ROOTKITS

by ThreatBlogger

TABLE OF CONTENTS

INTRODUCTION

Rootkits have been around many years and can remain undetected stealing resources, data or monitor communications. In this presentation we will explore at early development of rootkits and the most common popular. We will be looking under the hood, how rootkits work and the different types of rootkits. Explore Modern Rootkits from virtual machine to firmware rootkits and looking at various samples. Finally we will look at the prevention measures large tech organizations are taking, and the detection and removal of rootkits.

WHAT IS A ROOTKIT

A rootkit can be thought of as special software with built-in tools to create a second administrator on a targeted system, while hiding itself from other admins. In most cases, rootkits don't do much damage, apart from modifying the operating system. The main function of the rootkit is to keep the malware that it's linked to from being discovered. The malware does the actual damage.

The primary functions of rootkits are:

- To remain undetected, making it hard for security analysts to detect.
- For the rootkit to remain persistence, surviving removal attempts and reboots.
- Hide malware linked to the rootkit, which may be part of a larger sustained attack.
- Provide the attacker access to the machine, often via backdoors.
- The rootkit can escalate the privilege level which the malware operates.
- The compromised machine can be used as a member of a bot.

EARLY ROOTKITS

Back in 1983, one of the creators of the Unix operating system, a man called Ken Thompson, actually theorized a rootkit long before they came into existence. He conceptualized an exploit which could subvert a login command, allowing an attacker to use another password to gain access to an administrator account.

A rootkit was known as a group of tools belonging to the Unix operating system itself, tools such as netstat, passwd and ps. These tools were modified by a remote intruder to get unlimited access (or root access), without this being detected by a system administrator.

These terms and concepts were carried over to the Windows operating system which we still use today.

The Brain Virus

The Brain virus was discovered in 1986, it affected the IBM PC, it attempted hide its existence, and is known as the first stealth virus. It was created with good intentions, to prevent users from pirating their software.

It's a boot sector virus which intercept the system Functions access to disk. When the boot sector was read (by for example an antivirus solution) the virus would replace infected data with the original data, Clean data. It infected the boot sector of a floppy disk.

The brain virus intercepted attempts to read the boot sector, and redirect these attempts to elsewhere on the disk, where a copy of the original boot sector was kept.

Early Unix Rootkits

Rootkits as we know them now, came into being sometime in the mid 1990's. At that time, Sun operating system Unix administrators started seeing strange server behavior, missing disk space, CPU cycles and network connections that strangely did not shot in the command netstat.

Many of the tools (such as log file cleaners) that later became inherent parts of rootkits were known as long ago as 1989 and even earlier in the underground. Rootkits are automated software packages to setup and maintain an environment on a compromised machine.

- Utilities for escalating user privileges
- resource usage monitor utilities
- packet sniffers to monitor network traffic

From log cleaners to live kernel patching:

- 1989: First log cleaners found on hacked systems
- 1994: Early SunOS kits detected
- 1996: First Linux rootkits publicly appear
- 1997: LKM Trojans proposed in "Phrack"
- 1998: Non-LKM kernel patching proposed by Silvio Cesare
- 1999: Adore LKM kit released by TESO
- 2000: T0rnkit v8 libproc library Trojan released
- 2001: KIS Trojan and SucKit released
- 2002: Sniffer backdoors start to show up in kits

Early Windows Rootkits

- **NTRootkit:** This is the first documented windows rootkit, which targeted Windows NT machines. It was created by Greg Hoglund in 1999. NTRootkit was a proof-of-concept, developed to see what rootkits could do on Windows systems
- Greg was the first to build a stealth virus, later became known as a rootkit, which could hide data in the system, and using techniques for evading Windows protection.
- It hides certain files, processes and entries of the affected computer and logs the keystrokes typed by the user. Uses OS hooks to conceal its presence.
  NTRootkit is written in the programming language Visual C++

Other Windows Rootkits

- Researchers continued to investigate Windows system protection, and soon after NTRootkit was released several other tools appeared, all designed to hide objects in the operating system:
  - **He4hook (2000)** - The tool is not malicious, but it does hide files. It works in kernel mode.

- **Hacker Defender (2002) -** This is also just a tool, but a more powerful one: it can be used to hide files, processes and registry keys with flexible settings in the configuration file. It works primarily in user mode.
- **Vanquish (2003) -** This tool can be used to hide files, directories and registry keys. Moreover, it has a malicious payload – it log passwords. Vanquish works in user mode.

ROOTKIT TIMELINE

Timeline

Some of the other best know rootkits worth a mention

- **Machiavelli –** first seen in 2009 targeted Mac OS machines. This rootkit created hidden kernel threads.
- **Zeus –** An attack launched in 2007 targeting banking information using a man-in-the-browser (MITB) attack method, grabbing keystroke and logging information.
- **Flame –** Found in 2012, a spyware rootkit which infected windows systems.

