ERNW
providing security.

ERNW WHITE PAPER 77

UNIFIED SECURITY HARDENING WITH CROSS-PLATFORM NATIVE BINARIES

Version:	1.0
Date:	May 20, 2026
Classification:	Public

Julian Suleder

Digital unterschrieben von Julian Suleder
Datum: 2026.05.18 20:11:39 +02'00'

ERNW
providing security.

Table of Content

1 Handling

The present document is classified as *Public*. Any distribution or disclosure of this document REQUIRES the permission of the document owner as referred in Section *Document Status and Owner*.

1.1 Document Status and Owner

As the owner of this report, the document owner has exclusive authority to decide on the dissemination of this document and responsibility for the distribution of the applicable version in each case to the places.

The possible entries for the status of the document are *Initial Draft*, *Draft*, *Effective* and *Obsolete*.

Report Information	
Title:	ERNW White Paper 77 - Unified Security Hardening with Cross-Platform Native Binaries
Document Owner:	ERNW Enno Rey Netzwerke GmbH
Version:	1.0
Status:	Effective
Classification:	Public
Author(s):	Niklas Heringer

Table 2: Document Status and Owner

1.2 Classification Levels

Classification Level	Audience
Public:	Everyone
Internal:	All employees and business partners
Confidential:	Only employees
Secret:	Only selected employees

Table 3: Classification Levels

1.3 Document Version History

Version	Date	Details
1.0	May 20, 2026	Initial version after quality assurance.

Table 4: Document Version History

2 Abstract

System hardening is a fundamental activity for reducing attack surfaces and ensuring infrastructure resilience.
While current open-source frameworks such as Lynis[1] and OpenSCAP[2] provide valuable auditing functionality, they exhibit distinct structural limitations regarding remediation guidance, automation safety, and resource efficiency.
Lynis offers extensive auditing but lacks automated remediation, whereas OpenSCAP's automated processes can be resource-intensive and often lack reversibility.
Consequently, current solutions do not adequately address the requirement for transparent, easily understood, customizable, automatically reversible, and operator-accessible hardening across heterogeneous environments.

To address these deficits, this study introduces Hardener, a cross-platform framework implemented in Go that unifies auditing and remediation through a Documentation-as-Code architecture.

By embedding declarative YAML logic within human-readable Markdown, the tool establishes a Single Source of Truth that simultaneously generates executable automation as well as verifiable compliance documentation.
Key contributions include granular execution control to mitigate "all-or-nothing" risks, atomic state rollback for safety, zero-dependency portability across heterogeneous Unix-like environments as well as integrating contextual guidance directly into the execution flow, articulating security risks and configuration details to reduce the cognitive load on practitioners.

Comparative evaluation against Lynis and OpenSCAP confirms that Hardener significantly reduces integration overhead and cognitive load while ensuring safe, reproducible enforcement of hardening guidelines.

[1] *https://github.com/CISOfy/lynis*
[2] *https://www.open-scap.org/*

3 Introduction

3.1 Context & Background

System hardening, the process of configuring hosts to reduce attack surface and enforce security policy[3,4] sits at the intersection of configuration management, vulnerability assessment and automated remediation. [Bey25] Hardening operating systems (OS) and applications is crucial to protect them against the exploitation of both disclosed and yet unknown vulnerabilities. The primary goal of hardening is reducing the potential attack surface, both technical and operational, by ensuring systems are up-to-date, configuring them according to best practices (such as the least privilege access principle), and turning off nonessential services. [Lei23, pp. 8 sqq.] These measures, combined with essential cyber hygiene[5], are highly impactful, potentially helping to protect systems against 98% of attacks. [Mic23]

As suggested, hardening thereby involves multiple layers of protection, starting with selecting trusted hardware that supports necessary security features, and securing UEFI/BIOS settings, as system/application hardening effectiveness is diminished without these considerations. [Lei23, p. 11]

3.2 Hardening & Compliance Guidelines

Hardening is commonly performed according to established guidelines, checklists, and benchmarks developed by various agencies and communities, some of which are outlined below. [Lei23, p. 11]

[3] *Intel Corporation.* What is system hardening? *2025. URL: https://www.intel.com/content/www/us/en/learn/what-is-system-hardening.html (visited on Mar. 9, 2026).*

[4] *HYPR.* What is Systems Hardening? *2025. URL: https://www.hypr.com/security-encyclopedia/hardening (visited on Mar. 9, 2026).*

[5] *Strong passwords & password rotation, use of MFA, checking links pre-access, regularly producing backups & updating software, ... [BSI24]*

3.2.1 Industry- & Company-specific Standards

Compliance often requires mapping technical hardening to sector or company-specific standards.
Examples: PCI-DSS[6] for payment card environments, HIPAA[7] for protected health information, ISO/IEC 27001[8] for organization-level ISMS, NIST SP 800-53 for US federal systems[9], and IEC/ISA-62443 for industrial/OT[10].

