core6

[ PRACTITIONER GUIDE ]

# Hardening Storage & Backup in the Age of AI-Accelerated Attacks

## A Guide for Storage & Infrastructure Teams

# Contents

# 1 Why the Timeline Just Collapsed

For most of the history of enterprise storage, the security clock ran slowly. A vulnerability was disclosed, a vendor eventually shipped an advisory, and storage teams folded the patch into the next maintenance window. The gap between disclosure and a working exploit was usually wide enough that a disciplined quarterly review kept you ahead of the threat. That assumption is no longer safe.

AI-assisted vulnerability research has compressed that timeline dramatically. Capabilities demonstrated by frontier models — Anthropic's Claude Mythos Preview being one of the first public examples — have shown that AI can not only surface deeply embedded flaws in foundational protocols but autonomously construct working exploits for them. The consequence for defenders is structural, not incremental: more vulnerabilities, disclosed faster, in deeper layers of the stack than the ones most storage teams have historically watched.

**The shift in one sentence**

When the time from disclosure to working exploit collapses, patch cycles stop being a sufficient defense on their own — and the systems that hold your last-resort recovery data are exactly the ones you cannot afford to leave exposed while you wait for a fix.

## Where the new vulnerabilities actually live

The first wave of AI-associated CVEs relevant to storage and backup shares a revealing pattern: almost none of them originate in vendor product code. They sit in the embedded components that storage and backup products are built on — TLS libraries, kernel subsystems, cryptographic providers, and BSD-derived protocol stacks. A flaw in an NFS/RPC implementation, an OpenSSL read path, or a certificate-validation routine can propagate into dozens of appliances from different vendors, many of which will not know they are affected until they audit their own software bill of materials.

This is why “we'll patch when the vendor tells us to” is now a slow and incomplete strategy. The exposure exists before the advisory is published, and it lives in dependencies your vendor may not immediately map to their own products.

## The perimeter moved to the control plane

At the same time, AI is improving attackers' ability to achieve initial access. As the classic network perimeter dissolves, the effective perimeter becomes the IT control plane — the management interfaces, administrative accounts, and protocol services that govern your infrastructure. Storage and backup systems sit squarely on this new perimeter, and they are high-value targets for a simple reason: they hold everything. A compromised storage array is the equivalent of breaching hundreds of database servers at once, and compromised backups remove the safety net you rely on to recover from ransomware.

## The two-phase defense model

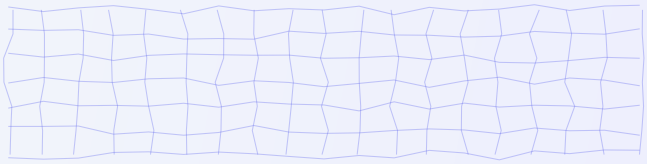
Everything in this guide is organized around a defense model with two distinct phases, because the right action depends entirely on which phase you are in:

- **Before a patch exists —** Prevention relies on continuous hardening and configuration discipline. If you have restricted management access, disabled unused protocols, enforced correct authentication, and segmented your networks, an exploit that depends on those exposures has nowhere to land. Hardening is the control you own outright; it does not wait on a vendor.
- **After an advisory lands —** The game becomes speed. How quickly can you determine which systems are exposed, understand the compensating controls, prioritize by risk, and remediate under change control? The faster you can answer those questions across your whole estate, the smaller the exploitation window.

Sections 2 through 4 build out the first phase — knowing your estate and hardening it. Section 5 builds out the second — responding when an advisory drops. Section 6 addresses the reality that makes both hard: point-in-time hardening decays, and only continuous validation keeps a hardened baseline hardened.

> **Background reading**
>
> This guide draws on two Core6 analyses of AI-accelerated disclosure and its impact on storage and backup. For the CVE-level detail behind the framing here, see “One of the First Public Mythos Showcases — and Its Impact on Storage” and “Anthropic-Associated CVEs Worth Watching for Storage and Backup” on the Core6 blog.

# 2 Know What You Have: Building Your Inventory



You cannot harden what you cannot see, and you cannot assess exposure to a new advisory if you do not know which systems run the affected component. Before any of the hardening in Section 3 is useful, you need a current, queryable inventory of your storage and backup estate — not a spreadsheet someone built two reorganizations ago, but a living picture of what is deployed, how it is exposed, and what it is built on.