ROOTKITS: UNDER THE HOOD

The Kernel Level

When we think of rootkits we think of kernel level rootkits. What is the kernel and how does it operate?

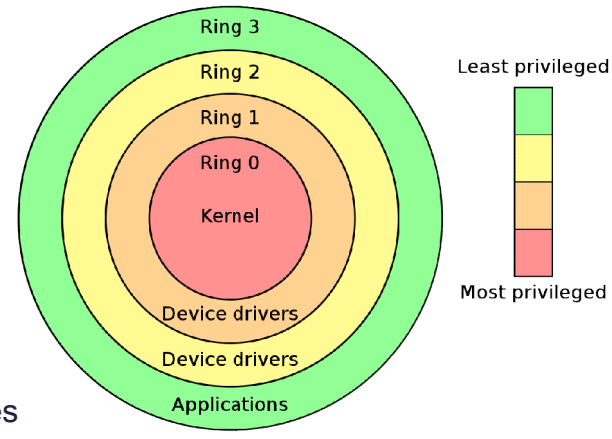
The kernel-Level: The kernel level is the unseen portion of the system where system drivers interact with the hardware. Low-level tasks are carried out by the kernel such as disk management, memory management, task management, etc. It's an interface between the user and hardware, and it primary purpose is stability and security the kernel is the central module of an operating system and it's the first to be loaded into main memory, it's placed in a protected part of memory so it won't be overwritten by other programs.
The kernel is responsible for:

- Process management for application execution
- Memory management, allocation, and I/O
- Device management through the use of device drivers
- System call control, which is essential for the execution of kernel services

The Kernel Protection Rings

Protection rings are a part of the system architecture which separates levels of interaction within an operating system in order to provide fault protection among users, components, applications and processes.

- Protection rings are a part of the system architecture which separates levels of interaction within an operating system in order to provide fault protection among users, components, applications and processes.
- Each ring is a separate privileged layer, giving processes different levels of access to system resources and hardware. The central ring (ring 0) otherwise known as the kernel has the highest privilege, it can access to everything, and has control of the operating system resources.
- The other layers have less privilege, ring 3 is the least privileged, and has access to user processes in user mode only.



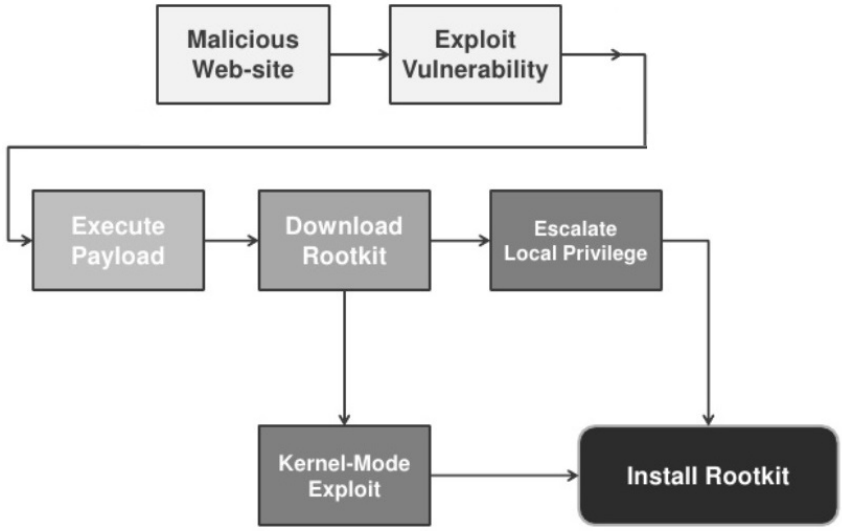
DIFFERENT TYPES OF ROOTKITS

- **Persistent Rootkits –** this type of rootkit activates automatically as the system boots, this type of rootkit is stored in system registry.
- **Memory Based Rootkits** – A memory based rootkit will not automatically stay after a reboot, and are only kept alive in active memory. When a system reboots they are lost.
- **User-Mode Rootkits –** The user mode rootkit operates at the application layer, using system API calls to the kernel. These types of rootkits normally change system binary files to malicious code, which can redirect control of the system to the hacker.
- **Kernel-mode Rootkits** – kernel mode rootkits hook to the kernel API's and modify the systems data structures within the kernel itself. These are the most persistent and difficult type of rootkits to detect and can remain on the system giving no indication of being active
- **Bootkits** - A bootkit is a variation of the kernel-mode rootkit which infect the Master Boot Record (MBR), this allows malicious code to be executed before the system boots.
- **BIOS / Firmware** - This type of rootkit infects a system hardware device, such as a system BIOS or network card. Malicious code is inserted into these devices usually by an update, or may have been preinstalled on the chip undetected by the manufacturer.
- **Hypervisor** - A hypervisor rootkit takes advantage of the hardware virtualization and is installed between the hardware and the kernel acting as the real hardware. Hence, it can intercept the communication/requests between the hardware and the host operating system. Common detection applications that run in user or kernel mode are not effective in this case as the kernel may not know whether it is executed on the legitimate hardware.