These standards define mandatory controls, assessment/audit processes and evidence requirements that must be translated into concrete hardening checks and measurable remediation/rollback procedures. [Wan+24]

3.2.2 STIGs (Security Technical Implementation Guide)

A set of guidelines, called *Security Technical Implementation Guides* and *Security Requirements Guides*, often used by the U.S. Department of Defense (DoD) and government agencies, that assists in hardening a system. [Def25; Lei23, pp. 7, 12]

3.2.3 CIS Benchmarks (Center for Internet Security)

CIS Benchmarks offer recommendations for technical hardening against cyber-attacks across numerous platforms, including different Linux distributions, cloud providers, mobile devices, and server software. [Lei23, pp. 12, 13]

CIS Benchmarks can assist organizations with compliance requirements for standards like the Payment Card Industry Data Security Standard (PCI-DSS) and the Health Insurance Portability and Accountability Act (HIPAA).[Ben25, p. 2; Lei23, p. 12; Tev21, p. 573]

3.2.4 NIST (National Institute of Standards and Technology)

NIST regularly produces various security publications and guidelines, including the SCAP framework, the NIST Cybersecurity Framework (CSF), and Special Publications (SP) such as SP 800-53[11], which provides a catalog of security and privacy controls for federal information systems. [Tev21, p. 500; Lei23, p. 33] These publications are widely adopted

[6] Payment Card Industry Data Security Standard (PCI DSS). *PCI Security Standards Council. 2024. URL: https://www.pcisecuritystandards.org/standards (visited on Mar. 9, 2026).*

[7] HIPAA Security Rule / HHS Guidance. *U.S. Department of Health and Human Services. 2024. URL: https://www.hhs.gov/hipaa/for-professionals/security/index.html (visited on Mar. 9, 2026).*

[8] ISO/IEC 27001:2022 – Information security management. *International Organization for Standardization (ISO). 2022. URL: https://www.iso.org/standard/27001 (visited on Mar. 9, 2026).*

[9] *NIST.* NIST Special Publication 800-53, Rev. 5: Security and Privacy Controls. *Tech. rep. National Institute of Standards and Technology, 2020. URL: https://csrc.nist.gov/publications/detail/sp/800-53/rev-5/final (visited on Mar. 9, 2026).*

[10] ISA/IEC 62443 series – Security for industrial automation and control systems. *ISA / IEC. 2024. URL: https://www.isa.org/standards-and-publications/isa-standards/isa-iec-62443-series-of-standards (visited on Mar. 9, 2026).*

[11] *National Institute of Standards and Technology.* Security and Privacy Controls for Information Systems and Organizations (SP 800-53 Rev. 5). *2020. URL: https://csrc.nist.gov/pubs/sp/800/53/r5/upd1/final (visited on Mar. 14, 2026).*

beyond the federal government, serving as foundational references for organizations seeking to establish or assess their security posture.

3.3 Automation in Hardening

Automating hardening is necessary in large IT environments to reduce manual workload, ensure consistency, and prevent configuration drift. [Lei23, pp. 8 sqq., 2; NVI20]

Configuration management (CM) solutions like Ansible, Puppet, or PowerShell DSC are crucial for implementing secure configurations reliably, often treating the desired infrastructure state as Infrastructure as Code (IaC). [Has21; NIS11]

4 Analysis of Existing Tools

To understand the necessity for a new hardening framework, it is essential to analyze the capabilities and limitations of current state-of-the-art tools. Two prominent tools in the auditing scene, especially for Unix-like systems, are Lynis and OpenSCAP. While both are industry standards [Ahm+26], they represent distinct philosophies (manual expert-driven auditing versus standards-based automated compliance) and both exhibit significant structural limitations.

4.1 Lynis

4.1.1 Overview & Capabilities

Lynis is a free and open-source command-line interface (CLI) security auditing tool primarily designed for Linux, macOS, and Unix-based systems (including AIX, HP-UX, and Solaris). In the course of this paper we are talking of standalone Lynis, not Lynis Enterprise with additional paid features.

