what types of known malware use them, positions the organization to more effectively answer these questions.

How rootkits work

A rootkit works by essentially inserting itself into a system to moderate – or filter - requests to the operating system. By moderating information requests, the rootkit can provide false data, or incomplete data, to utterly corrupt the integrity of the affected system. This is the key function of a rootkit and explains why rootkits are a serious threat - after a rootkit is installed, it is no longer possible to trust any information that is reported back from the affected computer.

For example, requesting a list of processes on a computer affected by a rootkit may return a list of all running processes minus any relating to the rootkit, or other components that it

protects. Commonly, malware uses rootkit functionality to hide files, registry modifications, evidence of network connections, and processes, as well as other possible indicators of the malware's presence.

There are several places that a rootkit can insert itself into an operating system to perform its filtering function. The "type" of rootkit is determined by where the rootkit performs its subversion of the execution path. For these reasons, rootkits have historically been referred to as either *user mode rootkits* or *kernel mode rootkits*:

- **User mode rootkits**: This type of rootkit filters requests for information that originate from user-mode applications by *hooking* application programming interface (API) functions. Hooking covers a range of techniques used to alter or augment the behavior of applications by intercepting function calls or messages or events passed between software components. Code that handles such intercepted function calls, events or messages is called a "hook." This rootkit functionality is more accessible for malware developers, as writing successful user-mode code is generally easier than writing successful kernel-mode code[2]. However, user-mode hooks are also generally easier to detect.
- **Kernel mode rootkits**: This type of rootkit performs its filtering and hooking at the kernel level. Filtering at this level is more effective, but is also more difficult to successfully achieve without corrupting the affected system. One way of introducing code into the kernel is by using a device driver. There are several different methods that are used for hooking at this level, including for example, *inline hooking* where code is modified in place, or *patching the System Service Dispatch Table* to hook particular events.

Some more recent rootkits, sometimes referred to as MBR rookits, or "bootkits," modify the Master Boot Record (MBR) to gain control of the system and start the process of loading the rootkit at the earliest possible point in the boot sequence[3].

Conceptually at least, it is possible to go deeper still into the execution path, and there have been several rootkit proofs-of-concept that illustrate this. The "Blue Pill"[4] concept focuses on the idea of using a thin hypervisor to create a virtual instance of the operating system that interfaces with the affected user. The *hypervisor* is the processor-specific virtualization platform that allows multiple isolated operating systems to share a single hardware platform. The hypervisor would be able to intercept and modify almost every request for data, regardless of the source, because of its position between the "bare-metal" operating system and the user's virtual operating system. Deeper yet in the execution path lies the possibility of compromised firmware intercepting data at the network level[5].

The further down the rootkit can embed itself in the execution path to intercept and filter requests to the operating system, the more successfully it can provide stealth. However, the deeper the placement of the rootkit, the more difficult it becomes to successfully implement,

---

[2] Kasslin, K. et al (2005) *Hide n' seek revisited – Full stealth is back*. Virus Bulletin Conference October 2005
[3] MMPC Malware encyclopedia DOS/Alureon description
www.microsoft.com/security/portal/Threat/Encyclopedia/Entry.aspx?Name=DOS%2fAlureon
[4] http://theinvisiblethings.blogspot.com/2006/06/introducing-blue-pill.html
[5] Presentation at Hack.lu:Reversing the Broadccom NetExtreme's firmware – Sogeti Esec Lab Blog, http://esec-lab.sogeti.com/dotclear/index.php?post/2010/11/21/Presentation-at-Hack.lu-%3A-Reversing-the-Broacom-NetExtreme-s-firmware

and the more complicated and expensive it becomes to develop the rootkit. Correspondingly, the deeper the location of the rootkit in the execution path, the more difficult it can be to remove it. The following figure illustrates the possible effect of a kernel mode rootkit compromise.