.

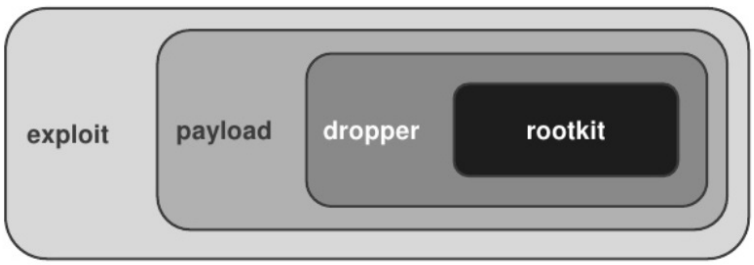
HOW A ROOTKIT INFECTS A SYSTEM

When you have a document that shows a lot of numbers, it's a good idea to have a little text that explains the numbers. You can do that here.



A rootkit has a number of ways it can be installed on a system:

- **Piggybacking** – rootkit's can be installed with apparently trustworthy software. When the system admin gives permission to install the software, the rootkit is also silently installed.
- **Blended Threat** – A rootkit cannot infect a target system on its own. It needs a combination of exploited vulnerabilities and a dropper and a loader to work.
    - **Dropper** – used to install the rootkit on the system. A dropper can be distributed in a number of ways such a social engineering or brute force attack.
    - **Loader** – The loader is launched after the dropper is executed by the user by either clicking or opening a file. The loader exploits vulnerabilities within the system ensuring the rootkit loads correctly.

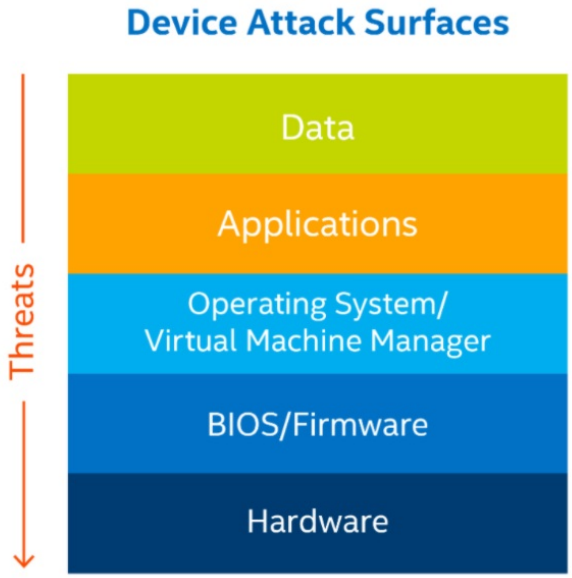


MODERN ROOTKITS

Recent years have witnessed an increasing number of attacks using more sophisticated, advanced rootkits, aimed at the lower system architecture.

These threats have been met by substantial efforts in the security community to improve current detection methods.

In this section we are going to look at two lower level attack surfaces, firmware and virtual machine based attacks.



Firmware Rootkits

The development advanced anti-malware software makes the cybercrimals respond, and create kernel root kits and firmware root kits, storing it in hardware. The code that starts right after the computer is turned on has the ultimate power over the operating system is called firmware. In the past this was done with the Bios, but today it's has been replaced with the Unified Extensible Firmware Interface or UEFI. The UEFI also boots up your system, plus starts up everything else. It resides on your flash memory, same place as the UEFI rootkit is hidden.