It performs detailed security scans, system configuration checks, and vulnerability assessments by inspecting the local system without requiring agents (agentless). [Ben25, p. 2; Tev21, p. 569; Web25]

Key features include:

- **Hardening index:** Provides a numerical score (0-100) reflecting the system's security posture. [Ben25, p. 3; Web25]
- **Indirectly actionable output:** Results include `[WARN]` messages for immediate issues and `[SUGGESTION]` messages for improvements. [Ben25, p. 3; Web25]

4.1.2 Limitation: Absence of Guided Remediation

Although Lynis results are rather detailed, they leave users facing uncertainty regarding remediation of its findings:

```
- Checking IPv6 configuration                    [ ENABLED ]
    Configuration method                         [ MANUAL ]
    IPv6 only                                    [ NO ]
- Checking configured nameservers
  - Testing nameservers
      Nameserver: 127.0.0.53                     [ OK ]
  - DNSSEC supported (systemd-resolved)          [ UNKNOWN ]
- Checking default gateway                       [ DONE ]
- Getting listening ports (TCP/UDP)              [ DONE ]
- Checking promiscuous interfaces                [ OK ]
- Checking waiting connections                   [ OK ]
- Checking status DHCP client                    [ NOT ACTIVE ]
- Checking for ARP monitoring software           [ NOT FOUND ]
- Uncommon network protocols                     [ 0 ]
```

Figure 1: Lynis results provide findings but lack direct remediation commands or rollback instructions should the remediation break necessary functionality.

Lynis findings are categorized by severity and status (e.g., `[ NONE ]`, `[ SUGGESTION ]`, `[ NOT FOUND ]`).
While this granular output is valuable for experienced administrators, it lacks direct remediation guidance.

- **No Automated Fixes**: Lynis does not offer remediation of *any* kind. Users must manually research and apply fixes for every finding. [Ded24; CISnd]
- **Lack of Context**: A result such as – `DNSSEC supported (systemd-resolved) [ UNKNOWN ]` informs the operator of a risk but does not explain the configuration file involved nor the steps required to resolve it properly.

4.2 OpenSCAP

4.2.1 Overview & Capabilities

OpenSCAP is an open-source suite based on the *NIST Security Content Automation Protocol* (SCAP) standard. [Tev21, pp. 18, 500]

Unlike Lynis, it is compliance-driven, supporting profiles like DISA STIG, PCI-DSS, and HIPAA, and features automated remediation capability (`oscap xccdf eval --remediate`). [Tev21, 499ff. Red25, 49ff.]

4.2.2 Limitation 1: Setup Complexity & Fragmentation

While OpenSCAP provides a mature framework, its initial configuration on non-Red Hat-based systems is often non-trivial. On Ubuntu 24.04, for instance, the standard `openscap-utils` package frequently lacks the associated compliance content (SCAP Security Guide or SSG), or the available content is out of sync with the installed OS version, requiring users to manually compile it from source.

Furthermore, the generated content files must match the target OS version exactly; a mismatch (e.g., scanning Ubuntu `24.04` with content built for `22.04`) results in *every* rule being reported as *not applicable*, rendering the tool effectively useless without deep tool-specific expertise.

4.2.3 Limitation 2: Opaque Logic

OpenSCAP content is encoded in deeply nested XML structures (XCCDF/OVAL, [Nat05]) that are difficult for humans to audit. A simple check, such as verifying if `rpcbind` is disabled, is wrapped in layers of namespaces and logic:

```
<ds:extended-component id="scap_org.open-scap_ecomp_ubuntu2404-checks-sce-service_rpcbind_
↪ disabled.sh">
  <sce:script>#!/bin/bash
if [[ $(systemctl is-enabled rpcbind.service) == "masked" ]] ; then
    exit "$XCCDF_RESULT_PASS"
fi
exit "$XCCDF_RESULT_FAIL"
```

```
</sce:script>
</ds:extended-component>
```

This lack of transparency makes it difficult for operators to "sanity check" what the tool is actually doing. Understanding the intent, effect, and potential risks of a rule requires navigating thousands of lines of XML rather than reading a simple configuration file.

4.2.4 Limitation 3: Dangerous Remediation (No Rollback)

Automated remediation with OpenSCAP is very linear: it applies changes based on the profile. Crucially, there is no automated rollback mechanism. [Red25, 49ff.]

- **Risk of System Non-Functionality**: If a remediation action (e.g., disabling a network protocol or changing permissions) breaks a critical service, the administrator must manually debug and reverse the change. [BSI24, p. 8]
- **Testing Overhead**: Because of this irreversibility, extensive preliminary testing in non-production environments is required before applying any changes. [Lei23, 25ff.]

