

How to Fix CVE-2026-97631: Critical EDK II UEFI Vulnerability

TechRounder PDF Edition

Live article: <https://www.techrounder.com/how-to/how-to-fix-cve-2026-97631-critical-edk-ii-uefi-vulnerability/>

By Vipin PG | Published October 28, 2026 | Updated October 28, 2026 | Format: Guide | 17 min read

Quick answer

A critical security vulnerability in TianoCore EDK II could allow an attacker to execute malicious code before Windows, Linux, or another operating system starts. Identified as CVE-2026-97631, the issue affects the UEFI PXE network-boot driver and carries a CVSS 4.0 severity score of 9.5 out of 10.

A critical security vulnerability in TianoCore EDK II could allow an attacker to execute malicious code before Windows, Linux, or another operating system starts. Identified as CVE-2026-97631, the issue affects the UEFI PXE network-boot driver and carries a CVSS 4.0 severity score of 9.5 out of 10.

The vulnerability becomes relevant when a computer or server boots over a network. An attacker capable of delivering a specially crafted ICMP error message to the affected firmware during PXE boot may cause memory corruption, interrupt the boot process, or potentially execute arbitrary code.

TianoCore developers have already merged a correction into the EDK II source code. However, computers using affected firmware still need an updated BIOS or UEFI release from their manufacturer. Virtual machines using EDK II-based OVMF firmware may also need an updated firmware package.

For systems that do not require network boot, disabling PXE is a useful immediate precaution. Servers that depend on PXE for deployment or recovery need a different approach.

This guide explains the vulnerability, how to check your system's exposure, and the available fixes for Windows PCs, Linux machines, physical servers, and virtualization environments such as Proxmox.

Last reviewed: October 28, 2026. Firmware patch availability may change as vendors release updates.

What is CVE-2026-97631?

CVE-2026-97631 is a buffer-overflow vulnerability in the PXE Base Code driver included with TianoCore EDK II, an open-source framework used to develop UEFI firmware.

UEFI, or Unified Extensible Firmware Interface, initializes hardware and prepares a computer to load its operating system. It runs before Windows or Linux takes control of the machine.

EDK II provides many of the components firmware developers need to implement UEFI. Some computer manufacturers use EDK II-derived code in their firmware, while virtualization platforms such as QEMU/KVM can use EDK II through OVMF.

The affected component is located at:

NetworkPkg/UefiPxeBcDxe

This driver supports PXE, or Preboot Execution Environment. PXE allows computers to obtain boot resources over a network instead of relying entirely on a local storage device.

Organizations commonly use PXE when installing operating systems on multiple machines, provisioning servers, or performing network-based recovery.

According to the official TianoCore security advisory, the vulnerability occurs because the driver does not correctly handle certain ICMP error messages during network boot.

ICMP is a network protocol used to communicate errors and other network conditions. The vulnerable code processes these messages using a fixed-size buffer, but incorrect memory-copy calculations can allow data to be written beyond the buffer's allocated space.

Technical details of the vulnerability

Property: CVE ID | Details: CVE-2026-97631

Property: GitHub advisory | Details: GHSA-99cj-7hg9-mp4m

Property: Severity | Details: Critical

Property: CVSS 4.0 score | Details: 9.5

Property: Affected software | Details: TianoCore EDK II

Property: Affected component | Details: NetworkPkg/UefiPxeBcDxe

Property: Vulnerability type | Details: Buffer overflow and out-of-bounds write

Property: Attack surface | Details: PXE network boot over IPv4 or IPv6

Property: Possible impact | Details: Boot failure or pre-OS code execution

Property: Public disclosure | Details: October 22, 2026

Property: Upstream patch | Details: Merged October 15, 2026

The weakness is associated with CWE-120, CWE-131, and CWE-787. These classifications describe inadequate buffer-size checks, incorrect buffer calculations, and out-of-bounds memory writes.