When an AI-surfaced CVE in a protocol stack or crypto library is published, the teams that respond fastest are the ones who can already answer "which of our systems touch this?" in minutes rather than weeks. The inventory is what makes that possible.

## What to capture

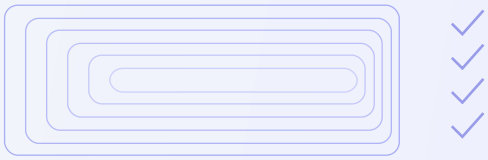
For every storage and backup system, record enough to answer both hardening and exposure questions:

- **Protocol exposure —** Which systems expose NFS, SMB/CIFS, iSCSI, NDMP, S3/object, or FTP, and on which network interfaces. NFS exposure in particular is the recurring theme in AI-surfaced CVEs.
- **Management-plane exposure —** Which systems expose management interfaces (web UIs, APIs, CLIs, SSH), and whether those interfaces are reachable from general user networks or only from a management network.
- **Underlying platform lineage —** Which systems are FreeBSD- or BSD-derived, which are Linux-based, and which run proprietary operating systems. Several early AI-associated CVEs are FreeBSD- or Linux-kernel-specific, so lineage directly predicts exposure.
- **Embedded components —** Where you can determine it, which crypto and TLS libraries (OpenSSL, wolfSSL, Bouncy Castle) and NFS implementations (in-kernel nfsd vs. user-space such as NFS-Ganesha) each system uses. This is exactly what a software bill of materials should tell you.
- **Data criticality —** Which systems hold regulated or mission-critical data, and which hold recovery data (snapshots, replicas, backups). This drives remediation priority when you cannot fix everything at once.

- **Software and firmware versions —** The storage OE, firmware, and backup-software versions in production, so you can match advisories to systems without a manual audit each time.

> **Inventory is a snapshot, and snapshots decay**
>
> An inventory is accurate the day you build it and drifting the day after. Upgrades, hotfixes, new deployments, and configuration changes all move systems out of the state you recorded. This is the first place configuration drift enters the picture — and it is why Section 6 argues that inventory and hardening both have to become continuous rather than periodic activities.

# 3 The Core Hardening Checklist



This is the heart of the guide. It is organized around the same five domains as the Core6 Enterprise Storage Security Self-Assessment — Access Control, Audit, Encryption, Cyber Resilience, and Threat Detection — so that you can read this section, then benchmark your own estate against the assessment and see your gaps scored in the same categories.

Each control is framed three ways: what to check, what good looks like, and how to remediate. Work through them domain by domain, or jump to the domain where you already suspect you are weakest. None of this depends on a specific vendor; Section 7 adds vendor-specific notes on top.

> **A note on ransomware:** resilience against ransomware is not a separate checklist — it is the payoff of getting Cyber Resilience (3.4) right. Immutability, isolation, secure snapshots, and secure time synchronization are the controls that determine whether you can recover when prevention fails. They are covered in depth in 3.4.



## 3.1 Access Control

Access control is where the control-plane perimeter is won or lost. If an attacker cannot reach your management interfaces and cannot authenticate against them, most storage and backup exploits never get started.

| Control | What good looks like | How to remediate |
|---|---|---|
| **Network isolation & management access** | Control-plane (management) networks are isolated from data-access traffic on separate VLANs. Management interfaces are reachable only through a dedicated management network or a jump host — never directly from general workstation networks. | Move management interfaces onto a dedicated management VLAN; place a hardened jump host as the sole entry point; block management ports on data interfaces at the switch and host firewall. |
| **Authentication & session controls** | Strong authentication is enforced on all management interfaces: password complexity (12+ characters, dictionary checks, expiry), idle-session timeouts, maximum concurrent-session limits, and restricted token lifetimes. | Set password policy and lockout thresholds in each system's auth config; enable idle timeout and concurrent-session caps; shorten API token lifetimes; integrate with central identity (LDAP/AD) and enable MFA where supported. |
| **Inactive account handling** | An automated process disables or removes accounts inactive for 90 days or more, so dormant credentials cannot be revived by an attacker. | Script or schedule an inactive-account review against the identity source; disable first, then remove after a grace period; alert on re-enablement of dormant accounts. |
| **Least-privilege administrative roles** | Role-based access control is in force, with administrators granted the minimum privileges their function requires. Full-control 'super admin' accounts are rare, named, and monitored. | Map roles to job functions; replace shared admin logins with named RBAC roles; separate storage-admin from security-officer duties so no single role can both weaken a control and delete the evidence. |
| **Unused protocols & services disabled** | Every unused storage and backup protocol, service, and host port is disabled or removed — unused FC/Ethernet ports, iSCSI, NFS, NDMP, FTP, S3/SWIFT, SNMP, NetBIOS, and similar. | Audit enabled services per system against what is actually used; disable the rest; close unnecessary ports at the host and network firewall; recheck after every upgrade, which often re-enables defaults. |