OpenSCAP functions as a powerful compliance engine, rigorous in its enforcement of standards, but demanding significant expertise and caution to rely on its remediation capabilities.

4.3 Summary

In conclusion, while Lynis excels at transparent auditing and OpenSCAP at standardized compliance, neither offers a solution that is simultaneously cross-platform, transparently automated, and safely reversible. This creates the need for a framework like Hardener, which combines the readability of documentation with the safety of reversible code.

5 Methodology and Requirements Engineering

5.1 Problem Statement

Current state-of-the-art auditing frameworks necessitate a structural trade-off between inspection transparency (Lynis) and automated enforcement (OpenSCAP). Consequently, heterogeneous environments currently lack a unified solution that combines granular auditing with reversible, cross-platform remediation. This study addresses these disjointed operational capabilities by engineering a framework that bridges the gap between diagnostic reporting and active system hardening.

5.2 Gap Analysis

To derive objective design requirements, a Gap Analysis is employed to map the structural limitations of existing tools directly to necessary architectural capabilities. [Ita+18] This approach derives system requirements directly from the structural limitations of State-of-the-Art tools, specifically Lynis and OpenSCAP, as identified in Section 4. The following Traceability Matrix maps these architectural limitations to necessary design requirements, ensuring the solution bridges the identified gaps.

Gap ID	Identified Deficit	Operational Impact	Design Requirement
G1	**Lack of State Reversibility** (OpenSCAP)	Remediation is inherently destructive and not automatically reversible.	**Atomic Rollback:** System must support the exact restoration of pre-execution file states and permissions.
G2	**Logic Opacity** (Complex XML/Hidden Scripts)	Inhibits validation of automated logic ("Black Box" effect), reducing operator trust and auditability.	**Cognitive Transparency:** Policy logic must be strictly declarative and human-readable (*Doc-as-Code*).
G3	**Fragmented Lifecycle** (Audit-Only Tooling)	Disconnect between detection and correction necessitates manual intervention, preventing scalable enforcement.	**Integrated Remediation:** System must unify diagnostic auditing and active enforcement in a single workflow.
G4	**Environmental Coupling** (Dynamic Dependencies)	High fragility in heterogeneous environments due to interpreter or library version mismatches.	**Static Portability:** Zero-dependency architecture (static binary) executable on any standard Unix-like kernel.

5.3 Requirements for the New Tool

Based on the Gap Analysis, the following Functional (FR) and Non-Functional (NFR) requirements are defined. Each requirement is annotated with the specific Gap ID it addresses.

5.3.1 Functional Requirements (FR)

These define specific behaviors the system must support to overcome the identified gaps.

- **FR-01 (Policy Parsing)** [Ref: G2]: The system shall parse hardening policies defined strictly in Markdown files containing declarative YAML metadata, ensuring the documentation is the code.
- **FR-02 (Audit Mode)** [Ref: G3]: The system shall verify the current system state against the policy and report compliance (Pass/Fail) without modifying the system, mirroring the diagnostic capability of Lynis.
- **FR-03 (Remediation Mode)** [Ref: G1,G3]: The system shall provide an active enforcement mode to apply configuration changes (e.g., file edits, service commands) to align the system with the defined policy.
- **FR-04 (Safety Snapshots)** [Ref: G1]: Before applying any modification, the system must automatically capture the current state of the target configuration file or system parameter.
- **FR-05 (Rollback Capability)** [Ref: G1]: The system shall provide a dedicated command to restore system state using previously created snapshots, effectively reversing specific remediation actions.
- **FR-06 (Human-Centric Auditability)** [Ref: G2]: All policy logic (checks and fixes) must be readable and understandable by a human operator without specialized tooling, avoiding opaque formats like XCCDF/OVAL.
- **FR-07 (Idempotency)** [Ref: G1]: Repeated execution of remediation on an already compliant system must result in zero changes to the system state, preventing configuration drift and unnecessary write operations.

5.3.2 Non-Functional Requirements (NFR)

These define the quality attributes and constraints under which the system operates.

- **NFR-01 (Cross-Platform Portability)** [Ref: G4]: The artifact must be a single, statically linked binary. It must execute on heterogeneous Unix-like environments (Linux, macOS) without requiring the installation of interpreters (e.g., Python, Ruby) or system libraries.

6 System Architecture and Design

6.1 Modular Architecture and Portability

The system is engineered as a statically linked binary to satisfy **NFR-01 (Portability)**, ensuring consistent execution across Unix-like environments without external runtimes. Minimal dependencies, such as standard shell tools (e.g., `grep`), are explicitly validated via the `00_README.md` gatekeeper file.