How does the EDK II UEFI vulnerability work?

To understand the issue, consider what happens when a computer starts through PXE.

Instead of loading its operating system directly from an internal SSD, the computer initializes its network adapter and communicates with the network-boot infrastructure. UEFI handles this communication before the operating system has started.

The PXE driver must process various network messages, including ICMP errors. The vulnerable implementation stores ICMP error information in a memory area called 'Mode.IcmpError'.

That memory area has a capacity of 504 bytes.

The problem lies in the way the driver calculates offsets and copies incoming data into this buffer. The original implementation can write beyond the allocated space when processing a specially constructed message.

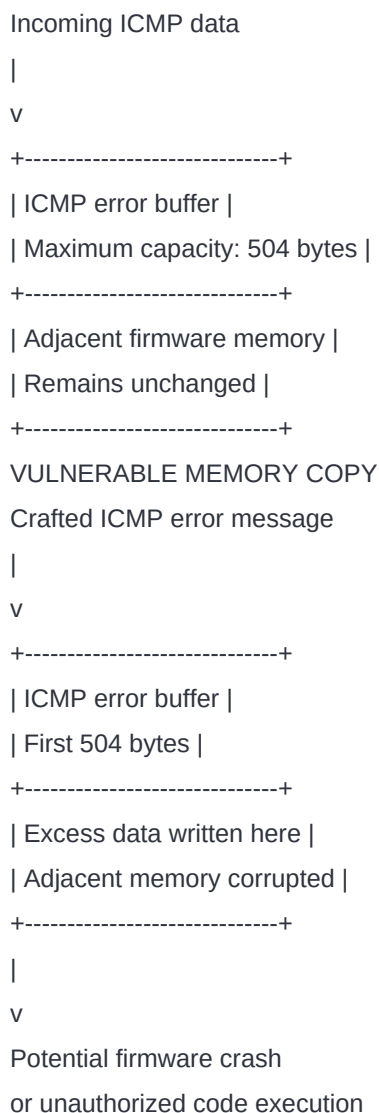
Understanding the 504-byte buffer overflow

Imagine that the firmware has reserved exactly 504 bytes for ICMP error data. A normal message fits inside that memory area without affecting anything around it.

An oversized message should be rejected or truncated to fit the available space. In the vulnerable implementation, incorrect calculations can allow the copy operation to extend beyond that boundary.

The memory layout can be represented as follows:

EXPECTED MEMORY COPY



This diagram simplifies the memory handling, but it shows why a buffer overflow can be dangerous. The additional data may overwrite structures that the firmware depends on while executing.

In this case, the affected code also incorrectly advances the destination pointer while copying fragments. It uses an accumulated length where the length of the current fragment is required.

The two affected functions are:

- 'PxeBcIcmpErrorDpcHandle' , which handles IPv4 ICMP errors.
- 'PxeBcIcmp6ErrorDpcHandle' , which handles IPv6 ICMP errors.

Both functions are addressed by the official EDK II correction.

What happens if the vulnerability is exploited?

The immediate result of an unsuccessful or disruptive exploit may be a firmware exception. This can leave the computer frozen during network boot or prevent it from completing the startup process.

The more serious possibility is arbitrary code execution within the DXE environment.

DXE stands for Driver Execution Environment. It is a stage of UEFI initialization in which firmware drivers run and prepare the system for later boot operations.

The TianoCore advisory explains that an attacker with sufficient knowledge of the target firmware layout may be able to overwrite event-related structures and redirect execution through the DXE core.

Such an attack would take place before the operating system's normal security tools are available.

Reliable code execution is more complicated than causing a crash. The attacker may need information about the exact firmware implementation and memory layout. The vulnerability is still rated critical because successful exploitation could provide control at a highly privileged stage of the boot process.

Which computers and servers are affected?

A computer does not become vulnerable to CVE-2026-97631 simply because it uses UEFI.

