Bootkits: evolution and detection method

Positive Technologies

ptsecurity.com

Contents

What is a bootkit and what's the risk

Cybercriminals are constantly on the lookout for new ways to gain a long-term foothold in the target system with maximum privileges, while avoiding detection, for example, by antivirus tools. Most protection tools are started together with the operating system, so if malware loads before the OS does, the likelihood of detection decreases. Another goal of malware developers is to retain control and privileges after OS reinstallation. This requires the malware to be loaded into low-level software—the device firmware or the first sectors of the hard drive. That is how bootkits appeared.

A **bootkit** is malicious code that runs before the OS boots. The main goal of a bootkit is to gain a foothold in the system and shield other malware from detection by security tools.

Real or conceptual?

Bootkits were previously thought to exist mainly in proof-of-concept[1] form, and not used in real attacks. However, only two years separated the appearance of the first PoC and the first bootkit attack.

Bootkit PoCs are of particular interest to analysts and researchers because they give an insight into which methods and techniques attackers are likely to use and what to look for to provide preventive protection.

Malware developers are now adding bootkit functionality to their creations, including Satana, Petya, and various botnets, such as TrickBot. APT groups are also active users of bootkits, for example, Careto, Winnti (APT41), FIN1, and APT28.

In preparing this report, we analyzed 39 bootkit families, both in PoC form and ones encountered in real attacks from 2005 to 2021.

Cybercriminals generally use targeted phishing via email to inject malware into the infrastructure; this is how, for instance, the Mebromi and Mosaic Regressor bootkits are distributed. Another delivery route is through websites, including the drive-by compromise technique, which was used to infect targets with the Pitou and Mebroot malware; cybercriminals distributing the latter hacked into more than 1,500 web resources and placed the malware there. The FispBoot bootkit got onto devices that were first infected with the Trojan-Downloader.NSIS.Agent.jd trojan, which victims downloaded under the guise of a video clip.

[1] A proof of concept (PoC) is a demonstration of the exploitability of a vulnerability.

The difference between a bootkit and a rootkit

Bootkits are often confused with rootkits.[2] The main difference is that bootkits start operating even before the OS boots. They have the same level of control as legitimate loaders (Master Boot Record (MBR), Volume Boot Record (VBR), or UEFI) and interfere with the OS boot process, allowing them to monitor and alter the boot process, as well as introduce, for example, malicious code, bypassing security mechanisms. Bootkits often create the environment for the stealthy introduction of kernel-level rootkits.

The **Master Boot Record (MBR)** contains information and code needed to properly boot the device. It is stored in the first sectors of the hard drive.

The **Volume Boot Record (VBR)** or Initial Program Loader (IPL) loads data needed to boot the OS. It is stored in the first sector of a partition on the hard drive.

Bootkit features

Most often, bootkits have the following features:

- Covert installation of the main payload, such as a rootkit or a backdoor in user mode
- Covert malicious activity, bypassing or even disabling protection tools
- Downloading of additional malware
- Escalation of system privileges

Some bootkits allow attackers to bypass authentication; the PoCs of the Vbootkit x64 and DreamBoot bootkits, for example, have this capability.

[2] A rootkit is a program (set of programs) for concealing the presence of malware in the system.

Bootkits as tools for highly targeted attacks

Developing one's own bootkit is no trivial task for an attacker, but in real life bootkits are quite common. For example, attackers spying on diplomats and members of non-governmental organizations from Africa, Asia, and Europe used the Mosaic Regressor bootkit to gain a foothold in the target systems. Having analyzed an attack using another state-of-the-art bootkit, MoonBounce, researchers were amazed at the attackers' deep knowledge of the victim's IT infrastructure. They saw that the attackers had thoroughly studied the device firmware, which suggests that this was a highly targeted attack.

However, cybercriminals use bootkits not only in targeted attacks, but also in mass ones. For example, the Rovnix bootkit was distributed as part of a phishing campaign using information about a new World Bank coronavirus initiative as bait. The purpose of the campaign seems to have been cyberespionage, since malware for remote control and spyware were subsequently installed on victims' devices. The Adushka bootkit is known for targeting regular users and being used for espionage, including data theft from online gaming accounts.