The codebase is modularized to decouple the interface from the core compliance logic:

- **CLI (`/cmd`)**: Manages argument parsing and run orchestration.
- **Internal Engine (`/internal`)**: Handles policy ingestion, security level filtering, and shell execution.
- **Persistence and Safety (`/rollback`)**: Governs pre-remediation state snapshots and transactional rollbacks to satisfy **FR-04** and **FR-05**.

6.2 Documentation-as-Code Policy Engine

Hardener implements a *Doc-as-Code* architecture to satisfy *FR-01 (Policy Parsing)* and *FR-06 (Human-Centric Auditability)* [Wri24]: Unlike opaque XML schemas, policies are embedded within standard Markdown files, serving simultaneously as executable code and human-readable documentation.

The following structure illustrates how Hardener balances automated execution with operational safety:

```
checksuites:
- id: "01-ssh-full-stack-installed" # CIS Rule SV-270665r1067133
  description: "VerifiY ..." # Human-readable intent (FR-06)
  command: |
        dpkg -l | grep -E '^ii\s+openssh-(client|server|sftp-server)' | wc -l
  expected: 3                                # Value required for a 'Pass' status
  sudo: false                                # Elevation required for the audit command
  fix: "sudo apt update && sudo apt install ..."         # Automated command OR safety-gat
↪ e string
  fix_sudo: true                             # Elevation required for remediation
  affected_file: "/var/lib/dpkg/status"            # Path targeted by this check
  post_action: 'None'                  # Required follow-up to finalize changes
  security_level: 'baseline'                      # Policy filter tier (baseline vs. high)
  arch: [x86_64, aarch64]                            # Supported CPU architectures
  risk_level: "low"                          # Assessed impact on system stability
  risk_desc: "Ensures all necessary SSH components..."  # Contextual warning of possible s
↪ ide effects
```

Hardener mitigates remediation risks by constraining automation to defined security levels and implementing "manual safety gates". Fix attributes prefixed with `Manual action required` bypass automated execution, instead providing the operator with documented procedures directly via the CLI.

This architecture ensures high-risk modifications, such as bootloader security, remain under explicit human control, fulfilling *FR-01 (Policy Parsing)* by consolidating intent, method, and consequence of a single declarative file.

6.3 Execution Flow and Logic

The execution flow contains two distinct operational modes to address *FR-02 (Audit Mode)* and *FR-03 (Remediation Mode)*.

- **Target Selection:** The engine recursively scans the repository for `.md` files, ignoring hidden files (starting with a leading `.`) to prevent unintended policy application.
- **Security Level Filtering:** The system matches the `security_level` flag against the user-provided `security`-flag (*baseline* (default) or *high*) to adhere to user instructions concerning the desired level of hardening.
- **Privilege Escalation:** Hardener dynamically toggles `sudo` based on YAML attributes for both auditing and remediation.
- **Success Metrics:** Distinguishes between "Passed" (already compliant) and "Fixed" (successfully remediated) to provide granular transparency.
- **Diagnostic Audit:** The system executes shell commands via `sh -c`, capturing `stdout`. It compares the output against the expected value to determine compliance (Pass/Fail) without state modification (*FR-02*).
- **Conditional Remediation:** In fix mode, non-compliant checks trigger a remediation sequence. This sequence is strictly guarded: it executes only after a successful safety snapshot is confirmed, fulfilling *FR-04*.

6.4 Rollback Architecture (State Management)

To satisfy *FR-05 (Rollback Capability)* and ensure operational safety, Hardener utilizes a localized, delta-based state management system. This mechanism ensures atomic reversibility of any applied remediation.

6.4.1 Delta-Based Snapshots

Rather than archiving full file copies, which would be resource-intensive, Hardener utilizes a *diff-match-patch* algorithm to compute cryptographic deltas.

- **Pre-Remediation Backup:** Before a `fix` is applied, the `PreBackup` function captures the current file content and permissions (*FR-04*).

- **Delta Computation:** A cryptographic patch (a compact, text-based representation of only the changed characters between two states) is generated, representing the difference between the pre-remediation and post-remediation states. [Fra06]
- **Persistence:** Patches are serialized into a local `runs.json` database alongside SHA-256 checksums, ensuring data integrity without external database dependencies.

Hardener also generates report summaries for each run. Reports and backups are stored *relative* to the binary's position in the current directory, no persisting directory is created on the system.

6.4.2 Restoration Logic

Restoration logic reconstructs the original file state by applying the stored reverse-patch to the current file.