The firmware must contain the affected EDK II PXE implementation or equivalent vulnerable code. The documented attack also requires the relevant network-boot driver to be active and reachable.

For example, a desktop computer that starts directly from an internal SSD and never uses PXE has much less exposure to this attack path than a server that regularly boots through the network.

That does not automatically establish that the desktop's firmware is patched. The vulnerable code could still be present even if the feature is not currently used.

System: Windows desktop or laptop | Exposure considerations: Limited exposure during normal local-disk boot if PXE is not used.

System: Linux desktop | Exposure considerations: Similar to Windows; firmware and PXE configuration determine exposure.

System: Enterprise server | Exposure considerations: Relevant when using network-based provisioning or recovery.

System: Bare-metal deployment server | Exposure considerations: Potential exposure during PXE installation workflows.

System: QEMU/KVM virtual machine | Exposure considerations: Depends on the OVMF firmware and network-boot configuration.

System: Proxmox virtual machine | Exposure considerations: Requires checking the VM firmware type and host firmware package.

System: LXC container | Exposure considerations: Does not run an independent UEFI boot process.

Firmware vendors such as AMI, Insyde, and Phoenix may use EDK II-derived components, but this does not mean every firmware release from those vendors contains the vulnerability.

The manufacturer, firmware version, included network drivers, and boot configuration all need to be considered.

Intel also told Ars Technica in its reporting on the vulnerability that none of its supported commercial products were known to be affected at the time. However, OEMs may incorporate EDK II code independently, so that statement should not be treated as confirmation for every Intel-based computer.

Is there an official fix for CVE-2026-97631?

Yes. TianoCore merged a source-code correction on October 15, 2026, through pull request #13302 in the EDK II GitHub repository.

The public security advisory followed on October 22, 2026.

There is an inconsistency worth mentioning in the official records. The security advisory's patched-version field still indicates that a patched version is unavailable, while the linked pull request confirms that the corrective code was merged.

The repository history establishes that the upstream source code has been fixed. It does not confirm that all existing BIOS, UEFI, and OVMF firmware builds have received the update.

What did the developers change?

The correction changes how the IPv4 and IPv6 ICMP handlers calculate destination offsets and handle incoming data fragments.

The updated implementation applies the correct buffer boundaries and truncates data that would otherwise exceed the available space.

It also clears the relevant buffer before copying data and includes tests for fragment handling, truncation, and memory integrity.

According to the upstream patch discussion, the added tests failed against the vulnerable implementation and passed after the correction.

For firmware developers, the patch is available to integrate into maintained EDK II branches. For ordinary computer users, the corrected source code must first be included in an officially supported firmware update.

How to fix CVE-2026-97631

The appropriate fix depends on the type of system. Physical computers generally need an update from their BIOS or UEFI manufacturer, while virtual machines may receive corrected firmware through virtualization packages.

If you do not use PXE, disabling network boot is a sensible interim measure.

Method 1: Disable PXE network boot in BIOS or UEFI

This is the simplest precaution for a computer that always boots from an internal SSD or hard drive.

Disabling unused PXE functionality prevents the normal network-boot process from activating the affected driver.

Step 1: Open BIOS or UEFI setup

Restart the computer and press its firmware setup key during startup.

Common keys include 'F2', 'Delete', 'F10', and 'Esc'. The correct key depends on the computer or motherboard manufacturer.

On Windows 11, you can also open Settings > System > Recovery and select the advanced startup option.

Once the recovery menu appears, open the UEFI firmware settings option if your computer provides it.

Step 2: Locate the network-boot settings

Look under the Boot, Advanced, Network Configuration, or Integrated Devices section.

The setting may appear under one of these names:

- PXE Boot
- Network Boot
- Boot to LAN

- IPv4 PXE Support
- IPv6 PXE Support
- UEFI Network Stack

Some systems provide separate IPv4 and IPv6 network-boot options.