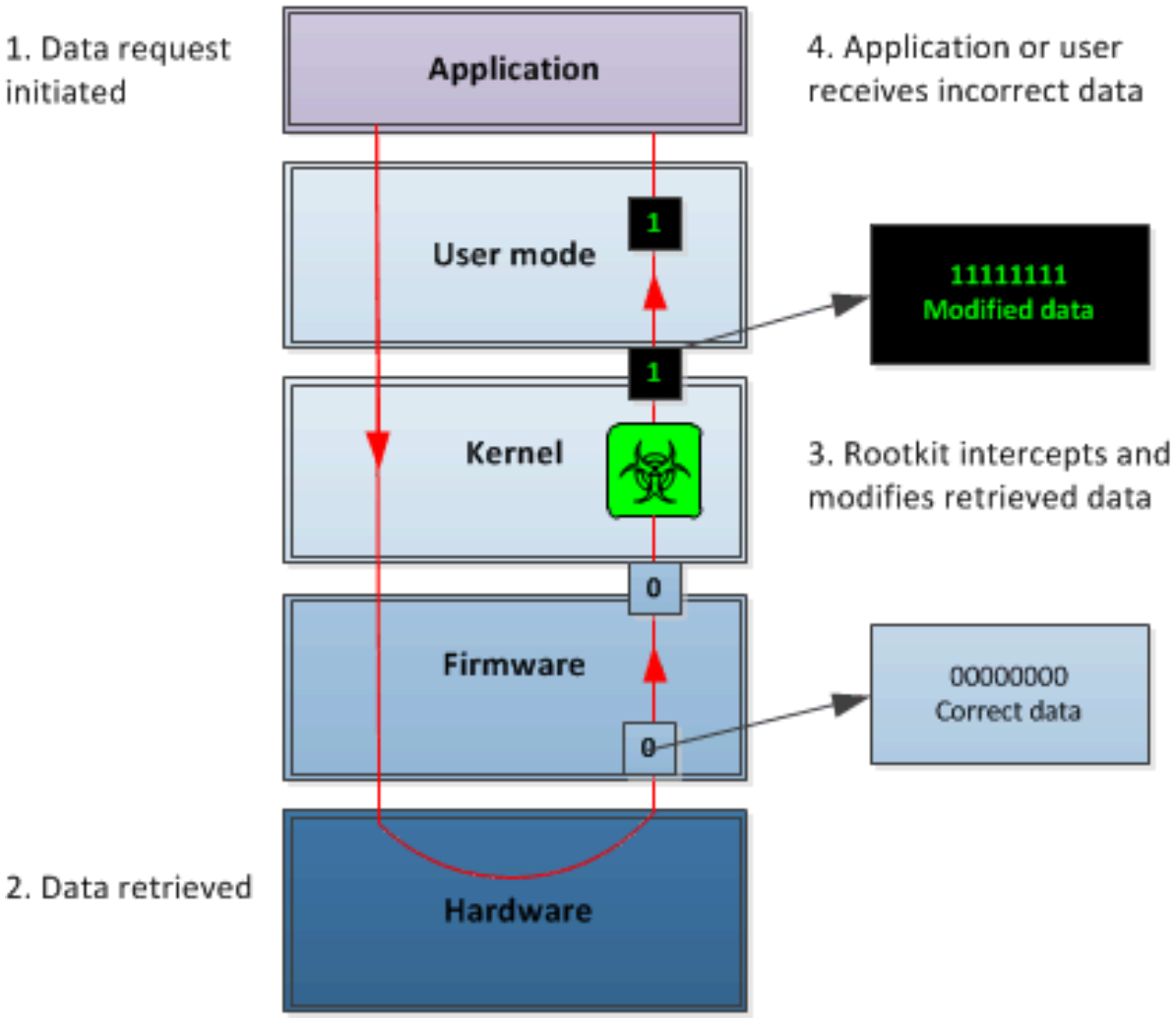

Figure 1. Possible effect of a kernel mode rootkit compromise

Scope of the rootkit problem

Many modern malware families use rootkit techniques to hide from affected users and to avoid possible detection and removal. The use of stealth functionality by malware has become increasingly common over much of the last decade. Despite the increasing use of these techniques, the MMPC has gathered intelligence on many of these threats, and spends a considerable amount of time and effort performing research to help protect technology users from these threats.