1. **Run Selection:** The user can target a specific execution via a timestamp or roll back the latest session by default.
2. **Patch Application:** The system merges the stored patch with the current system file to restore the previous text.
3. **Atomic Write-Back:** Hardener attempts an atomic rename (using `.tmp` files) to replace the configuration for safety-purposes, should errors arise during rollback.
4. **Privilege Elevation:** If standard write permissions are insufficient, the system falls back to `sudo tee` and `sudo chmod` to ensure the restoration of both content and original file permissions.

By storing the original file permissions (`origPerm`) alongside the content delta, Hardener prevents security regressions where a restored file might otherwise be left with overly permissive access rights (*FR-05*).

7 Evaluation and Discussion

7.1 Experimental Setup

To evaluate the hypotheses of this study, a standardized test set of *57 security controls* was defined, derived from the *CIS Ubuntu Linux Benchmark* standard with an additional few deemed noteworthy that were not to be found in the latest CIS benchmark. [Cen25]

While the utilized benchmarks are formally validated for `Ubuntu 24.04 LTS`, the evaluation was performed on a fresh installation of `Ubuntu 25.10`. This choice was made to test the frameworks' resilience on a newer, non-LTS release; the technical security controls remain applicable due to the architectural continuity of the underlying Linux subsystems (e.g., the Linux kernel, systemd, and OpenSSH).
Currently, no dedicated CIS Benchmarks for version 25.10 have been published.

The selected controls span four critical system domains: Kernel Parameters, SSH Configuration, File system Permissions, and Service Hardening. To ensure a fair comparison, a pre-hardening "blank" snapshot of the test virtual machine was created as a baseline.
This allows each tool to be evaluated under identical unhardened conditions by reverting to the baseline state after each test cycle.

Each framework (Lynis v3.1.4, OpenSCAP v1.4.2, and Hardener v1.0) was assessed against three operational metrics:

- **Detection Accuracy**: The ability of the tool to correctly identify the unhardened state of a specific control and provide adequate information regarding the vulnerability.
- **Remediation Efficacy**: The tool's capability to autonomously correct the configuration (where safely applicable) while providing a transparent risk assessment associated with the remediation logic.
- **Operational Safety**: The framework's ability to provide a "Rollback" or restoration mechanism to return the system to its previous state, mitigating the risk of accidental service disruption.

To ensure a balanced evaluation, it is critical to acknowledge the divergent operational scopes of the three frameworks: while Lynis serves primarily as a passive security auditor focused on diagnostic health checks, OpenSCAP is an enterprise compliance orchestrator designed to enforce rigid, industry-standard XML profiles across massive infrastructures. In contrast, *Hardener* is designed as a lightweight, developer-centric remediation tool that prioritizes transparency and reversibility over extensive profile databases. Although *Hardener* can emulate enterprise automation through binary execution in e.g., CI/CD pipelines, its primary goal is to bridge the "Logic Opacity" gap (*G2*) by providing human-readable, verifiable hardening that remains accessible to non-specialists.

7.2 Security Control Selection

The complete set of 57 security controls, including their technical command logic and remediation steps, is provided in *Appendix B* (available as a dataset in Appendix B), where the structure of checks can be viewed.

It should be noted that each control is cross-referenced with its primary source.
In instances where the CIS Benchmark did not provide a specific recommendation for a noteworthy vulnerability, for example the restriction of `PermitRootLogin` in SSH setup, the security control was marked as `MISSING IN CIS`.

To demonstrate the structure of the Hardener policy artifacts, a complete example of the 'SSH Hardening' module, comprising check suites as well as explanations, is also provided in Appendix B.

While the Hardener framework will naturally identify 100% of the 57 security controls (as these were manually integrated into the test), the evaluation of "Detection Accuracy" in this study transcends a binary pass/fail result. Instead, it focuses on *Human-Centric Auditability (FR-06)*: the quality, verifiability, and clarity of the information provided to the end-user.

7.3 Lynis Evaluation

Lynis (v3.1.4) was executed against the Ubuntu 25.10 test environment via `sudo lynis audit system`. While Lynis provides a high-level health score, its performance across the three core metrics reveals significant gaps in deep system hardening.

7.3.1 Metric 1: Detection Accuracy

Lynis achieved a total detection rate of ~54.4%. However, the qualitative analysis shows that detection does not always equate to actionable insight.

System Domain	Total Controls	Detected / Partial	Missed	Profile Coverage
SSH Configuration	13	3	10	23.1%
Kernel Parameters	20	17	3	85.0%
File system / Auth	12	6	6	50.0%
Services / Logging	12	5	7	41.6%
Total	57	31	26	~54.4%

All Lynis results can be found in Appendix C.