## 3.2 Audit

Audit is both a security control and a compliance obligation. Without tamper-resistant, centralized logging you cannot reconstruct an incident, and you cannot prove to a regulator that a control was in place at a point in time.

| Control | What good looks like | How to remediate |
|---|---|---|
| **Centralized, tamper-resistant logging** | Logs from all storage and backup systems are forwarded to a centralized, tamper-resistant destination (SIEM or secured log store), so an attacker who compromises a system cannot erase the record of it. | Configure syslog/API log forwarding on every system to your SIEM; restrict local log deletion; alert if a system stops sending logs — silence is often the first sign of compromise. |
| **Coverage of admin & config-change events** | Logging captures administrative actions and configuration-change events — not just data-access events — so a change that weakens a hardened control is recorded and attributable. | Enable configuration-change and admin-action auditing in each system; verify the key hardening settings emit an event when altered; forward those events with priority. |

| Control | What good looks like | How to remediate |
|---|---|---|
| **Audit-ready compliance evidence** | You can generate audit-ready evidence of storage and backup compliance (NIST, ISO, PCI DSS, DORA, NIS2) without manual, error-prone data gathering each cycle. | Map your controls to the frameworks you report against; automate evidence collection so a report can be produced on demand rather than reconstructed under deadline; retain historical posture snapshots as point-in-time proof. |

## 3.3 Encryption

Encryption protects data when other controls fail — but only if it is enabled consistently across at-rest and in-transit paths, and only if key management is disciplined. Gaps here are common precisely because encryption is often assumed rather than verified.

| Control | What good looks like | How to remediate |
|---|---|---|
| **Data-at-rest encryption** | At-rest encryption is enabled at the volume, LUN, or storage-pool level for all data classified as sensitive or regulated — not just on a subset of arrays. | Inventory which volumes/pools hold sensitive data; enable native at-rest encryption where it is off; confirm self-encrypting-drive or software encryption is actually active, not merely licensed. |
| **Data-in-transit encryption** | Data is encrypted in transit between hosts and storage (NVMe/TCP TLS, iSCSI CHAP + IPSec, SMB encryption, NFS Kerberos + TLS, DDBOOST TLS) and for replication and backup sessions. | Enable transport encryption on each protocol path; prioritize replication and backup links that cross network boundaries; see Section 4 for the TLS and certificate specifics AI-surfaced CVEs make urgent. |
| **Key management hygiene** | Encryption keys are managed centrally, rotated, and protected — ideally via a dedicated key manager — rather than left in default or locally stored states. | Integrate systems with a central key management service; enable key rotation; remove default keys; restrict who can export or escrow keys. |

## 3.4 Cyber Resilience

This domain determines whether you can recover when prevention fails. It is also where ransomware resilience lives: the entire point of ransomware is to make your data unrecoverable, and these controls are what keep a clean, restorable copy beyond the attacker's reach. If you read only one part of this checklist, read this one.

The AI-assistant examples in Core6's own product work surface exactly these failure modes in real estates — misconfigured retention-lock settings, backups not isolated from primary Active Directory, and missing secure time synchronization. They are common, and they are the difference between an incident and a catastrophe.

