ARTICLE

Windows Server 2022 Hardening: Disable SMBv1 and Secure File Sharing

SMBv1 is still found on legacy Windows Server estates, but leaving it enabled creates unnecessary exposure on file servers and adjacent systems. This article explains when to disable SMBv1 on Windows Server 2022, how to validate dependencies before removal, what secure file-sharing controls to keep in place, and how to verify the change before production use.

Operating Systems - July 10, 2026

Why SMBv1 still matters on Windows Server 2022

The practical problem is not whether SMBv1 is old; it is whether it is still enabled anywhere it does not need to be. On Windows Server 2022, SMBv1 increases attack surface, complicates hardening, and can leave file services exposed to legacy protocol behavior that modern environments no longer require. If you operate domain file servers, application servers that mount shared storage, or jump hosts that still touch older systems, SMBv1 can remain hidden in plain sight until a scanner, incident, or failed connection exposes it.

After reading this article, you should be able to decide whether SMBv1 is actually required in your environment, understand the operational impact of disabling it, and validate that file sharing remains functional on safer SMB versions before production rollout. The goal is not to remove a protocol blindly; it is to remove one only when the dependency check and verification evidence are clear.

Key takeaways

- SMBv1 should be considered a legacy compatibility feature, not a normal production setting.
- The safest default on Windows Server 2022 is to keep SMBv1 disabled unless a verified dependency proves otherwise.
- File sharing is not just about disabling a protocol; authentication, share permissions, NTFS permissions, and audit visibility still matter.
- A production-safe change includes dependency discovery, staged removal, and post-change validation.

How SMBv1 affects secure file sharing

SMB is the protocol family Windows uses for file and printer sharing. SMBv1 is the oldest member of that family and lacks the modern protocol improvements that organizations rely on for stronger security and better behavior under current workloads. In practical terms, leaving it enabled increases the number of ways a client or service can negotiate older behavior that you do not want to support.

The important operational point is that secure file sharing is not achieved by protocol choice alone. A hardened file server also depends on correct share ACLs, NTFS permissions, account hygiene, network segmentation, and logging. If you are hardening a broader Windows Server 2022 estate, this control fits naturally alongside How to Harden Windows Server 2022 with Baseline Security Settings and Hardening Windows Server 2022 with Security Baselines and Audit Policies, because protocol reduction and auditability should be validated together.

When disabling SMBv1 is the right decision

Disabling SMBv1 is appropriate when your inventory shows no remaining dependency on legacy clients, embedded devices, or applications that require it. That includes most current Windows estates, most supported Linux SMB clients, and nearly all modern storage and application integrations. It is also the right choice when the server provides normal enterprise file shares and there is no documented requirement to support very old systems.

The decision becomes more nuanced when you still have one of these cases:

- An old NAS or appliance that only negotiates SMBv1.
- A packaged application with a hard-coded dependency on legacy SMB behavior.
- A scanning result that shows SMBv1 enabled but no actual evidence of client usage.
- A transitional environment where replacement of the legacy system is already scheduled.

In those cases, the operational question is not whether SMBv1 is undesirable; it is whether the business dependency is real, time-bound, and isolated enough to justify temporary exception handling.

Practical workflow for deciding and validating

A safe change sequence is simple in concept: confirm dependency, disable SMBv1 where unnecessary, and validate that access works over newer SMB versions. The exact tooling varies by environment, but the decision flow should remain consistent.

This workflow matters because SMBv1 is often enabled out of inertia rather than requirement. A scanner can tell you the protocol is available, but it cannot tell you whether any production system actually needs it. Conversely, a single failed connection from a legacy device may be enough to justify a replacement plan instead of a permanent exception.

How it works on Windows Server 2022

On Windows Server 2022, SMBv1 is treated as a removable feature rather than something that needs to remain part of the normal file-sharing stack. In a hardened configuration, modern SMB versions handle client access, while SMBv1 is removed or left disabled so the server only supports current protocol behavior.

That distinction is operationally useful because it lets you separate compatibility from security. If a host cannot connect after SMBv1 is disabled, the issue is usually not file sharing in general. It is often a legacy client, an outdated application component, or an external device that needs remediation. This makes the protocol change a useful discovery control as well as a hardening measure.

A practical environment where this matters

Consider a file server in a mixed estate where most endpoints are Windows 10/11, a few Linux application servers mount SMB shares, and one older scanner or label printer still points to a share on the server. The file server has been in place for years, the share works, and nobody has actively touched the SMB settings.

A vulnerability assessment flags SMBv1 as enabled. The immediate reaction might be to disable it everywhere. But an operational review shows that the label printer?s firmware is the only known legacy dependency, and even that dependency is undocumented. In this situation, the right answer is not to leave SMBv1 on indefinitely; it is to validate whether the device can be upgraded, isolated, or replaced. If it cannot, the exception should be explicitly time-bound and tightly scoped rather than treated as a permanent server default.