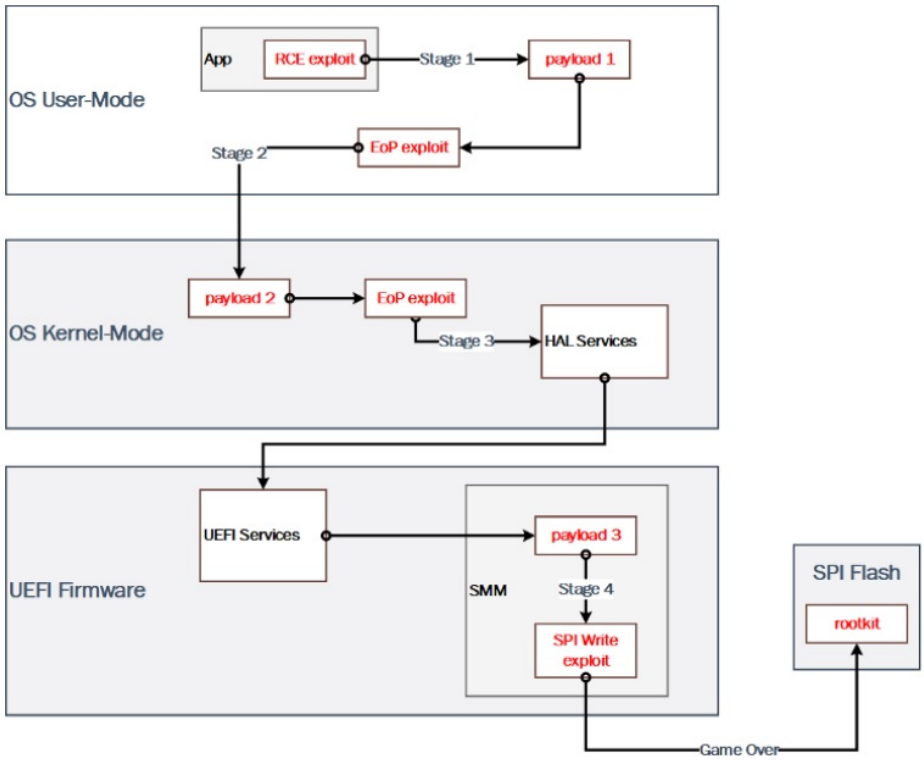
UEFI Rootkits

Like most of the first rootkits, the UEFI rootkit started out as a theory based rootkit. Some security professionals presented the UEFI rootkit as a poof-of-concept at security conferences. Up until August in 2018 no UEFI rootkit was ever detected in a real cyberattack. The genius of the UEFI rootkit is it resides in the one place where it's hard to get rid of it using normal security measures. Antivirus scanners does not find it, even wiping the hard drive does not remove it.

UEFI INFECTION

The infection is distributed via different methods, usually a phishing email message. The phishing email contains a Microsoft Word document, when clicked the user is prompted to engage the built-in macros. Embedded in the word document is a powershell dropper, when executed downloads a malicious code which flashes the rootkit to the UEFI firmware.

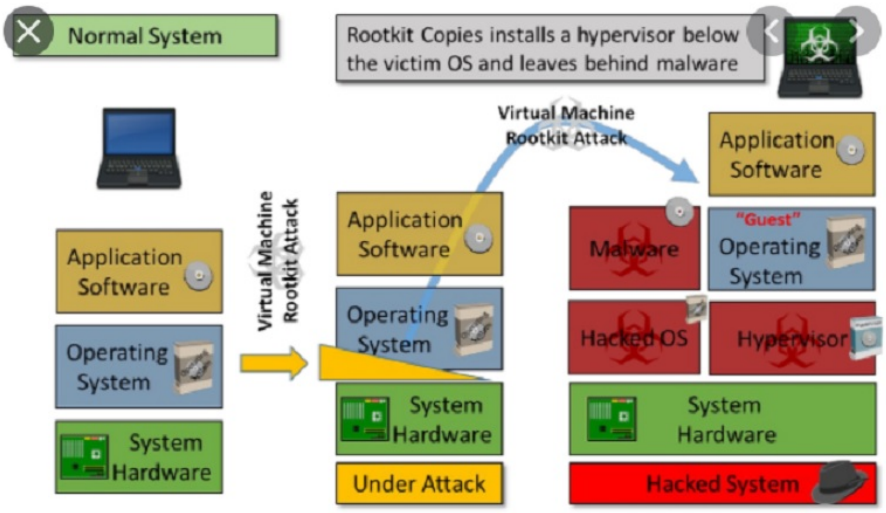
- **First-Stage of the attack (OS User-Mode) –** The first stage involves bypassing security measures to get the payload to run on system, escalating privileges, such as admin control.
- **Second-Stage attack (OS Kernel-Mode) –** The payload now infects the hardware Abstraction Services
- **UEFI Infection –** The final stage involves writing the malicious code, to the UEFI firmware which flashes the rootkit to the UEFI firmware.

