TUTORIAL

How to Configure Windows Server 2025 Secure Boot and TPM

Configure Secure Boot and TPM on Windows Server 2025 with a practical workflow: check prerequisites, enable firmware settings, validate attestation signals, and confirm production readiness.

Operating Systems - July 13, 2026

What you are building

Secure Boot and TPM are foundational firmware protections, but they only help if the server is configured correctly and the deployment path preserves those settings. In practice, many Windows Server 2025 rollouts fail here because firmware defaults are inconsistent, the machine boots in legacy mode, or TPM is present but not enabled, activated, or visible to the operating system.

This tutorial shows how to configure a Windows Server 2025 machine so it boots with Secure Boot enabled and a usable TPM in place, then validates that the system is actually in the expected state before production use. By the end, you will know how to check prerequisites, change firmware settings safely, verify the result from Windows, and confirm whether the server is ready for security features that depend on those controls.

Prerequisites and stop-here warnings

Before changing anything, confirm that the platform can support the target state. Not every server will.

Goal

Avoid partial configuration: Secure Boot without UEFI, TPM present but disabled, or a firmware state that breaks boot after a settings change.

Action

Check the following before you proceed:

- The server firmware supports UEFI boot mode.
- The motherboard or virtual machine exposes TPM 2.0.
- You have console access, iLO/DRAC/IMM/VM console, or another out-of-band recovery path.
- You know how to restore firmware defaults if boot fails.
- If BitLocker or other protector-based encryption is already enabled, you have the recovery material and maintenance window you need.

If the machine currently boots in Legacy BIOS mode, stop here and plan a controlled migration to UEFI first. Secure Boot depends on UEFI. If the firmware does not offer TPM 2.0, stop here as well; do not assume Windows can ?add? a TPM in software.

If this server is part of a hardened build process, align this work with your baseline workflow so you do not later undo the firmware state during general hardening. A controlled deployment approach such as Deploy Windows Server 2025 with Secure Baseline Hardening can help keep these settings consistent across builds.

Expected output

You have verified that the hardware or virtual platform can support UEFI, Secure Boot, and TPM 2.0, and you have a recovery path if the configuration blocks boot.

Validation

In firmware setup, confirm that the boot mode is UEFI and that TPM is available. In Windows, you should later be able to confirm the TPM using tpm.msc or PowerShell.

Common failure

The most common failure is changing Secure Boot before moving the system to UEFI-compatible boot media or disk layout. Another common failure is assuming a ?TPM present? label means the TPM is enabled and ready for Windows use.

Check the current state first

Goal

Record the baseline so you know what changed and can detect drift later.

Action

From Windows, collect the current firmware and TPM state.

Confirm-SecureBootUEFI returns whether Secure Boot is active, but only when the system is actually booted in UEFI mode. Get-Tpm shows whether Windows can see the TPM and whether it is ready.

You can also open tpm.msc to check the management console state if you prefer a GUI confirmation.

Expected output

You should know whether the server is already in UEFI mode, whether Secure Boot is on, and whether TPM is present and ready.

Validation

Expected signs of a correct baseline include:

- Confirm-SecureBootUEFI returns True when Secure Boot is enabled.
- Get-Tpm shows TpmPresent : True and TpmReady : True when the TPM is usable.
- Firmware menus show UEFI boot mode rather than Legacy or CSM.

Common failure

If Confirm-SecureBootUEFI throws an error, the machine may still be in legacy boot mode. If Get-Tpm reports that no TPM is present, the firmware may have TPM disabled, hidden, or unset to firmware-only mode.

Enable TPM in firmware

Goal

Make the TPM visible and usable to Windows.

Action

Reboot into firmware setup and look for the TPM option. Depending on the platform, it may be labeled as TPM, fTPM, PTT, security chip, or trusted platform module.

Set it to the enabled or activated state required by the vendor. On some systems you may need to:

- Enable the TPM device.
- Select TPM 2.0 rather than a compatibility or firmware-only mode.
- Save changes and reboot.

If the firmware offers both a discrete TPM and firmware TPM, choose the option that matches your hardware policy. Do not switch between them casually on a production server unless you understand the impact on measured boot, disk protection, and platform attestation.

Expected output

After reboot, Windows should be able to see the TPM and report it as present.

Validation

Run:

You want TpmPresent : True. If the TPM is present but not ready, review the firmware setting and then check whether the operating system needs ownership or initialization.

Common failure

A frequent failure is leaving the TPM in a hidden or disabled state in firmware. Another is assuming a firmware toggle alone is enough when the platform also requires a physical presence action or an explicit save-and-reboot cycle.

Switch the machine to UEFI boot

Goal

Ensure the operating system boots in the mode that Secure Boot requires.

Action

In firmware setup:

- Disable Legacy BIOS or Compatibility Support Module if the platform allows it.
- Set boot mode to UEFI.
- Confirm the Windows boot entry is the primary boot option.
- Save and restart.