That is the kind of environment where the control pays off twice: you reduce exposure and you uncover undocumented dependencies before they become incidents.

What this means in practice

In practice, disabling SMBv1 should be treated as a change-control event with evidence, not a checkbox. The server should remain able to serve files over modern SMB versions, and the team should know which clients tested successfully after the change. You are looking for proof that the business service still works, not just proof that a registry or feature state changed.

For production use, the minimum evidence set should usually include:

- Confirmation that SMBv1 is not required by any current production client or application.
- Validation that normal file access works from representative Windows and non-Windows clients.
- Confirmation that share and NTFS permissions still enforce the intended access model.
- Review of logs or monitoring for failed legacy connection attempts after the change.

If your environment uses security baselines, this change should also be reflected in your hardening evidence so that the server configuration is consistent with the broader policy set and audit expectations.

Implementation trade-offs to weigh

The main trade-off is compatibility versus exposure. Keeping SMBv1 enabled reduces short-term friction with old devices, but it preserves a protocol surface you probably do not want. Disabling it improves security posture, but can reveal hidden dependencies that must be fixed or replaced.

There are also secondary trade-offs:

- Operational support load: disabling SMBv1 may generate help desk noise if a legacy client fails unexpectedly.
- Migration effort: some dependencies can be remediated quickly, while others require procurement, firmware upgrades, or application change windows.
- Change risk: if you disable the protocol without inventory evidence, you may create service disruption that is avoidable.
- Visibility gain: a failed SMBv1 dependency is often a useful signal that your environment still contains unmanaged legacy technology.

The most secure posture is usually the least compatible one that still meets a documented business requirement.

Verification checks before and after the change

Validation should focus on both protocol state and service behavior. A useful operational check is to confirm that SMBv1 is disabled on the server and that clients can still access shares using newer protocol versions. On Windows, the feature state can be reviewed with standard management commands, while file-share access can be checked from representative client systems using normal access paths.

A compact validation set might look like this:

These checks help confirm the server-side state, but they are not enough on their own. You still need a functional test from a client that uses an approved share path and account. If possible, validate both read and write access, and confirm that disconnected legacy devices do not silently retry with old settings.

After the change, monitor for failed connection attempts, authentication errors, or unexpected fallback behavior. If your logging and audit policy are already in place, this is where they become valuable, because they help you prove that the environment is behaving as intended rather than simply assuming it is.

Common mistakes that create avoidable risk

The most common mistake is disabling SMBv1 without checking whether something still depends on it. This is especially risky in environments with printers, scanners, appliances, OT-adjacent devices, or old application servers that were never formally inventoried.

Another frequent error is treating protocol removal as the full hardening story. A file server can still be weak if share permissions are overly broad, local admin rights are excessive, or audit visibility is poor. That is why SMB hardening should be viewed as one control among several rather than a standalone fix.

A third mistake is assuming that one successful test from one admin workstation proves compatibility for all clients. It does not. You want to test representative endpoints, especially any system class that is known to behave differently, such as older Linux builds, backup systems, or embedded devices.

Finally, teams sometimes disable SMBv1 but fail to document the dependency check. That makes the change hard to defend during an audit or during incident review, and it creates uncertainty if a related outage occurs later.

Decision guidance for production environments

If you are deciding whether to disable SMBv1 on a Windows Server 2022 host, the safest rule is straightforward: disable it when no current business dependency is verified, and keep the exception narrow and temporary when a dependency is real.

Use this decision logic:

- Disable now if the server provides standard file services and all known clients support modern SMB versions.
- Stage and validate if one or more clients are unknown, undocumented, or only partially tested.
- Keep temporarily with a remediation plan if a legacy device or application absolutely requires SMBv1 and there is no immediate replacement.
- Escalate for exception review if the dependency is broad, poorly understood, or tied to a critical service with no migration path.

That approach keeps the environment secure without forcing an unrealistic all-or-nothing decision.

Compact production readiness checklist

Before considering the change ready for production, verify the following:

- SMBv1 dependency is checked and documented.
- Representative clients can access the share after validation.
- Share permissions and NTFS permissions still match the intended access model.
- Monitoring or logs can show failed legacy attempts if they occur.
- Any exception has an owner, end date, and remediation plan.
- The hardening state is aligned with the server?s broader security baseline and audit policy.

If any item is missing, the configuration may still be acceptable in a lab or pilot, but it is not yet ready to be treated as a stable production hardening state.

Final takeaway

Disabling SMBv1 on Windows Server 2022 is one of the most practical hardening decisions you can make, but only when it is backed by dependency checks and validation evidence. The secure default is to remove legacy protocol exposure, keep file sharing on modern SMB versions, and verify that permissions, logging, and client access still behave as expected. If a legacy dependency exists, treat it as an exception to be managed and retired, not as proof that the old protocol should remain part of the standard build.

Use this guidance together with Windows 10 hardened security baseline and SELinux policy controls to connect the workflow with related operational context already available on the site.