| Control | What good looks like | How to remediate |
|---|---|---|
| **Immutable copies — local** | Local immutable snapshots or WORM copies are configured for critical data, with a retention period that prevents deletion by storage admins — including during a ransomware attack where admin credentials may be compromised. | Enable immutability/retention-lock on critical snapshot policies; set retention that cannot be shortened by a standard admin; separate the role that can alter retention from day-to-day storage admins. |
| **Immutable copies — remote** | Immutable backup or replication copies are maintained at a remote or air-gapped location, isolated from the primary environment's authentication domain, so a domain compromise cannot reach them. | Establish a remote/air-gapped immutable copy; ensure it authenticates against a separate domain; verify isolation by confirming primary-domain credentials cannot mount or delete it. |
| **Backup isolation** | Backup platform and target systems are isolated at the network, domain, DNS, and AD/LDAP levels, and different user accounts are used to access primary storage versus backup systems. | Separate backup infrastructure onto isolated network segments; give it an independent identity domain; ensure no shared admin account spans primary and backup; block lateral paths between them. |
| **Secure snapshots** | Snapshots are protected from deletion by untrusted hosts or non-security-officer administrators, and snapshot scheduling is monitored for unexpected changes. | Restrict snapshot-delete rights to a security-officer role; alert on schedule or retention changes; treat an unexplained snapshot-policy change as a potential early attack indicator. |
| **Secure time synchronization** | NTP on all storage systems synchronizes with a trusted, authenticated time source, ensuring accurate timestamps for logs and snapshots — which underpins both forensics and retention integrity. | Point all systems at authenticated internal NTP; disable unauthenticated external time sources; verify drift is monitored, since bad time undermines log correlation and retention enforcement. |
| **Versioning** | File or object versioning is enabled so previous versions of data can be recovered after accidental deletion or ransomware encryption. | Turn on versioning where the platform supports it; set sensible version retention; confirm versions are themselves protected from mass deletion. |

## 3.5 Threat Detection

Hardening reduces the chance of compromise; detection reduces how long a compromise goes unnoticed. In a world of faster exploitation, the window between intrusion and impact is shrinking, so detection at the storage layer — where the attacker ultimately wants to be — matters more than ever.

| Control | What good looks like | How to remediate |
|---|---|---|
| **Anomaly & activity monitoring** | The storage environment integrates with anomaly detection that alerts on unusual access patterns (mass reads/writes, high entropy, off-hours access) and on mass file renames, deletions, or permission changes; NAS and file storage are scanned for malware. | Enable native anomaly/behavioral detection or integrate a monitoring solution; turn on file-activity alerting for mass-change events; enable native AV or connect a scanning solution for file/NAS storage. |
| **Baseline drift alerting** | You are alerted when a previously-hardened system drifts from its secure baseline — because an upgrade reverted a setting, or someone changed a control — rather than discovering it at the next manual review. | Define a secure baseline per platform; monitor live configuration against it; alert on deviation. This is the control that turns Section 6's 'periodic decays' problem into a real-time signal. |
| **Continuous assessment & remediation tracking** | Storage and backup security is assessed regularly, and remediation of identified gaps is tracked to closure rather than logged and forgotten. | Schedule recurring posture assessments; track findings with owners and due dates; measure mean-time-to-remediate so gaps do not linger long enough to be exploited. |

# 4 Protocol Hardening in Depth: NFS, RPC & TLS

The checklist in Section 3 covers the full breadth of hardening. This section goes deep on the specific protocol and transport layers where AI-surfaced CVEs have concentrated — because these are the areas where the collapsed timeline bites hardest, and where a few precise configuration choices meaningfully reduce exposure before any patch exists.

## NFS and RPC

NFS is ubiquitous across NAS systems, backup appliances, data-protection repositories, and archive gateways — and its long-standing RPC implementations are exactly the kind of deeply embedded, network-accessible code that AI-assisted research is now surfacing flaws in. Similar classes of vulnerability may exist across different NFS/RPC implementations regardless of whether a system runs FreeBSD, Linux, or a proprietary OS. Harden accordingly:

- **Restrict NFS access —** Limit NFS to trusted internal networks and avoid external exposure entirely. Network-accessible RPC services are the trigger surface for this class of flaw.
- **Segment storage networks —** Isolate storage, backup, and management traffic from each other and from general user networks, so a reachable service is not reachable from everywhere.
- **Harden export policies —** Enforce least privilege on exports; eliminate overly broad or wildcard exports that expand who can reach the service.
- **Prefer NFSv4.1 or later —** Where you can, move off NFSv4.0. Some AI-associated CVEs — CVE-2026-31402 in the Linux kernel nfsd, for example — specifically affect the NFSv4.0 code path; NFSv4.1 and later use a different session model that does not traverse the vulnerable logic.
- **Use Kerberos correctly, and only where needed —** If RPCSEC_GSS is required, prefer krb5i or krb5p — but audit whether Kerberos is actually needed in a given deployment before assuming it provides protection. In at least one showcased flaw, the vulnerable path involved RPCSEC_GSS itself.
- **Minimize attack surface —** Disable unused protocols and services; close unnecessary ports. Every service you turn off is one you never have to patch under pressure.