Step 3: Disable unused PXE options

If the computer does not need network boot, disable the available PXE options. Where IPv4 and IPv6 PXE controls are separate, disable both.

Disabling IPv6 alone does not address the full attack surface because the vulnerability affects both protocol handlers.

Step 4: Check the boot order

Make sure the internal system drive is the primary boot device.

For a typical Windows computer, Windows Boot Manager should normally appear before any network-boot entry.

A Linux system may show an entry for its installed bootloader or distribution.

Step 5: Save and restart

Save the configuration changes and restart the computer. Confirm that Windows or Linux starts normally.

Disabling PXE should not interfere with ordinary Ethernet or Wi-Fi connectivity after the operating system loads.

Keep in mind that this method reduces exposure to the documented PXE attack. It does not remove vulnerable code from the firmware itself.

Method 2: Install an updated BIOS or UEFI firmware

If the manufacturer confirms that your firmware contains the vulnerable driver, installing a corrected firmware release is the preferred permanent solution.

Before downloading anything, identify the exact system model and installed firmware version.

Step 1: Check the BIOS version on Windows

Open PowerShell and run:

```
Get-CimInstance Win32_BIOS |  
Select-Object Manufacturer, SMBIOSBIOSVersion, ReleaseDate
```

To identify your system model, run:

```
Get-CimInstance Win32_ComputerSystem |  
Select-Object Manufacturer, Model
```

Record the manufacturer, model, and BIOS version.

Step 2: Check the firmware version on Linux

On Linux, you can use:

```
sudo dmidecode -t bios
```

Alternatively, the following commands read the firmware information exposed through the system:

```
cat /sys/class/dmi/id/bios_vendor
```

```
cat /sys/class/dmi/id/bios_version
```

```
cat /sys/class/dmi/id/bios_date
```

```
cat /sys/class/dmi/id/product_name
```

These commands identify the installed firmware. They cannot independently determine whether the vulnerable EDK II driver is present.

Step 3: Check the manufacturer's firmware updates

Visit the official support website for your computer, motherboard, or server.

Search using the exact model number and review the available BIOS or UEFI downloads.

Look for references to CVE-2026-97631, EDK II PXE security fixes, ICMP handling, or NetworkPkg/UefiPxeBcDxe.

A firmware release may incorporate the correction without naming the CVE in its public release notes. If the documentation is unclear, request confirmation from the manufacturer.

Step 4: Prepare for the update

Back up important files and follow the manufacturer's update instructions.

If the computer uses BitLocker or another encryption mechanism linked to TPM measurements, make sure the recovery key is available. Some firmware updates require temporarily suspending protection according to the manufacturer's guidance.

Use a reliable power source and avoid interrupting the update.

Step 5: Install and verify the firmware

Use the manufacturer's supported update utility or firmware-management interface.

After the update, check the BIOS version again and compare it with the installed release.

For confirmation that CVE-2026-97631 is fixed, refer to the manufacturer's security advisory or documented affected and corrected versions.

Do not install a generic EDK II firmware image on a physical computer. Firmware images are hardware-specific, and flashing an incompatible image can prevent the device from starting.

Method 3: Protect servers that require PXE boot

Disabling PXE may not be practical for servers that use network provisioning, automated operating-system deployment, or recovery tools.

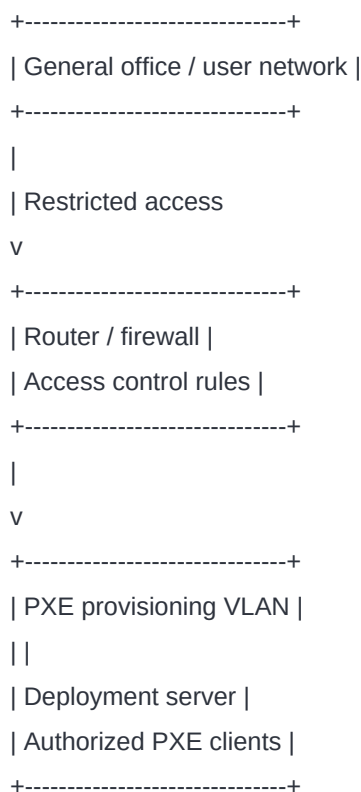
In these environments, the immediate priority is to limit which systems can communicate with PXE clients while their firmware network stack is active.