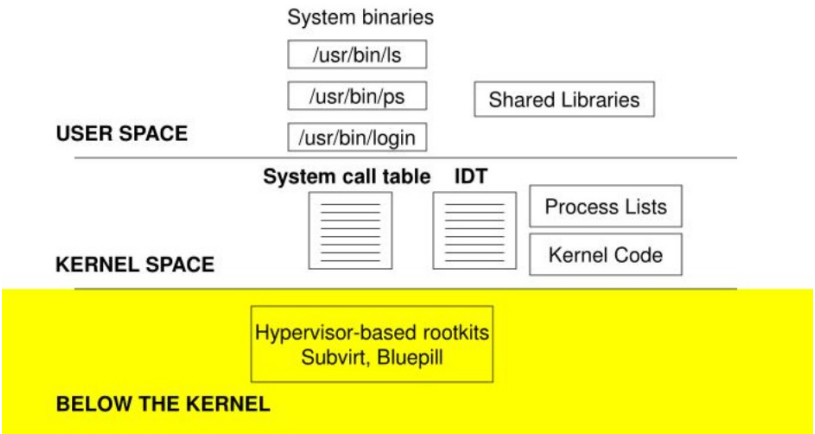


VIRTUAL BASED ROOTKIT

How Virtual Machine based rootkits (VMBR) works is to load itself underneath the existing OS. The existing OS then runs as a VM on top of the VMBR. When running this way, a VMBR could go undetected unless special tools are used to look for its existence. VMBRs are possible for both Linux and Windows platforms.

- A VMBR moves the targeted system into Virtual machine.
- Instead of moving the attack code lower into the kernel space, it pushes the user higher into user space.
- The pervious (unhooked) OS runs over a virtual machine (as the guest software)
- The guest is not allowed to interact with states outside of its Virtual machine.
- The attacker has the liberty to run anything on the machine.
- Any anti-rootkit software run inside of the virtual machine will not detect any modifications to its state.





SubVirt

- The proof-of-concept rootkit, called SubVirt, exploits known security flaws and drops a VMM (virtual machine monitor) underneath a Windows or Linux installation.
- SubVirt relies on commercial virtualization technology like VMware or Virtual PC.
- Once the target operating system is hoisted into a virtual machine, the rootkit becomes impossible to detect because its state cannot be accessed by security software running in the target system.

BluePill

- The "blue pill" references one of the pills offered to the hero Neo in the movie "The Matrix“.
- Blue Pill can do an on-the-fly install and simply shift your Operating System from direct control of the physical computer to a virtualized state.
- Blue Pill uses hardware virtualization and allows the OS to continue talking directly to the hardware.
- Blue Pill then acts as an ultra-thin Hypervisor that lies dormant most of the time using virtually zero overhead (on most tasks) and waits for "interesting" events such as keyboard input. Once keyboard input is tapped, any password entered in to the computer can be key logged with ease. Blue Pill can also have interaction with the network.

MODERN ROOTKIT EXAMPLES

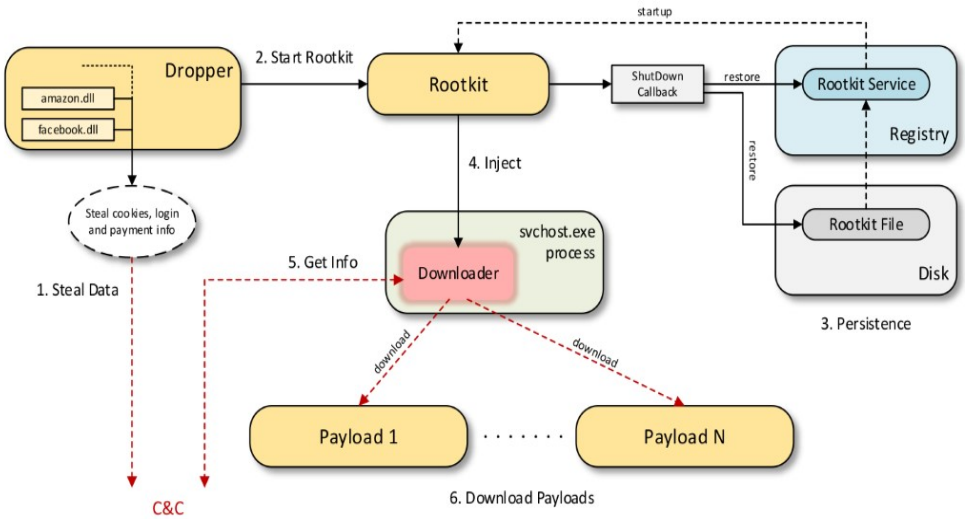
Rootkits of today are known for accomplishing stealth and persistence, they are a modular design and can be linked to other types of malware such as bots and are used in wider cyberattacks. They are Increasing Sophisticated, disable Aniti-virus products, have multiple Channels of Communication, extremely flexible and adaptive.
Here we will look at examples of rootkits of today.