If Windows was installed in legacy mode, you may need a conversion step before Secure Boot can be enabled. Do not force Secure Boot on a legacy-installed system and expect it to work; the boot path must be UEFI-compatible first.

Expected output

The system boots normally in UEFI mode and reaches Windows without falling back to legacy boot.

Validation

After reboot, check:

If the command runs successfully, the system is booted in UEFI mode. If it returns True, Secure Boot is already enabled; if it returns False, the machine is in UEFI but Secure Boot is still off.

Common failure

The most common failure is a bootable disk whose partition layout or boot loader is still tied to legacy BIOS. Another is losing the boot entry order after disabling CSM, which can prevent the machine from finding Windows.

Enable Secure Boot in firmware

Goal

Turn on firmware verification so the machine only boots approved boot components.

Action

In the UEFI setup utility, locate the Secure Boot setting and enable it. On some platforms, you may also need to:

- Select standard or default Secure Boot keys.
- Restore factory keys if the firmware keys were cleared.
- Confirm the OS type is set to a Windows/UEFI-compatible profile.

Save the firmware settings and reboot.

If you are managing a larger server estate, make sure the Secure Boot state aligns with your other security controls. In environments where operator access is part of the threat model, this typically belongs with broader access restrictions such as the approach described in Windows Server 2025: Harden RDP with Just Enough Administration.

Expected output

The server boots successfully with Secure Boot active.

Validation

Run:

A result of True confirms Secure Boot is enabled and active in the current boot session.

Common failure

Secure Boot often fails to enable when the firmware key store is empty, the system is still in legacy mode, or the installed boot chain is not signed in a way the firmware accepts.

Verify TPM readiness from Windows

Goal

Confirm that Windows can use the TPM, not just detect its existence.

Action

Check TPM status from Windows:

Review whether the TPM is present, enabled, activated, and ready. If required by your environment, open tpm.msc to confirm management state and available actions.

If the TPM exists but is not ready, investigate whether it needs initialization or ownership through the local security policy and vendor firmware rules. Avoid making unnecessary changes to a production TPM without understanding the platform state, especially if encryption or attestation is already in use.

Expected output

Windows reports the TPM as present and ready for use.

Validation

Useful indicators include:

- TpmPresent : True
- TpmReady : True
- No errors opening TPM management tools

Common failure

A TPM may be present but not ready after a firmware reset, motherboard replacement, or virtualization change. Another common issue is a VM template that exposes vTPM only after a separate configuration step in the hypervisor.

Validate the finished state

Goal

Confirm the server is in the secure boot-ready state you intended, not just partially configured.

Action

Collect the final verification data:

Use this as your post-change acceptance check:

- BIOS firmware type should indicate UEFI.
- Secure Boot should be active.
- TPM should be present and ready.

If the server will later support features that depend on firmware trust, this verification becomes your evidence that the platform is capable of supporting them. For high-availability systems, capture the result before the server joins a cluster or enters production. That keeps firmware remediation from becoming a disruptive maintenance task later, especially when paired with a broader availability design such as Configure Windows Server 2025 Failover Clustering for High Availability.

Expected output

You have a repeatable evidence set showing UEFI boot, active Secure Boot, and an available TPM.

Validation

The simplest validation matrix is:

- BiosFirmwareType shows UEFI.
- Confirm-SecureBootUEFI returns True.
- Get-Tpm returns TpmPresent : True and TpmReady : True.

Common failure

The most dangerous failure is a configuration that appears correct in firmware but was not verified from within Windows after reboot. Always validate the running OS state, not just the menu state.

Operational follow-up

Goal

Keep the secure boot and TPM state stable after deployment.

Action

Document the configuration in your build record and monitor for changes after firmware updates, motherboard swaps, virtualization changes, or recovery operations. After any maintenance that touches firmware, re-run the validation commands and confirm that no boot mode drift occurred.

If you use BitLocker, measured boot, attestation, or endpoint security tooling that relies on TPM-backed trust, verify those features after the first reboot and again after the next planned restart. Firmware settings can survive normal updates, but do not assume they will survive every lifecycle event unchanged.

Expected output

The server remains in the intended UEFI + Secure Boot + TPM state over time.

Validation

Re-run these checks after maintenance:

If either result changes unexpectedly, investigate firmware defaults, boot order, and hardware replacement procedures before returning the server to service.

Common failure

Common drift sources include BIOS updates that reset defaults, cloning workflows that alter UEFI identity, and VM reconfiguration that removes the virtual TPM or changes the boot generation.

Final check

A correctly configured Windows Server 2025 system for Secure Boot and TPM should boot in UEFI mode, report Secure Boot as enabled from within Windows, and show a TPM that is present and ready. If any of those three conditions is missing, the server is not yet in the finished state this tutorial targets.

Use the validation commands as your acceptance test every time you build, update, or recover the server. That is the practical difference between a firmware menu that looks right and a system that is actually protected.

Use this guidance together with FirewallD configuration and disable unnecessary services in Windows 11 to connect the workflow with related operational context already available on the site.