Separate the provisioning network

Where possible, use a dedicated VLAN or isolated network for provisioning traffic.

For example:

NETWORK SEGMENTATION



This arrangement limits the number of devices that can reach computers during network boot.

Review routing and firewall rules to ensure that untrusted networks cannot send unnecessary traffic to PXE clients.

Review ICMP traffic controls

The vulnerability involves ICMP error handling, so administrators may consider restricting unexpected ICMP and ICMPv6 traffic reaching the provisioning environment.

However, blocking every ICMP message is not advisable. Some ICMP traffic is necessary for normal network operation, including path MTU discovery.

Any filtering policy should be designed around the deployment environment and tested before it is applied broadly.

Maintain a firmware inventory

Record the device manufacturer, model, installed firmware version, PXE configuration, and available security updates.

This helps identify systems that need a firmware update and distinguishes them from machines that are unaffected or no longer use PXE.

These network protections reduce exposure while waiting for an official fix. They should not be considered a replacement for corrected firmware.

Method 4: Update OVMF firmware on QEMU, KVM, and Proxmox

Virtual machines need separate attention because their UEFI firmware may be provided by the virtualization host.

QEMU/KVM commonly uses OVMF, an EDK II-based UEFI firmware implementation.

A virtual machine using affected OVMF firmware may therefore have the same vulnerable PXE component, even if its guest operating system is fully updated.

Check OVMF packages on Debian or Ubuntu

On a Debian-based virtualization host, run:

```
dpkg -l | grep -E 'ovmf|edk2|qemu-efi'
```

Check the available OVMF version:

```
apt-cache policy ovmf
```

You can also inspect the package changelog:

```
apt changelog ovmf
```

Before applying an update, verify whether the distribution has released a package containing the CVE-2026-97631 correction.

If a corrected version is available through the supported repositories, update the package using:

```
sudo apt update
```

```
sudo apt install --only-upgrade ovmf
```

The Debian security tracker for CVE-2026-97631 listed the vulnerability as unresolved for its tracked EDK II packages in Bookworm, Trixie, Forky, and Sid on October 28, 2026.

Check the tracker again before assuming that an ordinary package upgrade includes the fix.

Check EDK II firmware on Proxmox

On a Proxmox VE host, start by checking the installed version and packages:

```
pveversion -v
```

Then inspect firmware-related packages:

```
dpkg -l | grep -E 'pve-edk2-firmware|ovmf|edk2'
```

To examine the configuration of a particular VM, run:

```
qm config VMID
```

Replace 'VMID' with the actual virtual machine ID.

Look for an entry such as:

```
bios: ovmf
```

This indicates that the VM uses OVMF firmware.

If the VM uses SeaBIOS instead, it follows a different firmware implementation, and exposure to this specific EDK II driver should not be assumed.

Also check whether the VM boots from a virtual disk or uses network boot.

For an affected OVMF implementation, install the corrected firmware through the supported Proxmox repositories when it becomes available.

After updating the host's firmware package, fully shut down and start the relevant virtual machines so they can load the updated firmware image. A reboot performed only inside the guest operating system may not replace the firmware instance already in use.

Avoid manually copying firmware images from unrelated distributions into a Proxmox installation.

Check Fedora OVMF updates

Fedora has already recorded the correction in its Rawhide EDK II package history.

The Fedora Rawhide edk2-ovmf changelog records the CVE-2026-97631 fix on October 26, 2026, in build '20260812-12.fc46'.

This confirms that the correction entered the tracked Rawhide package lineage. It does not establish that all stable Fedora releases have received the same fix.