- **Scranos** – (browser)
- **Lojax** – (firmware)
- **Pro-Ocean** (cloud)

Scranos

Scranos was first detected in China in 2019 by security researchers from Bitdefender, who now see it spreading to other countries making it a global cyberattack.

- The main component of Scranos is the dropper which executes the malicious software to install the rootkit.
- This rootkit is a digitally-signed rootkit driver, issued by a Chinese company. Cybercriminals may have obtained the original digital code-signing certificate or may have illegally compromised it.
- When the dropper is installed, it tries to communicate with its Command and Control (C&C) server and downloads other malicious payloads.



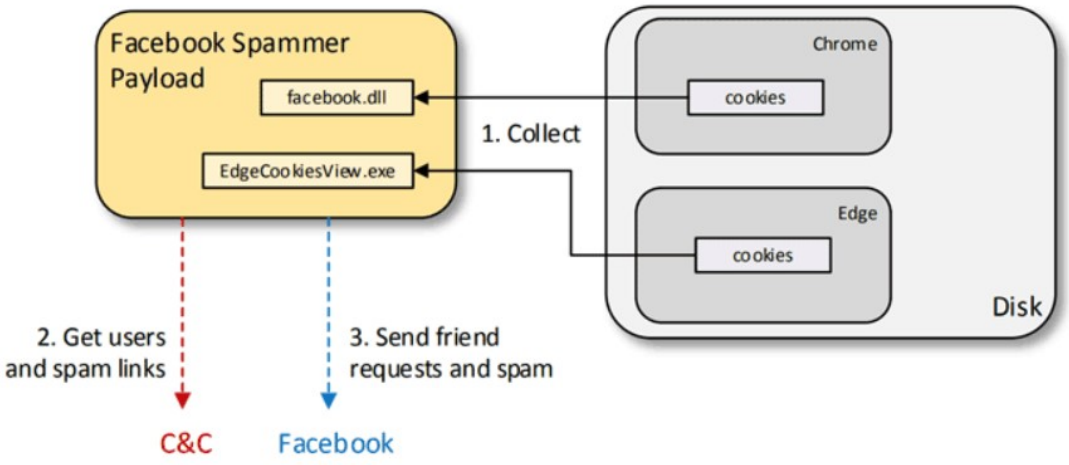
SCRANOS ACTIVITY

- It has been observed this malware aggressively promoting four different YouTube videos on different channels to users, in a bid to generate video revenue.
- Another downloadable component sends friend requests to other users, and also spams contacts with links to malicious Android apps.

The main functions of Scranos are:

- Injects adware into browsers and infiltrating browser history.
- Install browser extensions, launching malicious adware.
- Steals user credentials for the users account on Steam.
- Sends friend requests from the users Facebook account to other accounts.
- Install and run other malicious payloads.

The process is as follows:

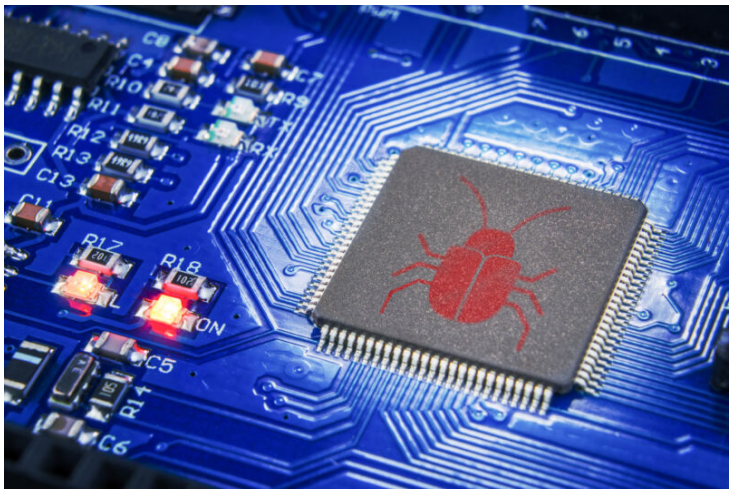


LOJAX

Lojax is the first rootkit to be detected that directly attacks the UEFI, allowing it to circumvent operating system reinstall. It was found bundled together with a toolset called LoJack

LoJack Software: Lojack is an anti-theft software package, which helped to protect Pc's by working its way deep inside the UEFI. Lojack helps to connect a Pc's operating system to its firmware. Lojax has taken advantage of this technology and modified it so it can remain hidden inside a pc.

- Lojax hooks into the system firmware and re-infects the system before the OS loads.
- Lojax can communicate with its remote command and control servers and execute tasks in relative safety.
- The only method to remove Lojax is to flash new firmware over the suspect system.