However there are some rootkits, and other malware, that are developed specifically to target particular organizations and are therefore not prevalent in the general threat landscape, making them harder to detect. Judging the scope of this particular type of malware/rootkit threat, and how many organizations it threatens, is challenging. For these reasons, regardless of current intelligence, there is sufficient evidence indicating that these threats are significant, and that all organizations that hold data of value need to take appropriate precautions.

Notable malware families that use rootkits

Some of the most prevalent malware families today consistently use rootkit functionality. The following list describes a number of the more notable examples:

Win32/Alureon[6]. A multi-component family of trojans that is involved in a broad range of subversive activities online that generate revenue from various sources for its controllers. Win32/Alureon is mostly associated with moderating affected user activities online to the attacker's benefit. As such, the various components of this malware family have been used to:

- Modify affected user search results (also known as *search hijacking*).
- Redirect affected user browsing to sites of the attacker's choice (also known as *browser hijacking*).
- Change DNS settings to redirect users to sites of the attacker's choice without affected user knowledge.
- Download and execute arbitrary files, including additional components and other malware.
- Serve illegitimate advertising.
- Install rogue security software.
- Perform banner clicking (for pay-per-click advertising)

Win32/Alureon has been actively developed, aggressively deployed, and professionally managed by its authors for many years. The pervasiveness of its components in the wild, which other malware families often use, and its use of stealth, makes this malware family a notable threat.

Alureon has used several methods to hide its processes and other system changes, including the following:

- Installing malicious device drivers that enable Alureon to hook the System Service Dispatch Table (SSDT) and Windows APIs in order to intercept file system requests that allow it to hide and prevent access to files, registry entries, and processes with names that contain a particular string.[7]
- Infecting existing system device drivers with malicious code that enables Alureon to insert itself into the part of the kernel that handles disk operations to hide files and disk

[6] MMPC malware encyclopedia Win32/Alureon description
http://www.microsoft.com/security/portal/Threat/Encyclopedia/Entry.aspx?Name=Win32%2fAlureon
[7] MMPC malware encyclopedia Trojan:WinNT/Alureon.C description
www.microsoft.com/security/portal/Threat/Encyclopedia/Entry.aspx?Name=Trojan%3aWinNT%2fAlureon.C

sectors[8]. More recent variants of Alureon perform these actions without infecting system files.

- Infecting the Master Boot Record (MBR), including successfully infecting the MBR on 64-bit Windows operating systems allowing it to bypass the operating system's kernel mode code signing policy and PatchGuard protection.

Win32/Rustock[9] – (for detailed information about the Rustock family download the **Microsoft Malware Protection Center Threst Report – Rustock** available at go.microsoft.com/?linkid=9777259). A multi-component family of rootkit-enabled backdoor trojans initially developed to aid in the distribution of "spam" email through a *botnet*. A botnet is a large attacker-controlled network of compromised computers. First discovered sometime in early 2006, Rustock evolved to become a prevalent and pervasive threat. Some reports suggest that at its peak, the million-strong Rustock botnet was responsible for almost 80 percent of spam traffic, sending more than 2,000 spam messages per second.

Rustock used a complex method to install its drivers to complicate its detection and removal.[10] In addition, the rootkit drivers hooked system functions to hide itself and its components. This was achieved by patching the SSDT to hook the events ZwCreateEvent, ZwCreateKey, and ZwOpenKey. This method made it possible for the rootkit drivers to filter requests containing each driver's name and return STATUS_UNSUCCESSFUL if matched, thus avoiding detection. Rustock also attempted to hide network and disk I/O operations. To achieve this, a driver of this rootkit hooked the set of ntoskrnl.exe and ntdll.dll APIs, and then communicated directly with the NTFS file system (NTFS) and TCP/IP devices, such as NTFS, IP, TCP, UDP, RawIP, and IPMULTICAST.