To check the installed package:

```
rpm -q edk2-ovmf
```

Once an appropriate corrected update is available for your release, use:

```
sudo dnf upgrade edk2-ovmf
```

Review the package changelog to confirm the CVE correction is included.

Method 5: Apply the upstream patch to custom EDK II firmware

Developers maintaining their own EDK II firmware builds can integrate the upstream correction into their source tree.

The affected source file is:

```
NetworkPkg/UefiPxeBcDxe/PxeBcSupport.c
```

The official patch and code review contain the corrected buffer handling and related tests.

Firmware maintainers should incorporate the appropriate commits into their maintained branch, build the firmware for the target platform, and complete regression testing before deployment.

Copying the patched C source file onto an installed operating system will not repair the machine's firmware. The correction must be compiled into and deployed as part of the appropriate firmware image.

Current CVE-2026-97631 patch status

The source-code fix is available, but downstream firmware updates are being released at different times.

The following statuses reflect the published advisories and package trackers reviewed on October 28, 2026.

Platform: TianoCore EDK II | Status: Upstream correction merged October 15, 2026.

Platform: Debian EDK II | Status: Tracked releases listed as vulnerable.

Platform: Fedora Rawhide OVMF | Status: Fix recorded October 26, 2026.

Platform: SUSE | Status: Overall vulnerability state listed as pending.

Platform: Amazon Linux 2 | Status: EDK II package listed as pending fix.

Platform: Amazon Linux 2023 | Status: Relevant EDK II package classified as not affected.

Platform: Amazon Linux 2027 Preview | Status: Relevant EDK II package classified as not affected.

Platform: Physical PC or server firmware | Status: Depends on the manufacturer and firmware release.

Platform: Proxmox VE | Status: Requires verification of installed firmware package and release.

For distribution-specific information, consult the SUSE vulnerability tracker and the Amazon Linux CVE-2026-97631 advisory.

These package-level classifications should not be interpreted as assessments of every physical server running the corresponding operating system.

Does Secure Boot prevent CVE-2026-97631?

Secure Boot does not directly prevent exploitation of this vulnerability.

Secure Boot verifies the authenticity of certain boot components to help prevent unauthorized software from loading during the boot process.

CVE-2026-97631 affects firmware code that handles network traffic during the DXE phase, before the operating system takes control.

An attacker who successfully exploits the memory corruption may gain code execution in the firmware environment before ordinary operating-system security software is active.

Secure Boot and TPM-based protections remain useful security features, but neither corrects the defective ICMP handling code.

Reinstalling Windows or Linux also does not repair vulnerable UEFI firmware.

Can CVE-2026-97631 be exploited remotely?

The TianoCore advisory describes a remotely exploitable issue that does not require prior authentication.

However, the attacker must be able to reach the vulnerable PXE network-boot environment while the affected code is running.

A computer connected to the internet is not automatically exposed through this flaw during normal local-disk operation.

Practical exploitation depends on the firmware implementation, network configuration, and the ability to deliver malicious ICMP messages during the vulnerable stage.

The advisory also distinguishes between denial of service and reliable code execution. Causing a firmware exception can be easier than achieving controlled execution, which may require detailed knowledge of the target firmware layout.

Is the vulnerability being actively exploited?

The public material reviewed for this article did not establish confirmed exploitation in the wild as of October 28, 2026.

The vulnerability was reported by Alex Matrosov of Anthropic and Dan Holmes-Mitra of Google, according to the official TianoCore advisory.

Public disclosure makes the technical details available for further research, but it does not establish that attacks are already occurring against deployed systems.

Administrators should still review systems that depend on PXE boot, especially those with provisioning networks accessible from less-trusted environments.

How to confirm whether your computer is protected

There is no universal Windows or Linux command that can determine whether every manufacturer's firmware contains the corrected EDK II code.