- **Watch for in-kernel vs. user-space NFS —** Exposure to some kernel-level NFS flaws depends on whether a product uses in-kernel nfsd or a user-space implementation such as NFS-Ganesha. Treat Linux-based systems serving NFS as potentially affected until the vendor confirms which they use.

> **A practical NFS/RPC starting point**
>
> If you do nothing else this week: confirm no NFS service is reachable from outside its trusted segment, remove every wildcard export, and identify which of your systems still run NFSv4.0. Those three actions address the bulk of the currently showcased NFS/RPC exposure without waiting on a single advisory.

## TLS and certificates

A second cluster of AI-surfaced CVEs sits in TLS libraries and certificate-validation code — OpenSSL, wolfSSL, Bouncy Castle. These are embedded in management planes, replication links, and backup transports across many vendors. Certificate-validation flaws are particularly dangerous because they can allow forged digital identities. Harden the transport and trust layers:

- **Replace broad public-CA trust —** Swap broad public certificate-authority trust for an internal CA and a small, explicit allowlist of the certificates your systems should accept. This shrinks what a forged or mis-validated certificate can impersonate.
- **Enable mutual TLS where supported —** mTLS forces both ends of a connection to prove identity, raising the bar against a flaw that only weakens one-directional validation.
- **Prefer modern cipher configurations —** Retire legacy modes where you can. Some flaws affect specific, older cipher paths (for example AES-CFB-128 as it appears in S/MIME and some IPsec deployments) that modern TLS rarely uses.
- **Know your crypto dependencies —** You cannot assess a wolfSSL, OpenSSL, or Bouncy Castle CVE without knowing which systems embed which library and version. This is the SBOM question — covered next, in Section 5.

# 5 When an Advisory Drops: A Response Runbook

Sections 2 through 4 are the ‘before a patch exists’ phase. This section is the ‘after an advisory lands’ phase — the repeatable workflow that turns a frightening disclosure into a controlled response. The organizations that limit their exposure window are not the ones who panic fastest; they are the ones who run the same disciplined sequence every time.

## The response sequence

1. Scope affected systems. Using the inventory from Section 2, identify which systems run the affected component, protocol, or platform lineage. This is fast if your inventory is current and painful if it is not — which is the entire argument for keeping it current.
2. Apply compensating controls immediately. Before a patch is tested and deployed, reduce exposure with the hardening you already know: restrict access to the affected service, tighten segmentation, enforce restrictive access policies, and disable the vulnerable path where feasible.
3. Prioritize by risk. Rank remediation by exposure and value — internet-reachable and high-value systems (those holding regulated or recovery data) first. You will rarely be able to fix everything at once; sequence it deliberately.
4. Patch under change control. Test firmware and OS updates, then deploy through your change-control process. For storage and backup, an untested patch can be as disruptive as the vulnerability.
5. Validate and record. Confirm the fix is in place, verify the compensating controls can be safely relaxed, and record the whole sequence as point-in-time compliance evidence.

## Make your vendors do the mapping: the SBOM play

Because these CVEs originate in embedded components rather than vendor product code, the hardest question is often simply "am I affected?" A software bill of materials answers it. Ask every storage and backup vendor for an SBOM — and if they cannot provide one, ask targeted questions that are far more likely to get a useful answer today than a broad request:

- Do your products include the wolfSSL / Bouncy Castle / OpenSSL versions affected by these CVEs?
- Do you use in-kernel NFS (nfsd) or a user-space implementation such as NFS-Ganesha?
- Are any of your products FreeBSD-derived and affected by the relevant SA-26 advisories?

A specific, component-level question puts the burden of proof where it belongs and gives you an answer you can act on before a formal advisory is published.

### Why this phase is getting harder

AI-assisted research means more CVEs, surfacing faster, in deeper layers of the stack. Vendor advisories specific to storage and backup have so far lagged the underlying component disclosures. That lag is precisely the window an attacker exploits — and precisely why the ability to scope and respond quickly, on your own initiative, is now a core operational capability rather than an occasional fire drill.

# 6 From Periodic to Continuous

There is a hard truth underneath everything in this guide: hardening is not a state you reach, it is a state you maintain. A system you hardened last quarter is not necessarily hardened today, and the reasons it drifts are mundane, constant, and largely invisible.