Another bootkit deployed in mass attacks is Oldboot. Its focus was Android devices. The attackers infected more than 350,000 mobile devices. Malicious code added to the boot partition of the file system was launched when the device was switched on for the first time. The bootkit created the environment for the introduction of a loader and spyware that collected and deleted text messages. To avoid infection, researchers advise against purchasing devices from untrusted stores and downloading firmware from dubious sources.

27 bootkit families deployed in the wild

14 of which are used by APT groups

Now bootkits are becoming increasingly common in the toolkit of attackers. And the regular discovery of vulnerabilities in firmware is contributing to this trend. For UEFI, for example, 14 entries appeared in the National Vulnerability Database (NVD) in 2021 alone. The liveness of bootkits is underscored too by the emergence of new functionality for commercial malware: in 2021, the developers of the FinSpy spyware upgraded their creation with bootkit features.

Bootkit detection and removal

Bootkit detection is feasible if done before it becomes embedded in the firmware or the first partitions of the hard drive. Ascertaining whether a system is infected with a bootkit is not easy; even if possible, the victim will face even greater difficulties removing it. If the bootkit was introduced into the first partitions of the hard disk (MBR, VBR, or, in the case of a UEFI-based device, EFI System Partition), a complete reinstallation of the OS will remove the malicious code from the drive. However, any reinstallation of the OS will not affect the chip memory where the BIOS or UEFI firmware resides, so if the firmware is changed, the new OS may still be infected.

You can also determine which particular bootkit infected the system, and check for utilities from antivirus vendors for cleaning the system of malicious code.

Evolution of bootkits

In the early 1980s, the first precursors of bootkits appeared in the form of viruses that infected the boot sector of the hard drive, known as boot sector infectors. After establishing themselves in the system, these malware programs attempted to spread to other devices, infecting all connected removable media. The most famous examples of this type of malware are Elk Cloner, one of the first viruses to target Apple computers, and Brain, which resulted in the first computer epidemic and prompted software developers to create the first antivirus. Another high-profile boot sector infector was the Stoned virus, which appeared in 1987. Its source code has formed the basis of many malware programs that infect the boot sector, such as Michelangelo, AntiExe, and Angelina. The latter was discovered in 2007 on Medion laptops sold in Germany and Denmark. The preinstalled Bullguard antivirus could only detect the infection, but not clean the system.

Full-fledged bootkits appeared in the early 2000s and were BIOS-oriented. One of the first bootkit PoCs was eEye BootRoot.

As part of our study, we considered both PoC bootkits and real-world bootkits found in the wild. PoC bootkits accounted for 31% of our sample, and in-the-wild bootkits 69%.

Bootkit malware for attacks on BIOS-based devices can be injected directly into the MBR, VBR, or IPL. A bootkit can also be embedded into the firmware itself, but in practice this is hard to do.

Among the bootkits we analyzed, 76% were designed for BIOS. The proportion of these affecting only the MBR was 80%. Another 10% are injected into the VBR or IPL, and the remaining 10% support all of the above infiltration methods.

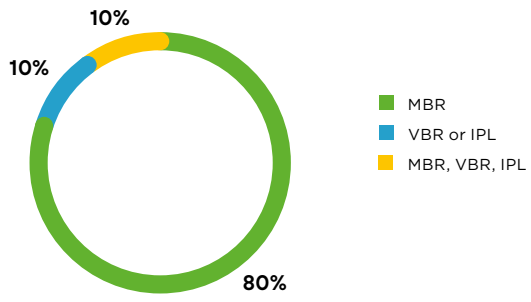

Figure 1. Bootkit types for BIOS attacks (by memory area)

Bootkits for BIOS attacks can compromise any system components, including the OS loader, hypervisor, and security tools. Intel withdrew support for BIOS back in 2020, but some companies cannot quickly update their IT infrastructure, hence BIOS infection bootkits are still live.

Why not everyone abandoned BIOS. Some organizations have difficulty upgrading their IT infrastructure. In Russia, according to our assessment, most often it is government institutions and industrial enterprises that encounter this problem. The issue is no less acute in the case of virtual infrastructure, since even hypervisor vendors recommend using BIOS as the default firmware. In our view, this is because BIOS-based virtual machines are easier to service and support.