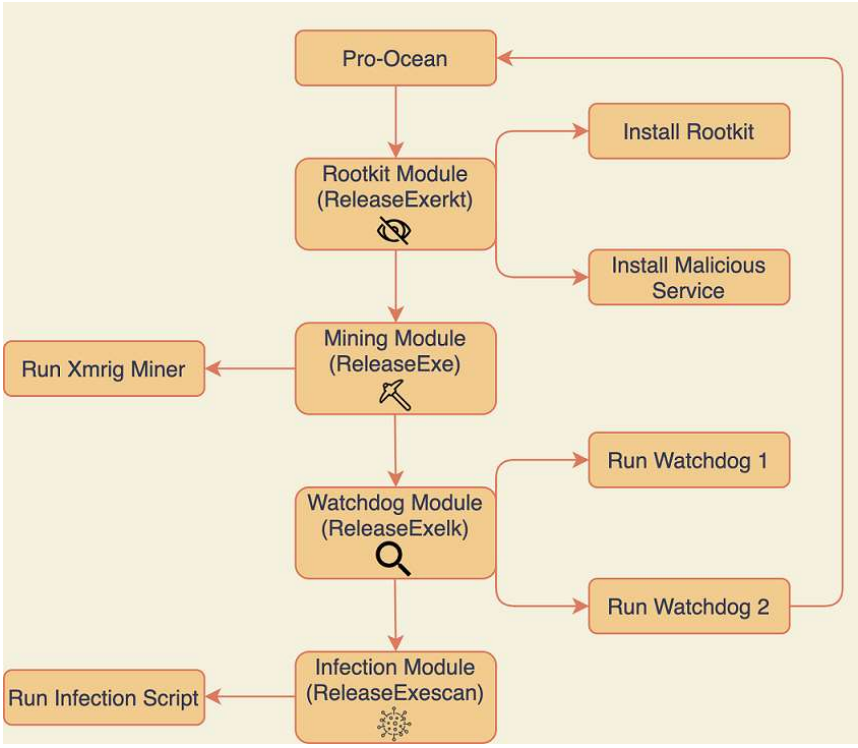


PRO-OCEAN MALWARE

Pro-Ocean first spotted in 2019, started out as crypto-mining malware, targeting the cloud infrastructure. It targets cloud applications using known vulnerabilities in out of date server packages like ApacheActiveMQ and Oracle WebLogic. A new revised and stealthier version has been seen this year, with added worm and rootkit modules.

A new growing trend can be seen of crypto-mining malware becoming a more sophisticated attack. Pro-ocean consists of four modules that get executed in order.

- Rootkit Module
- Mining Module
- Watchdog Module
- Infection Module

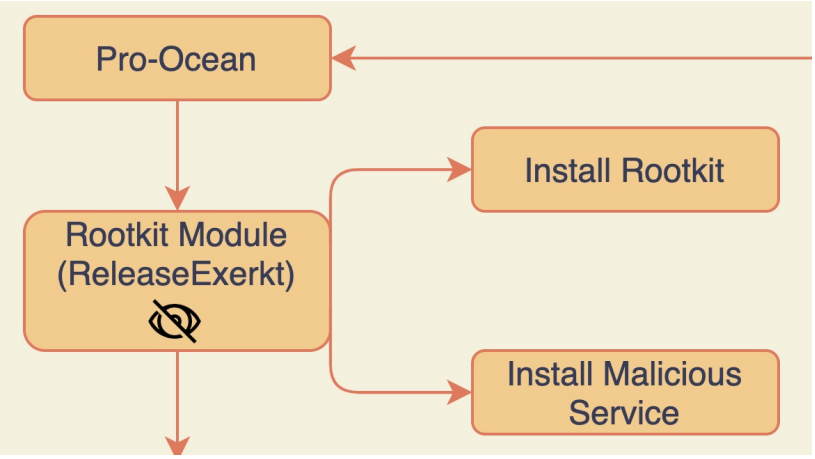


PRO-OCEAN: ROOTKIT MODULE

Pro-Ocean first spotted in 2019, started out as crypto-mining malware, targeting the cloud infrastructure. It targets cloud applications using known vulnerabilities in out of date server packages like ApacheActiveMQ and Oracle WebLogic. A new revised and stealthier version has been seen this year, with added worm and rootkit modules.

Pro-Ocean will try to gain persistence by copying itself into several locations, create malicious services, and execute the malware in case it's not running.

What is new is it uses publicly available code, helping to conceal its malicious activity.