## Why point-in-time hardening decays

- **Upgrades and hotfixes revert settings —** Updates to storage OE, firmware, storage software, and backup software routinely reset hardened settings back to insecure defaults, often without anyone noticing until an audit — or an incident.
- **Routine changes accumulate —** Changes to port zoning, file shares, LUNs, access rights, backup policies, and administrative access each nudge the security posture, and they happen continuously as part of normal operations.
- **New vulnerabilities arrive faster —** In the AI era, new flaws are discovered and disclosed faster than any manual review cadence can track, so a quarterly check is structurally behind from the day it is completed.
- **New guidance is published constantly —** Standards and advisories from bodies such as CISA, NIST, and CIS are updated regularly, expanding what "hardened" even means over time.

Put together, these mean that the gap between periodic reviews is exactly where exposure lives. A vulnerability or misconfiguration introduced the day after a review has months to be exploited before the next one finds it — if it finds it at all.

## What continuous looks like in practice

Moving from periodic to continuous does not mean reviewing everything by hand more often — that does not scale, and it is the manual burden teams are already drowning in. It means automating the posture check itself: continuously validating live configuration against a secure baseline, alerting on drift the moment it happens, and mapping newly published advisories to affected systems automatically as they emerge. The two-phase model from Section 1 only works at speed if both phases are continuous — hardening that self-monitors, and exposure analysis that runs the instant an advisory lands.

**Where StorageGuard fits**

This is the problem Core6 built **StorageGuard** to solve. StorageGuard is a Security Posture Management solution purpose-built for enterprise storage and backup. It continuously hardens and validates storage and backup configurations against a library of checks aligned to vendor hardening guides and industry standards, alerts on drift from a secure baseline, and automatically identifies which systems are exposed as new advisories emerge — moving teams from periodic, manual checks to continuous, automated posture management across the same five domains this guide is built on.

It is agentless, deploys on a customer-provided Windows VM, and scans read-only over standard protocols — so the tooling that watches your posture does not itself expand your attack surface.

Whether you adopt a platform for this or build the discipline another way, the strategic point stands: in an era where exploits ship faster than storage teams can patch, continuous hardening — not periodic patching — is how you stay ahead.

# 7 Appendix

## 7.1 Quick-reference hardening checklist

A condensed version of Section 3, for walking your estate. Mark each control as In place / Partial / Gap and turn the gaps into tracked remediation items.

### Access Control

- ☐ Management networks isolated from data traffic; management access via dedicated network or jump host only
- ☐ Strong authentication: password complexity, idle timeout, concurrent-session and token-lifetime limits, MFA where supported
- ☐ Inactive accounts (90+ days) automatically disabled or removed
- ☐ Least-privilege RBAC; storage-admin and security-officer duties separated
- ☐ All unused protocols, services, and ports disabled or removed

### Audit

- ☐ All systems forward logs to a centralized, tamper-resistant destination
- ☐ Administrative and configuration-change events are logged and attributable
- ☐ Audit-ready compliance evidence (NIST, ISO, PCI DSS, DORA, NIS2) generated without manual effort

### Encryption

- ☐ At-rest encryption enabled at volume/LUN/pool level for all sensitive or regulated data
- ☐ In-transit encryption on host-to-storage paths and on replication and backup sessions
- ☐ Keys managed centrally, rotated, and protected; no default keys

## Cyber Resilience (including Ransomware Resilience)

- [ ] Local immutable/WORM copies with retention that admins cannot shorten mid-attack
- [ ] Remote or air-gapped immutable copies, isolated from the primary auth domain
- [ ] Backup isolated at network, domain, DNS, and AD/LDAP levels; separate accounts for primary vs. backup
- [ ] Snapshots protected from deletion by untrusted hosts / non-security-officer admins; scheduling monitored
- [ ] Authenticated NTP to a trusted source on all systems
- [ ] File/object versioning enabled and protected

## Threat Detection

- [ ] Anomaly and file-activity monitoring enabled; NAS/file storage malware-scanned
- [ ] Alerting on drift from the secure baseline
- [ ] Regular posture assessment with remediation tracked to closure

## 7.2 Per-vendor hardening notes

The controls in Section 3 are universal; how you implement them varies by platform. The notes below point to where each control lives across the major storage and backup vendors StorageGuard supports. Treat them as orientation, not a substitute for each vendor's current hardening guide — and re-verify after every upgrade, since updates frequently reset these settings.