- **Logic Opacity (Gap G2):** In the file system domain (e.g., `35-fs-02-passwd-perms`), Lynis reports `[ OK ]` without displaying the expected baseline (e.g., octal `0644`). This forces the operator to trust an undocumented internal standard.

- **Shallow Scanning:** For the Bootloader Password (`37-fs-04-bootloader-password`), Lynis verifies file permissions but ignores the actual content (`password_pbkdf2` as requirement), creating a false positive for compliance.
- **Configuration Blindness:** While Lynis identifies the running SSH daemon, it exhibits "Detection Blindness" regarding internal cryptographic and session hardening. Of the 13 defined SSH controls, it failed to detect critical CIS requirements for strong MACs, KEX algorithms, and FIPS-compliant ciphers.

7.3.2 Metric 2: Remediation Efficacy

Lynis fundamentally lacks an automated remediation mechanism (*FR-03*). It operates strictly as a passive auditor.

- **Advisory Only:** Findings are paired with proprietary IDs (e.g., `AUTH-9286`) and links to external blog posts. This manual translation of suggestions into configuration commands is error-prone and slow.
- **Fragmented Lifecycle (Gap G3):** The disconnect between detection and correction places the entire operational burden on the user, failing the requirement for a unified hardening workflow.

7.3.3 Metric 3: Operational Safety

Because Lynis does not modify the system, it provides no safety features.

- **No Rollback:** Since no changes are made by the tool, no restoration mechanism is provided (*FR-05*).
- **Risk Blindness:** While it warns of "risks," it cannot simulate or safely test configuration changes, leaving the system's stability entirely to the administrator's manual expertise.

7.4 OpenSCAP Evaluation

OpenSCAP (v1.4.2) was evaluated to assess its remediation efficacy (FR-03). The evaluation utilized the official SCAP Security Guide (SSG) Data Stream (`ssg-ubuntu2404-ds.xml`) targeting the CIS Ubuntu Linux 24.04 LTS Benchmark.

7.4.1 Execution Constraints and Environmental Fragility

The initial evaluation on Ubuntu 25.10 demonstrated significant environmental fragility (Gap *G4*). Because the host OS version did not match the strictly compiled CPE (Common Platform Enumeration) dictionary, the OVAL engine rejected the host, returning a not applicable state for all 57 controls. This confirms a failure to meet *NFR-01 (Portability)* on non-LTS interim releases.

7.4.2 Metric 1: Detection Accuracy

Upon setting up another test environment to a supported Ubuntu 24.04 LTS baseline, functional execution was achieved. However, detection remains constrained by profile rigidness:

System Domain	Total Controls	Theoretically Covered	Missed	Profile Coverage
SSH Configuration	13	10	3	76.9%
Kernel Parameters	20	5	15	25.0%
File system / Auth	12	8	4	66.7%
Services / Logging	12	9	3	75.0%
Total	57	32	25	~56.1%

The 25 "Missed" controls highlight the framework's rigidity: Highly specific DISA STIG requirements (such as restricting `userns` clones or enforcing precise file permissions on `/etc/shadow`) are absent from the baseline CIS XML. Amending the profile to include these edge-case rules is hindered by *Logic Opacity (Gap G2)*, as it requires authoring complex OVAL schemas rather than utilizing standard administrative commands.

7.4.3 Metric 2: Remediation Efficacy

The remediation phase was executed on the supported Ubuntu 24.04 environment using the integrated `--remediate` directive. OpenSCAP demonstrated prolonged remediation times, completing the combined scanning and remediation cycle in approximately five minutes.

To evaluate the efficacy of the automated fixes, a post-remediation audit was performed to measure the delta between the pre- and post-hardened states of the 32 theoretically covered controls. The analysis accounts for both newly fixed controls and existing secure defaults that were affected by the run.

System Domain	Covered Controls	Pre-Fix (Pass or Fail/`notapplicable`)	Post-Fix (Pass or Fail/`notapplicable`)	Net Pass Variance
SSH Configuration	10	7 / 3	4 / 6	-3 (Regression)
Kernel Parameters	5	0 / 5	4 / 1	+4
File system / Auth	8	3 / 5	7 / 1	+4
Services / Logging	9	0 / 9	3 / 6	+3
Total	32	10 / 22	18 / 14	+8

Table: Delta analysis of compliance states before and after OpenSCAP remediation.