2009 saw the arrival of a new version of the Stoned virus, which constituted a full-fledged bootkit PoC. A trait of this version was that the virus infected the system even if the hard drive or a partition was encrypted with TrueCrypt. That dispelled the myth that drive encryption protects against malware infection. Stoned proved otherwise. The malware got into the MBR, which, even in the case of drive encryption, always remains unencrypted. Then, when the user entered their password to use the device, the bootkit intercepted it, thus gaining access to all encrypted information. This was the first clear-cut case of bypassing drive encryption. A year later, on the basis of Stoned, the Whistler bootkit was developed and seen in real attacks. Incidentally, in 2011, the first Stoned-based bootkit PoC was also developed, focused on attacks on UEFI firmware.

The Stoned research project was so popular that antivirus companies requested not to publicly release the source code for new versions of it.

Having studied all the shortcomings of BIOS, device manufacturers switched to the more secure UEFI. Compared to BIOS, UEFI features a number of major improvements, but what interests us most is the Secure Boot protocol, which checks the signatures of UEFI drivers, UEFI applications, and the OS itself. If these signatures match the data in the repository of signatures of trusted applications and hash sums, the UEFI applications are loaded and the UEFI firmware hands over control to the OS. The repository of these signatures and hash sums is itself located in non-volatile memory and is populated by the device manufacturer. For more details about the Secure Boot protocol, see Microsoft's website.

Secure Boot is only activated if the intruder does not have physical access to the device; otherwise they can add or substitute signatures for their own malicious drivers.

Due to Secure Boot, the switch to UEFI should have made it impossible to introduce bootkits, but things turned out differently. Several ways can be used to infect UEFI firmware:

- Performing a supply chain attack by injecting a bootkit into the supplied software or updating the software
- Gaining physical access to the device
- Exploiting errors in the firmware configuration or update mechanism
- Remotely infecting the device; before this, the attacker elevates their privileges to install an OS kernel-level payload to execute code in System Management Mode (SMM), thus bypassing the various protection mechanisms in the firmware and gaining direct access to its memory

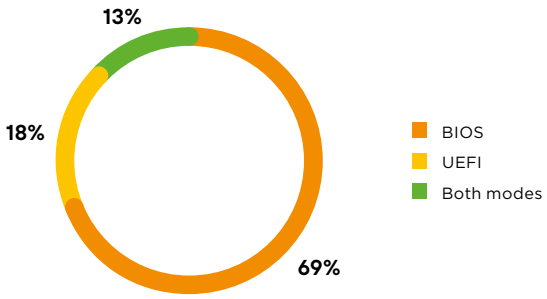

Figure 2. Bootkit types by targeted technology

In general, when infecting UEFI-based devices, attackers can infiltrate the firmware stored in SPI flash memory, make changes to an existing module, or create a new one in the EFI System Partition.

The **EFI System Partition (ESP)** is a special hidden partition on the hard drive of UEFI-based devices. This partition stores the boot manager. At device bootup, UEFI loads files (modules) from the ESP to start the OS and installed utilities.

After the release of the first bootkit PoC for UEFI infection, it took six years before the first bootkit was seen in the wild: LoJax in 2017. To overwrite the firmware in SPI flash, this bootkit's developers used both firmware vulnerabilities and gaps in the Secure Boot configuration. TrickBoot, part of the TrickBot malware that made headlines in 2020 and 2021, is also UEFI-oriented.

Since 2020, all bootkits found in the wild have targeted UEFI, in particular, Mosaic Regressor, TrickBoot, FinSpy, ESPecter and MoonBounce.