| Vendor / platform | Where the key hardening controls live |
|---|---|
| DELL Technologies<br>PowerScale, PowerStore, PowerProtect, PowerMax | RBAC and admin-role separation; management-network isolation; at-rest encryption (SED / D@RE); PowerProtect immutability and retention-lock for ransomware resilience; audit-event forwarding to SIEM. |
| NetApp<br>ONTAP | Role-based administrators and multi-admin verification; SnapLock for WORM/ immutability; Snapshot copy protection; NFS export-policy and Kerberos hardening; volume encryption (NVE/NAE); EMS log forwarding. |
| Everpure<br>formerly Pure Storage — FlashArray | SafeMode immutable snapshots isolated from admin deletion; management-plane access controls; at-rest encryption on by default — verify; directory-service integration and session controls; audit log export. |
| Hitachi Vantara | Resource-group and role-based access; management interface isolation; at-rest encryption; audit-log forwarding; replication-link and snapshot protection. |
| COHESITY<br>including Veritas NetBackup | DataLock immutability and retention; multi-factor and RBAC on the management plane; isolation from primary AD domain; anomaly detection; certificate and TLS configuration on management/replication. For NetBackup: immutable storage (WORM) targets, role-based access, and isolated master/media server credentials. |
| Commvault | Immutable/air-gapped copy configuration; role-based security and privileged-access controls; separation of backup identity from primary domain; TLS on data and control paths; audit trail export. |
| VEEAM | Immutable / hardened repository configuration; separate backup credentials and isolated backup domain; MFA on the console; encryption of backup data in flight and at rest; event logging to SIEM. |
| HPE<br>Alletra, Primera, 3PAR, StoreOnce | Role-based access and management-interface isolation; at-rest encryption; StoreOnce Catalyst and immutability/retention for backup resilience; secure replication; audit-log forwarding; disable unused services after firmware updates. |
| IBM<br>FlashSystem, Spectrum / Storage Defender, Safeguarded Copy | Safeguarded Copy immutable point-in-time copies isolated from admin deletion; role-based access and multi-factor on management; at-rest encryption; audit logging; NFS/SMB export hardening. |

| Vendor / platform | Where the key hardening controls live |
| --- | --- |
| Lenovo INFINIDAT<br>InfiniBox, InfiniGuard, InfiniSafe | InfiniSafe immutable snapshots, logical air-gap, and fenced forensic environment; RBAC and management-plane isolation; at-rest encryption; secure replication; audit-event export. |
| VAST | Element-store immutable snapshots and indestructible snapshot policies; access-control lists on the control plane; management-network isolation; at-rest and in-transit encryption; NFS/SMB export hardening; audit logging. |
| nasuni | Immutable object-versioned copies in the cloud back-end; identity-federated access controls and MFA; encryption in transit and at rest; edge-appliance and management hardening; audit logging of admin actions. |
| ctera | WORM / immutable snapshots and ransomware-protection engine; role-based access and MFA; end-to-end encryption; management-plane isolation; audit trails; hardening of edge filers and portal. |
| NUTANIX<br>Unified Storage / Files / Objects | Native ransomware protection and WORM policies; RBAC and management-plane isolation; data-at-rest encryption; snapshot and replication protection; audit logging via Prism. |
| cisco<br>MDS / Nexus SAN & storage networking | Fabric zoning and VSAN segmentation; management-interface hardening and role-based access; disabling unused ports, services, and protocols; secure firmware practices; audit logging. |
| rubrik | Append-only, immutable filesystem for backups; role-based access, MFA, and isolation from the primary AD domain; anomaly/ransomware detection; TLS on management and replication; audit-log export. |
| BROADCOM<br>Brocade — SAN fabric | Fabric zoning discipline; management-interface hardening and role-based access; disabling unused ports and services; secure firmware update practices; audit logging. |

## 7.3 Benchmark your estate

The five domains in this guide are the same five scored by the **Enterprise Storage & Backup Security Self-Assessment**. Having read the guide, the fastest way to turn it into action is to run your environment through the assessment and get a weighted scorecard that surfaces your gaps and prioritizes the controls that matter most — in the same categories you have just worked through.

**Next step**

**Complete the Enterprise Storage & Backup Security Self-Assessment** to benchmark your posture across Access Control, Audit, Encryption, Cyber Resilience, and Threat Detection — and receive a prioritized view of where to harden first.

https://www.core6.com/the-enterprise-storage-security-self-assessment/



core6

[ SECURELY CONFIGURE STORAGE & BACKUP SYSTEMS ]