A BIOS version check helps identify the installed firmware, but the version needs to be compared with reliable vendor information.

Use the following process:

1. Identify the exact computer or server model and installed firmware version.
2. Check whether the manufacturer has confirmed that the affected EDK II PXE driver is included.
3. Determine whether PXE network boot is enabled or used during deployment and recovery.
4. Look for a vendor advisory identifying an affected version and a corrected release.
5. Apply the supported update, then verify that the installed release includes the fix.

For virtual machines, check the firmware package on the virtualization host rather than relying solely on packages installed inside the guest OS.

If the vendor has not confirmed whether a particular build is affected, record its status as unknown rather than assuming it is protected.

Frequently asked questions

Does CVE-2026-97631 affect Windows 11?

It can affect the firmware of a computer running Windows 11 if the device includes the vulnerable EDK II PXE implementation. The issue is in UEFI firmware, not the Windows operating system itself.

Are all UEFI computers vulnerable?

No. The vulnerability affects specific EDK II PXE code. Firmware implementations that do not include the vulnerable driver are not affected by this particular defect.

Will disabling IPv6 fix the vulnerability?

No. Both the IPv4 and IPv6 ICMP handlers are affected. Disabling IPv6 alone leaves the IPv4 attack path available when PXE boot is used.

Does setting the SSD as the first boot device solve the problem?

It reduces the likelihood of entering network boot during normal startup, but it does not remove vulnerable firmware code. Disabling unused PXE options provides a stronger interim mitigation.

Can the vulnerability affect a running Windows or Linux computer?

The documented attack targets the firmware's PXE boot process. It is not described as an ordinary post-boot network vulnerability affecting Windows or Linux applications.

Can antivirus software block this attack?

Traditional antivirus software runs after the operating system has loaded. It cannot be relied upon to prevent exploitation of vulnerable firmware code during PXE boot.

Should I update my BIOS if I never use network boot?

Follow the manufacturer's security guidance. If your device is confirmed to be affected and a corrected firmware release is available, installing that update is advisable even when PXE is currently disabled.

Is the GitHub patch enough to fix an existing computer?

The upstream patch corrects the affected source code. An existing computer needs the correction included in the firmware it actually runs, usually through an official manufacturer update.

Conclusion

CVE-2026-97631 affects ICMP error processing in the EDK II PXE network-boot driver. A specially constructed network message can cause an out-of-bounds memory write, potentially crashing the firmware or enabling code execution before the operating system starts.

The TianoCore source-code correction was merged on October 15, 2026, but firmware and virtualization vendors need to distribute the updated implementation through their own releases.

For computers that do not need PXE, disabling network boot is a practical precaution. Systems that regularly use PXE should restrict access to their provisioning networks and install a verified firmware update when it becomes available.

To check your system's status, start with its firmware version, boot configuration, and the manufacturer's security advisory. For virtualization environments, include the host's OVMF package in that assessment.

References

1. github.com - tianocore / edk2 -
<https://github.com/tianocore/edk2/security/advisories/GHSA-99cj-7hg9-mp4m>
2. github.com - tianocore / edk2 - <https://github.com/tianocore/edk2/pull/13302>
3. arstechnica.com - security / 2026 -
<https://arstechnica.com/security/2026/10/code-execution-flaw-in-uefi-firmware-threatens-most-servers-on-planet-earth/>
4. security-tracker.debian.org - tracker / CVE-2026-97631 -
<https://security-tracker.debian.org/tracker/CVE-2026-97631>
5. packages.fedoraproject.org - pkgs / edk2 -
<https://packages.fedoraproject.org/pkgs/edk2/edk2-ovmf/fedora-rawhide.html>
6. suse.com - security / cve - <https://www.suse.com/security/cve/CVE-2026-97631.html>
7. explore.alas.aws.amazon.com - CVE-2026-97631.html -
<https://explore.alas.aws.amazon.com/CVE-2026-97631.html>