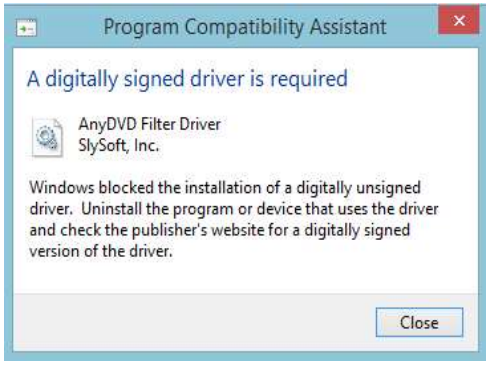


ROOTKITS: PREVENT, DETECT AND REMOVE

Anti – Rootkit Solutions

In this section we are going to look how large computer, software and chip manufactures have tried to eliminate, or stop the spread of rootkits infecting systems. We will also be covering the prevention, detection and removal of rootkits on company and personal devices.

Microsoft – Driver Signature Enforcement



Since Vista x64, Microsoft implemented the driver signature enforcement. When this feature is enabled, the drivers must be signed by a trusted vendor in order to be loaded by the system. This provides a reliable way of verifying the origin of a driver. It created problems for free Vista kernel-mode software, but was resolved by the release of windows 7.

Intel vPro Technology (1st – 10th Gen)

Software based tools were not keeping up with the stealthier malware, so a hardware approach was needed

In 2006 Intel released a new chip for the business community, the vPro processor, with new hardware-based security features that helps defend against stealth malware attacks and help secure virtual machines.

- **Intel Hardware shield**-provides enhanced protection against attacks below the OS level, protecting against firmware-level attacks.
- **Intel Threat Detection Technology -** is a set of technologies that harness hardware telemetry and acceleration capabilities to help identify threats and detect anomalous activity.
- **Intel vPro Technology (11th gen) -** The rise of ransomware attacks has lead Intel to provide application, data and lower-level security protection.

ROOTKITS: PREVENTION

Rootkit Prevention

There are many ways of preventing rootkit malware being installed on your system:

- Avoid Opening Suspicious Emails and downloading Cracked Software.
- Update your software, Windows and other companies release regular updates to fix bugs and vulnerabilities. Older programs may be exploited by cybercriminals taking advantage of vulnerabilities.
- Ensure you have stricter policies in place which only allow 3rd party drivers which are signed and verified.
- Use next-gen antivirus scanners, which can leverage modern security techniques like machine learning-based anomaly detection and behavioural heuristics.

Rootkit Detection

Rootkit detection is difficult, as these threats hide traces of themselves by nature. Attackers use rootkits so they can hide themselves and sit dormant for any amount of time, until the attacker executes the files or changes the configurations. Scan your systems with antivirus software, these can detect and remove application level rootkits.

- **Signature-Based Detection:** This is the most common technique for malware detection. However, it is the least efficient as it is only effective for already detected and wide-spread rootkits. Signatures from known rootkits are used to detect if any of them exist on a system.

- **Behavioral-Based Detection:** These detectors identify an abnormal behavior on a computer system based on heuristics and behavioral patterns. These patterns are derived from certain activities typically found in rootkits. The advantage of the behavioral based technique compared to the previous one, is that it may detect previously unknown rootkits.

ROOTKITS: DETECTION

Rootkit Detection

Look out for any strange activity happening on your system.

- One way attackers communicate is via the internet, so one place to start is reviewing is the TCP/IP packets travelling to and from that device. You should have a logging solution which alerts you to any usual traffic.
- If a system is misbehaving such as excessive CPU or internet bandwidth usage could be an indicator of a rootkit infection.
- Crypto-mining rootkits are known to kill high level processes on your system.
- Check your anti-virus is still working and your firewall configurations are intact.

Rootkit Removal Tools

- **Automatic Removal Tools:** The best method is to removal a kernel rootkit is to run automatic rootkit removals tools and Kernel level scanners. Malicious code can only be detected by kernel level scanners when the rootkit is inactive; this means all system processes have to be stopped. The most effective method is to reboot the computer in safe mode and scan with a variety of kernel level scanners, which should detect and remove any infections.

- **Manual Rootkit Removal:** To manually remove a rootkit takes expertise, so the usual option is to take your device to an expert. If you want to manually remove a rootkit infection. It can be time consuming, you can start by looking at suspicious files on your system, if you are familiar with windows legitimate services and programs the following set of tools may be helpful, AutoRuns, Processor Explorer and Msconfig.

ROOTKITS: REMOVAL

Manual Removal Tools

Windbg: is a free Window Debugger from Windows. Its original purpose was to help Window software developers debug code and fix software problems. It can debug user-mode and kernel-mode code, analyse crash dumps, and examine the CPU registers while the code executes.

- To perform Windows kernel analysis, we need a specific setup of the Virtual Machine and Windbg in order to remotely debug the VM from the host.
- Rootkits leverage kernel level components to facilitate activities such as hiding processes, files, network connections and other common objects.