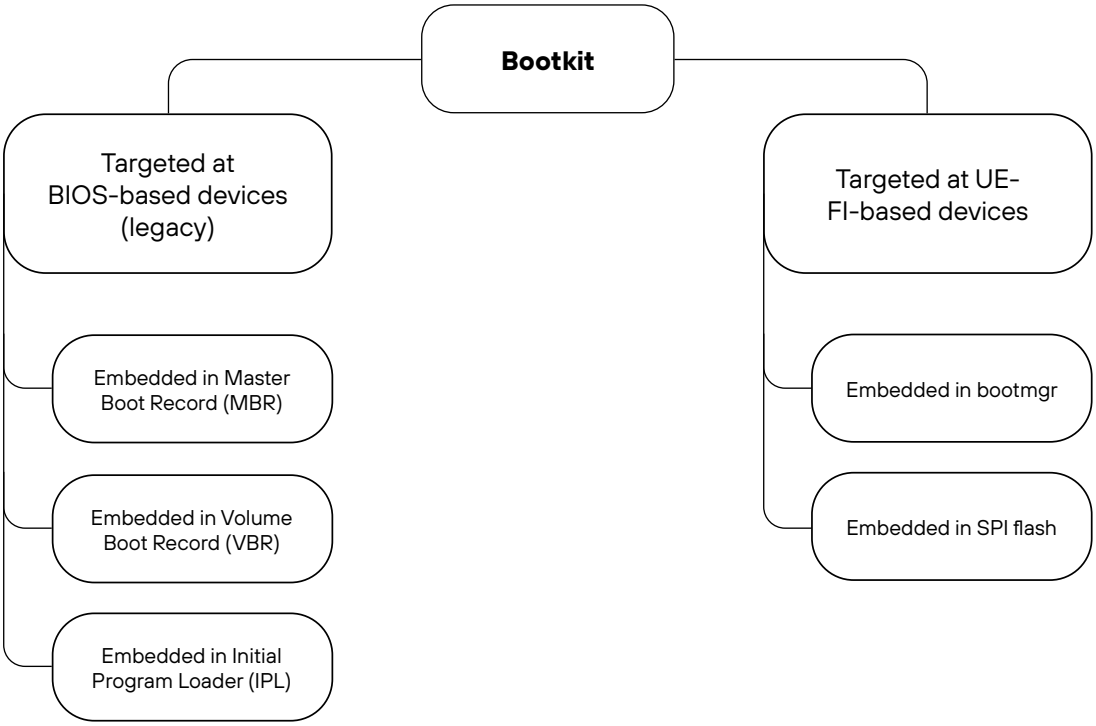

Figure 3. Bootkit classification

Some bootkits combine several techniques in targeting one type of firmware, for example, by embedding themselves in both the MBR and the VBR. One such bootkit is Gapz. There are also versatile bootkits for attacks on both BIOS- and UEFI-based devices, such as FinSpy and TrickBoot.

BOOTKITS

ptsecurity.com

Chronology

2005

- eEye BootRoot [B]

2007

- Mebroot [B]
- Vbootkit [B]

2008

- TDSS [B]

2009

- Stoned bootkit [B]
- Mebroot v2 [B]
- Vbootkit x64 [B]

2010

- Olmarik (TDL4/TDSS) [B]
- Mebratix [B]
- Whistler bootkit [B]

2011

- Rovnix [B]
- DeepBoot [B]
- Evil Core [B]
- Mebromi [B]
- Stoned Vienna [U]
- FispBoot/Fisp.A [B]
- Halcbot bootkit [B]
- Olmasco (MaxSS) [B]

2012

- DKFBootkit [B]
- XPAJ (Goblin) [B]
- Gapz [B]

2013

- Aduska bootkit [B]
- DreamBoot [U]

2014

- Pitou [B]
- OldBoot [B]

2015

- BOOTRASH [B]
- HDRoot [B]
- Thunderstrike [U]

2016

- Petya [B] [U]
- Satana [B]

2017

- LoJax [U]

2018

- DarkCloud bootkit [B]
- Pitou [B]

2019

- EfiGuard [U]

2020

- TrickBoot [B] [U]
- Mosaic Regressor [U]

2021

- MoonBounce [U]
- ESPecter [B] [U]
- FinSpy [B] [U]

Bootkit classification:

Bootkits used in the wild

Proof-of-Concept Bootkit

Bootkit in the wild:

[B] BIOS

[U] UEFI

Vulnerable firmware

Cybercriminals actively seek BIOS and UEFI vulnerabilities that allow bootkit injection. Analysts at Binarly have identified 23 critical vulnerabilities linked to SMM memory management in the UEFI firmware from InsydeH2O used by major hardware vendors, such as Bull (Atos), Dell, Fujitsu, HP, Intel, Lenovo, Microsoft, and Siemens. By exploiting these vulnerabilities, attackers were able to disable hardware security features and introduce a bootkit and malware for remote control. Analysts estimate that the flaws could have affected millions of devices, from laptops to servers, network equipment, and industrial control systems (ICS).

