U

UEFI Rootkit: Detection, Recovery, and Firmware Security

A UEFI rootkit or bootkit can execute before the operating system and survive ordinary reinstallation. Learn what evidence matters and why recovery requires vendor-approved firmware steps.

PUBLISHER
Gridinsoft Support

UPDATED
Oct 8, 2026

READ TIME
5 min

CONTENTS

In this guide

01 UEFI rootkit vs bootkit and ordinary rootkit

02 How attackers reach UEFI

03 Possible signs and stronger evidence

04 What to do when UEFI compromise is suspected

05 Safe recovery strategy

06 How to reduce UEFI rootkit risk

07 Frequently asked questions

08 Authoritative guidance

A UEFI rootkit is a malicious implant that compromises firmware or the pre-operating-system boot environment to gain stealthy, durable control. Because UEFI code runs before Windows or another operating system, a firmware implant can interfere with the boot chain and potentially restore an operating-system payload after ordinary cleanup.

True UEFI compromise is uncommon compared with ordinary malware. A persistent Windows detection, boot problem, or Secure Boot warning does not by itself prove a firmware rootkit. Investigation should establish evidence at the firmware or boot level before attempting recovery. Improper flashing or key changes can make a device unbootable.

01 UEFI rootkit vs bootkit and ordinary rootkit

Threat	Where it persists	Can OS reinstall remove it?
User-mode or kernel rootkit	Operating-system files, drivers, registry, or disk	Usually, if all disks and boot components are correctly rebuilt
Bootkit	Bootloader, EFI System Partition, or another boot-chain component	Not necessarily; the boot chain must also be repaired
UEFI firmware implant	Motherboard SPI flash or another firmware component	No; vendor-approved firmware recovery or hardware replacement may be required
Malicious option ROM or peripheral firmware	An expansion device or hardware controller	No; the affected component must be addressed

Reports sometimes use *UEFI rootkit* and *UEFI bootkit* interchangeably. The exact persistence location matters more than the label.

02 How attackers reach UEFI

- Exploitation of a firmware, bootloader, update, or Secure Boot validation vulnerability.

- Administrative or kernel-level access used to write insufficiently protected firmware.
- Physical access, a malicious peripheral, or abuse of vendor maintenance features.
- A compromised firmware update or supply-chain process.
- Misconfiguration, disabled Secure Boot, unprotected flash regions, or outdated revocation data.

Most scenarios require substantial prior access or a vulnerable component. UEFI persistence is usually a later stage of a serious intrusion, so responders must also identify how the attacker first obtained privilege.

03 Possible signs and stronger evidence

Symptoms such as slow boot, crashes, changed boot order, or malware returning are nonspecific. Stronger evidence includes a trusted firmware scanner alert, measured-boot or attestation anomalies, an unknown EFI executable, firmware regions differing from a vendor-approved image, Secure Boot database changes, or documented behavior matching a known bootkit.

- Security software reports a UEFI or bootkit detection with a specific component and measurement.
- Secure Boot or measured-boot validation changes unexpectedly.
- An operating-system payload returns after a verified clean-disk reinstall.
- Firmware version, hashes, variables, or boot entries differ from an approved baseline.
- Enterprise telemetry links the device to an intrusion that used known firmware techniques.

A firmware version shown in setup is not enough to prove integrity, and an antivirus scan inside the operating system may not have full access to every firmware region.

04 What to do when UEFI compromise is suspected

1. **Isolate the device.** Disconnect it from networks and sensitive peripherals. Preserve power state when an incident responder needs volatile evidence.

2. **Escalate appropriately.** Contact the organization's security team and the device manufacturer. Firmware incidents are not suitable for trial-and-error consumer cleanup.
3. **Preserve evidence.** Record model, serial number, motherboard revision, firmware version, Secure Boot state, TPM and attestation results, alerts, boot entries, disks, and timestamps.
4. **Use trusted external assessment.** Compare firmware and boot artifacts with vendor-approved baselines using supported enterprise or forensic tooling.
5. **Protect identities.** Revoke sessions and rotate credentials and hardware-backed secrets when the intrusion may have captured them.
6. **Scope the intrusion.** Hunt for the initial entry, privilege escalation, lateral movement, and other affected devices.

05 Safe recovery strategy

Follow the computer or motherboard manufacturer's exact recovery procedure for the specific model and hardware revision. Use firmware obtained from the official support channel and verify its integrity where the vendor provides a signature or checksum. Depending on the evidence, recovery may require:

- A vendor-supported capsule update or hardware recovery mode that rewrites all relevant regions.
- Restoring approved Secure Boot keys and installing current revocation updates.
- Recreating the EFI System Partition and reinstalling the operating system from trusted media.
- Resetting or reprovisioning the TPM and replacing device certificates after key impact is assessed.
- Replacing the motherboard or affected peripheral when firmware cannot be verified or fully rewritten.

Do not download random firmware tools, disable signature checks, erase the TPM, or program the SPI flash without a validated plan. TPM clearing can permanently remove access to encrypted data. Back up required recovery keys and evidence first.

06 How to reduce UEFI rootkit risk

- Install firmware and operating-system security updates from the manufacturer.
- Enable Secure Boot with approved keys and current revocation data where the platform supports it.
- Use TPM-backed measured boot and attestation on managed devices.
- Require firmware-update authentication and protect setup configuration with organizational controls.
- Apply least privilege, credential protection, and endpoint controls to prevent the attacker reaching firmware-write capability.
- Inventory firmware versions and retire hardware that no longer receives security updates.
- Control physical access and untrusted peripherals for high-risk systems.

07 Frequently asked questions

Will formatting the drive remove a UEFI rootkit?

Not if the implant is in motherboard or peripheral firmware. Formatting can remove disk-resident components, but recovery must address every verified persistence location.

Does Secure Boot prevent every UEFI rootkit?

No. It raises the barrier by validating boot components, but vulnerabilities, stolen signing trust, outdated revocations, configuration changes, or firmware compromise can undermine it.

Should I update the BIOS after any malware infection?

Routine malware rarely requires firmware recovery. Install normal vendor security updates, but do not treat flashing as a generic removal step without evidence and model-specific instructions.

08 Authoritative guidance

- Microsoft: UEFI scanning in Defender for Endpoint
- Microsoft: Secure the Windows boot process
- CISA: BlackLotus UEFI bootkit guidance

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