Microsoft, in conjunction with industry and academic partners, utilized a novel combination of legal and technical actions to take control of the Rustock botnet in March 2011 as part of Project MARS (Microsoft Active Response for Security)[11]. This action resulted in the gathering of evidence which became part of an ongoing criminal investigation[12].

Win32/Sinowal[13]. A multi-component family of malware that attempts to steal sensitive data, such as user names and passwords for different systems. This includes attempting to steal authentication details for a variety of FTP, HTTP, and email accounts, as well as credentials used for online banking and other financial transactions. Sinowal may specifically attempt to target and replace digital certificates used by the affected user during encrypted Secure Socket Layer (SSL) transactions, thus corrupting the integrity of these communications. Sinowal may also provide backdoor functionality to the remote attacker, allowing unauthorized access and control of an affected computer that that attacker may then use to download and execute

---

[8] MMPC malware encyclopedia Virus:Win32/Alureon.A
http://www.microsoft.com/security/portal/Threat/Encyclopedia/Entry.aspx?Name=Virus:Win32/Alureon.A
[9] MMPC malware encyclopedia Win32/Rustock description
http://www.microsoft.com/security/portal/Threat/Encyclopedia/Entry.aspx?Name=Win32%2fRustock
[10] Uprooting Win32/Rustock – MMPC Threat Research & Response blog
http://blogs.technet.com/b/mmpc/archive/2008/10/18/uprooting-win32-rustock.aspx
[11] Operation b107 – Rustock botnet takedown - MMPC Threat Research & Response blog
http://blogs.technet.com/b/mmpc/archive/2011/03/17/operation-b107-rustock-botnet-takedown.aspx
[12] http://blogs.technet.com/b/microsoft_blog/archive/2011/09/22/rustock-civil-case-closed-microsoft-refers-criminal-evidence-to-fbi.aspx
[13] MMPC malware encyclopedia Win32/Sinowal
www.microsoft.com/security/portal/Threat/Encyclopedia/Entry.aspx?Name=Win32%2fSinowal

arbitrary files. Sensitive data captured by Sinowal may also be uploaded to a website for retrieval by the attacker.

Sinowal's data stealing payload makes its extended presence on an affected computer a key determiner of the malware's success. Sinowal thus attempts to use stealth to maintain its presence and avoid being detected while it silently gathers data and sends it to a remote attacker. Similar to Rustock, Sinowal also uses a complex method to install its drivers. The eventual effect of these machinations is that the MBR is overwritten with malicious code, and the main driver is written to the end of the physical drive[14]. With these changes in place, Sinowal can gain control of the affected system loading its driver at an early point in the boot process.

Win32/Cutwail[15]. A trojan that downloads and executes arbitrary files. The downloaded files may be executed from disk or injected directly into other processes. While the functionality of the downloaded files is variable, Cutwail usually downloads other components that send spam. Cutwail also employs a rootkit and other defensive techniques to avoid detection and removal.

Cutwail uses a kernel mode rootkit. It installs several device drivers to hide its components from affected users. However, Cutwail not only can hide itself, it can also prevent the removal of its files and registry entries. To hide and protect its registry entries, Cutwail hooks the functions ZwDeleteValueKey(), ZwEnumerateKey(), ZwEnumerateValueKey(), ZwOpenKey(), and ZwSetValueKey()in the SSDT. To protect its files on disk, it also implements a file system filter driver.

---

[14] MMPC malware encyclopedia VirTool:WinNT/Sinowal.A
www.microsoft.com/security/portal/Threat/Encyclopedia/Entry.aspx?name=VirTool%3aWinNT%2fSinowal.A
[15] MMPC malware encyclopedia Win32/Cutwail
www.microsoft.com/security/portal/Threat/Encyclopedia/Entry.aspx?Name=Win32%2fCutwail

Protection against rootkits

The most effective way to avoid infection by rootkits is to defend against installation of the rootkit. Once the rootkit is installed, its stealth capabilities make it much more difficult to detect and remove it and its components, and other files it might download. For these reasons, it makes sense to take every possible precaution to avoid a possible compromise.

