R

Rootkit Malware: Types, Bootkit Differences, Detection, and Removal

A rootkit hides privileged malicious activity in an operating system, boot chain, or firmware. Learn the types, how bootkits differ, trusted detection methods, and when to rebuild.

PUBLISHER
Gridinsoft Support

UPDATED
Oct 8, 2026

READ TIME
6 min

CONTENTS

In this guide

01 Types of rootkits
02 Bootkit vs rootkit
03 How rootkits get installed
04 Possible signs of a rootkit
05 How rootkits are detected
06 What to do if a rootkit is detected
07 Why manual file deletion is unsafe
08 Rootkit prevention
09 Frequently asked questions
10 References

A rootkit is software that gives malware or an attacker stealthy, privileged access while hiding files, processes, drivers, connections, accounts, or other activity. The word describes concealment and control techniques, not a single malware family. Rootkits may operate in user space, the operating-system kernel, the boot chain, or device firmware.

A rootkit often supports another objective: credential theft, espionage, remote access, or persistence. Removing the visible payload is not enough if the component that hides or restores it remains. Because a successful rootkit can falsify information reported by the infected operating system, verification from a trusted environment is important.

01 Types of rootkits

Type	Where it operates	Why it is difficult
User-mode rootkit	Applications, libraries, or user-level API hooks	Can hide activity from ordinary tools but is generally below kernel privilege
Kernel or driver rootkit	Operating-system kernel or privileged drivers	Can intercept low-level operations and alter what security tools see
Bootkit	Bootloader, boot records, or early startup components	Runs before the main operating system and can influence its loading
Firmware rootkit	UEFI/BIOS or device firmware	May survive operating-system reinstallation and requires vendor-specific recovery
Virtualization-based rootkit	Hypervisor or virtualization layer	Attempts to observe or control the operating system from a lower layer

Firmware and hypervisor rootkits are uncommon compared with ordinary malware. Do not conclude that firmware is infected merely because a detection returns or Windows behaves strangely. More common causes include unresolved persistence, a downloader, a compromised account, or incomplete cleanup.

02 Bootkit vs rootkit

Rootkit is the broad category for stealth and privileged concealment. A **bootkit** is a rootkit that compromises the boot process so it runs before or while the operating system starts.

Question	Rootkit	Bootkit
Required location	May be user mode, kernel, boot, or firmware	Bootloader or another early boot component
Runs before the OS?	Not necessarily	Yes, by definition
Can an OS reinstall remove it?	Often, if all affected storage is replaced	Depends on whether boot partitions and boot records are rebuilt
Does Secure Boot help?	Helps with trusted boot components, not every rootkit type	Can prevent unauthorized boot components when correctly enabled and keys remain trusted

03 How rootkits get installed

- A phishing attachment, malicious installer, crack, or Trojan obtains initial execution.
- An attacker exploits a vulnerability to gain kernel or administrator privilege.
- A malicious or compromised signed driver is installed.
- Stolen administrator credentials are used to change boot or security settings.
- A supply-chain or firmware compromise affects a component before deployment.

Root access is not the delivery method by itself. Investigators need to identify both the initial access path and the privileged component that provides concealment.

04 Possible signs of a rootkit

- Security tools, logs, updates, or system utilities are disabled or return inconsistent results.
- A process, file, driver, port, or account appears in offline evidence but not in normal operating-system tools.
- Unknown drivers or boot changes fail signature or integrity checks.
- Security detections return after removing their visible files.
- Unexpected privileged activity, credential theft, or remote access continues after ordinary cleanup.
- Secure Boot becomes disabled or boot configuration changes without approval.

Crashes, high CPU, slow startup, and network traffic are not rootkit-specific. Hardware problems, drivers, updates, and ordinary malware are more common. Use evidence rather than assuming the deepest possible compromise.

05 How rootkits are detected

Method	What it can reveal
Offline or boot-time scan	Files and startup components before the suspected operating system can hide them
Integrity and signature checks	Unauthorized changes to drivers, boot files, protected components, and firmware measurements
Cross-view comparison	Differences between high-level system listings and lower-level disk, memory, or network evidence
Memory forensics	Hidden modules, hooks, injected code, kernel objects, and active connections
Secure Boot and measured boot telemetry	Whether expected trusted components participated in startup

Method	What it can reveal
Network and identity monitoring	Malicious activity that concealment on one endpoint cannot erase elsewhere

No single "rootkit scanner" can prove that every layer is clean. A security product can detect known artifacts and suspicious behavior, but confidence comes from multiple independent sources and a trusted recovery process.

06 What to do if a rootkit is detected

1. **Isolate the device.** Disconnect it from networks and do not use it for credentials, banking, or administration.
2. **Preserve the alert and evidence.** Record the detection, path, driver, hash, user, time, and related process. In an organization, collect volatile evidence before powering down when the response plan requires it.
3. **Use a trusted scanning environment.** Run the security vendor's supported offline or boot-time scan rather than relying only on tools inside the suspected OS.
4. **Assess privilege and persistence.** Determine whether the issue is a user-level component, kernel driver, bootkit, or suspected firmware compromise.
5. **Protect identities.** From a clean device, reset exposed credentials, revoke sessions and tokens, and replace compromised authentication methods.
6. **Rebuild when trust is lost.** For kernel or boot compromise, wipe relevant storage, recreate boot partitions, and reinstall from verified media.
7. **Handle firmware through the vendor.** Apply an authenticated firmware recovery or reflash procedure for the exact device. Replace hardware when integrity cannot be restored.
8. **Validate before restoration.** Update firmware and the OS, enable Secure Boot where supported, scan backups, and monitor for returning indicators.

07 Why manual file deletion is unsafe

A rootkit may intercept file operations, protect its driver, or recreate components through another payload. Deleting a driver without understanding its boot registration can also make the system unbootable. Follow the security vendor's documented remediation or rebuild from trusted media.

Restoring an old full-system image can reintroduce the compromise. Prefer data backups from a known-good date, scan them, and reinstall applications from verified sources.

08 Rootkit prevention

- Keep the operating system, firmware, drivers, VPNs, and security tools updated.
- Enable Secure Boot and hardware-backed boot protections where supported.
- Use driver-signing enforcement and block known vulnerable drivers.
- Limit local administrator rights and protect privileged accounts with strong MFA.
- Use application control and prevent untrusted installers, cracks, and kernel drivers.
- Monitor changes to boot configuration, firmware settings, security exclusions, and privileged services.
- Maintain offline or immutable backups and verified recovery media.

09 Frequently asked questions

Can antivirus remove a rootkit?

It can remove many known rootkits, especially with an offline scan. A trusted reinstall is often safer when kernel, boot, or firmware integrity cannot be established.

Will formatting the drive remove every rootkit?

It removes ordinary files on the formatted storage, but the boot layout must also be recreated, and firmware compromise would require separate remediation.

Does Secure Boot prove the computer is clean?

No. It protects the boot chain when correctly configured, but user-mode malware, validly signed vulnerable drivers, credential theft, and other attacks remain possible.

10 References

- Microsoft Defender for Endpoint: Rootkits
- Microsoft: Secure the Windows boot process

NEED MORE HELP?

We are here when the guide is not enough.

Visit the Gridinsoft Support Center for current product guidance or submit a support request with the details of your case.

Open this guide online at support.gridinsoft.com