At the Black Hat Asia 2017 conference, experts from Cylance demonstrated how two vulnerabilities (CVE-2017-3197 and CVE-2017-3198) in Gigabyte UEFI firmware can be used to inject malware. Both vulnerabilities are component design flaws. The first is related to incorrect implementation of write protection, the second to the lack of component signature verification.

BIOS firmware also has vulnerabilities. At the end of March 2022, for instance, Dell advised customers to update the BIOS on Alienware, Inspiron, Vostro, and Edge Gateway 3000 series computers without delay. These models were exposed to vulnerabilities allowing a remote attacker to bypass authentication and use a system management interrupt (SMI) to execute arbitrary code when processing system functions (CVE-2022-24415, CVE-2022-24416, CVE-2022-24419, CVE-2022-24420, and CVE-2022-24421).

Bootkit for sale

We have already mentioned how hard it is for cybercriminals to develop a rootkit. Bootkits are even more complex. Design flaws can, for example, prevent the device from booting, which will result in an investigation and potential detection of the malware and the cybercriminals. Complicating matters is the fact that there is not much information about this type of malware online. Attackers use all available means:

- Upgrading bootkit PoCs, as happened, for example, with the Stoned bootkit, which was turned into a bootkit for Whistler attacks;
- Searching for developers able to create a bootkit from scratch;
- Buying ready-made solutions.

We analyzed 58 Telegram channels and ten of the most active Russian- and English-language dark-web forums with ads offering bootkits for sale and jobs for malware developers.

The average price of a bootkit for rent is USD 4,900. For comparison, a rootkit can be rented for USD 100–200.

USD 10,000 can buy the bootkit source code, and USD 2,000 a runnable image. Cybercriminals are willing to pay USD 3,000–5,000 to develop a bootkit for MBR infection. The maximum price they are ready to pay for a bootkit for UEFI firmware is USD 2 million.