Toward this end, it is recommended that organizations ensure their virtual perimeter is strong and secure by investing in protective technologies, such as antivirus and firewall products. For more guidance and information about securing organizations see the Managing Risk section of the Microsoft Security Intelligence Report - www.microsoft.com/security/sir/strategy/default.aspx#!section_1.

Ensure that any antivirus solution takes a comprehensive approach to protection by using both traditional signature-based detection, heuristic detection, dynamic and responsive signature capability, and behavior monitoring. Ensure that the signature sets are kept up to date, ideally using an automative update mechanism. For more information about antimalware technologies, see the MMPC document "**Introducing Antimalware Technologies**" available at go.microsoft.com/?linkid=9776701.

It is also necessary to carefully examine and monitor for possible points of vulnerability in systems, limit the use of higher risk technologies by users in the organization, and apply security updates in a timely manner to all software in the organization.

Organizations should also protect their employees by making them aware of the risks that malware poses and ensuring that they receive appropriate security awareness training. For more information and guidance see the Internet Safety for Organizations Toolkit at www.microsoft.com/security/resources/powerpoint.aspx.

To effectively implement these recommendations, using some form of Network Inspection System (NIS) and Intrusion Prevention System (IPS) is warranted.

Once protection is in place, it is important to practice vigilance by monitoring systems and investigate possible deviations in traffic and behavior on both individual hosts and the greater network. In even a modest-size organization this can be a daunting task. Organizations should identify their high value assets (for example, key intellectual property) and design a monitoring and analysis scheme that focuses on these assets.

If a possible compromise has been detected other specialist technologies are available for use. Many antivirus offerings include special antirootkit technology. Microsoft antivirus solutions include a number of technologies designed specifically to mitigate rootkits, including live kernel behavior monitoring that detects and reports on attempts to modify an affected

system's kernel, and direct file system parsing that facilitates the identification and removal of hidden drivers.

Finally, if a system has been determined to be compromised, then an additional tool might be warranted that allows you to boot to a known good or trusted environment so that appropriate remediation measures can be taken. In this case, the Standalone System Sweeper tool (part of the Microsoft Diagnostics and Recovery Toolset (DaRT)) or Windows Defender Offline may be useful. Booting the compromised system using a known good, uncompromised operating system, will allow antivirus technolgieis and other tools to identify malware components that are otherwise hidden by the rootkit. This technique can be an effective tool to help defend and recover from compromise where rootkits are employed.

General Guidance: Defending Against Malicious and Potentially Unwanted Software

Effectively protecting users from malware requires an active effort on the part of organizations and individuals to maintain up-to-date antimalware defenses, and to stay informed about the latest developments in malware propagation techniques, including social engineering.

For in-depth guidance, see the following resources in the "Mitigating Risk" section of the Security Intelligence Report website:

- Promoting Safe Browsing

  http://www.microsoft.com/security/sir/strategy/default.aspx#!section_2_3

- Protecting Your People

  http://www.microsoft.com/security/sir/strategy/default.aspx#!section_4

Further reading

The following resources provide an excellent introduction to learn more about rootkits and how malware authors use rootkit functionality:

- Blunden. B., (2009) *The Rootkit Arsenal: Escape and Evasion in the Dark Corners of the System*. Jones & Bartlett
- Hoglund, G. and Butler, J. (2006) *Rootkits – Subverting the Windows Kernel*. Upper Saddle River: Addison-Wesley
- Kasslin, K. et al, (2005) *Hide 'n seek revisited – Full stealth is back*. Virus Bulletin Conference October 2005

One Microsoft Way
Redmond, WA 98052-6399
microsoft.com/mmpc