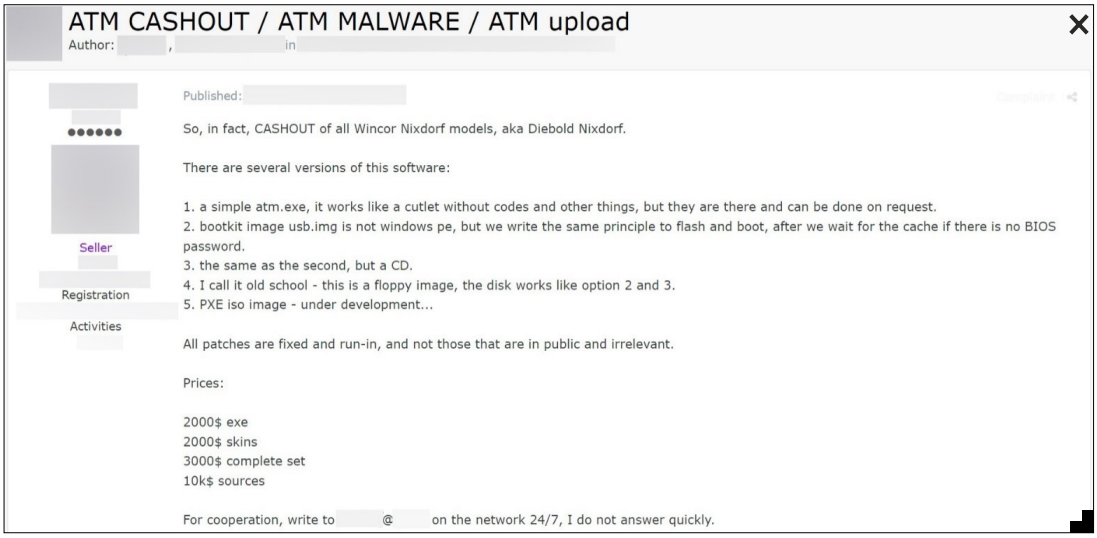

Figure 4. Ads selling a bootkit for ATM attacks

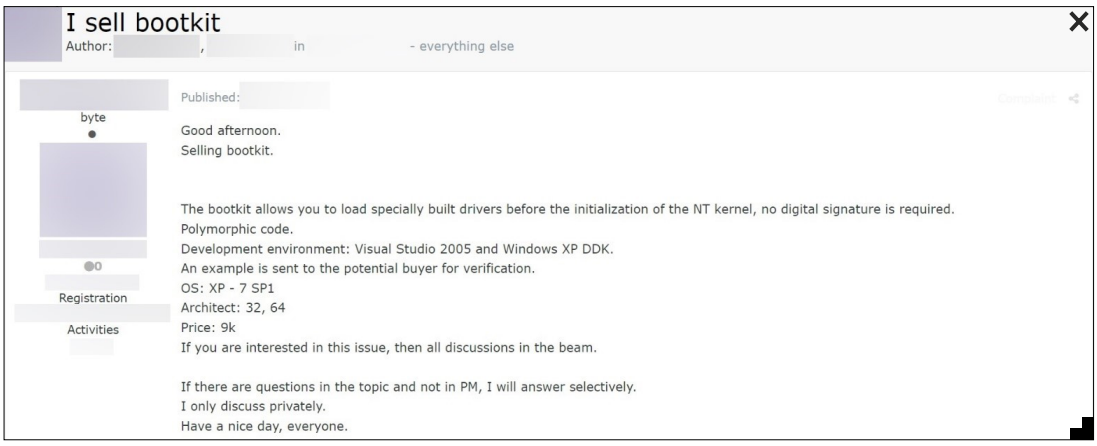

Figure 5. Ad selling a bootkit

We encountered messages in Telegram channels with attached archives containing the source code for bootkit PoCs and in-the-wild bootkits, allowing attackers to build a ready-made bootkit or use fragments of ready-made code in developing their own malware.

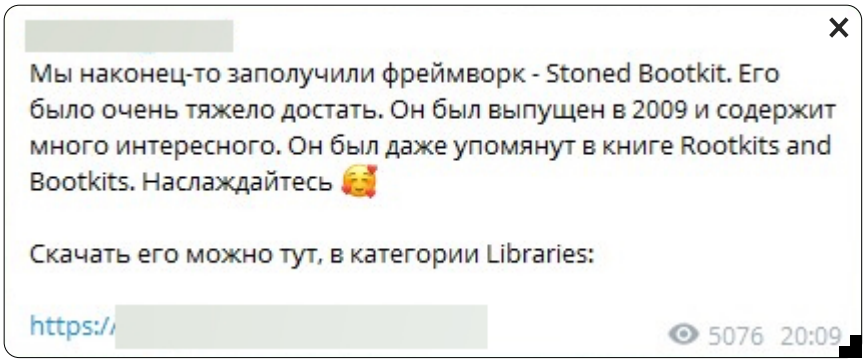

Figure 6. Post with a link to the source code for the Stoned Bootkit

Figure 7. Post with an archive containing the source code for the ESPecter bootkit

How to prevent bootkit infection

In our view, firmware research will result in the discovery of new vulnerabilities, and bootkits themselves will become increasingly common. As ever, prevention is better than a cure. With that in mind, we recommend:

- Monitoring potentially dangerous operations in the system: gaining direct access to the hard drive; installing a driver; reading the firmware;
- Enabling Secure Boot mode for UEFI, since if bootkit drivers are not digitally signed, this mode prevents them from running and thus infecting the system;
- Not booting the OS from untrusted media;
- When updating the OS version and firmware, checking for information about vendor compromise (so as not to fall victim to a supply chain attack).

Remember, too, the importance of detecting and countering malware downloaders and installers at an early stage, and use the latest antivirus tools and sandboxes to analyze the potential behavior of a file in the system before it is directly executed.

To detect infection, the integrity of boot records and firmware must be monitored.