While OpenSCAP successfully resolved several deficiencies (e.g., in Kernel Parameters and File system), the post-remediation analysis revealed significant architectural instability during the enforcement phase:

- **State Regression and Idempotency Failure:** Six SSH controls (e.g., `05-use-pubkey-auth`, `09-disable-x11-forwarding`) that initially evaluated as `pass` due to Ubuntu's native defaults regressed to a `fail` state following

remediation. This demonstrates a severe violation of idempotency (*FR-07*): the OVAL remediation scripts destructively overwrote existing, valid configurations with conflicting formatting, actively introducing non-compliance into a previously secure subsystem.

- **Opaque Execution and Missing Fixes:** During the remediation process, several failable controls did not appear in the remediation output. This is a design consequence of the SCAP standard: if a control in the XML profile lacks an explicit `<fix>` element, the engine silently skips it without notifying the operator, resulting in a confusing and non-transparent user experience.
- **Dependency Failures in Lifecycle:** In the Services domain, the remediation successfully installed the `auditd` package (Check 48) and enabled the service (Check 49). However, subsequent dependent rules (e.g., Check 52) remained `notapplicable`, while others (e.g., Check 51) transitioned to `fail`. This indicates that the remediation logic failed to dynamically initialize or fully reload the newly installed subsystem required for downstream rule validation within the same execution cycle.

These findings demonstrate that while OpenSCAP provides automated enforcement, its complex OVAL remediation scripts lack context-awareness. The propensity for idempotency failures and silent omissions significantly degrades operator trust and necessitates heavy manual post-remediation auditing.

The entire audit after remediation can be viewed in Appendix E.

Metric 3: Operational Safety

The evaluation of OpenSCAP's operational safety reveals an architectural deficit regarding system restoration: OpenSCAP does not natively generate *Safety Snapshots (FR-04)* or file-state deltas prior to modification. Consequently, the framework lacks any internal *Rollback Capability (FR-05)*; once remediation scripts are executed, the system's state is permanently altered. This design places the entire burden of disaster recovery on external solutions, such as hypervisor-level snapshots or manual backups, significantly increasing the risk of unrecoverable service disruption during automated hardening cycles.

7.5 Hardener

The hardener framework was evaluated in audit, fix, and rollback modes to assess its performance against the established metrics. The execution adheres to *FR-06 (Human-Centric Auditability)* by design as the hardening suites, attached in Appendix B are in coherence with the requirement of being readable, understandable and easily adjustable Markdown files.

7.5.1 Initialization and Precondition Validation

The framework initializes by parsing the `00_README.md` gatekeeper file.

During the implementation of the initialization phase, it was determined that granular, per-check architecture filtering introduced redundant complexity without providing significant operational utility.

Consequently, the framework shifted toward suite-level "Gatekeeper" validation using the `00_README.md` structure to ensure environment compatibility before execution. Future refinements may include formalized per-suite prerequisites to resolve the ambiguity of whether missing system structures, such as specific configuration directories, should be actively provisioned by the hardening framework or delegated to primary configuration management tools.

7.5.2 Metric 1: Detection Accuracy

Upon startup, the binary, acting as a standalone, statically linked artifact requiring no external interpreters besides some standard shell tools present by default on most modern Linux distributions (defined in `00_README.md`), executes as follows:

```
./hardener audit --path . --security-level high
[> AUDIT] In this mode, your system will be audited with a series of tests concerning secu
↪ rity hardening.

↪
   [> PASSED]: Target matched: OS, Architecture, and
   Preconditions supported by 00_README.md | Test succeeded.

   [...]

? Select Hardening Suites to Execute (Use Space to toggle, Enter to confirm)
> [x]  [+++] Execute ALL Applicable Suites
  [ ]  [+] SSH Hardening (13 checks)
  [ ]  [+] Kernel Hardening (20 checks)
  [ ]  [+] Filesystem Hardening (12 checks)
  [ ]  [+] Service Hardening (12 checks)
```

The initialization phase validates the system against the *Policy Parsing (FR-01)* requirement. The framework utilizes a "Documentation-as-Code" structure where security policies are stored as readable Markdown files.

```
[> HEADER] === Running suite: SSH Hardening ===

  [> FAILED]: 01-ssh-full-stack-installed | Test failed,
  fix needed.
  Desired Output: 3
  Output: 1
  Used command: dpkg -l | grep -E '^ii\s+openssh-
  (client|server|sftp-server)' | wc -l
↪
  [...]

  [> PASSED]: 15-kernel-02-randomize-va-space | Test
  succeeded.
  [...]

Successfully saved audit report to /mnt/shared/reports

[> SUMMARY] Check pass rate: 22.81%
[> SUMMARY] Fix applied rate: 0.00%
```

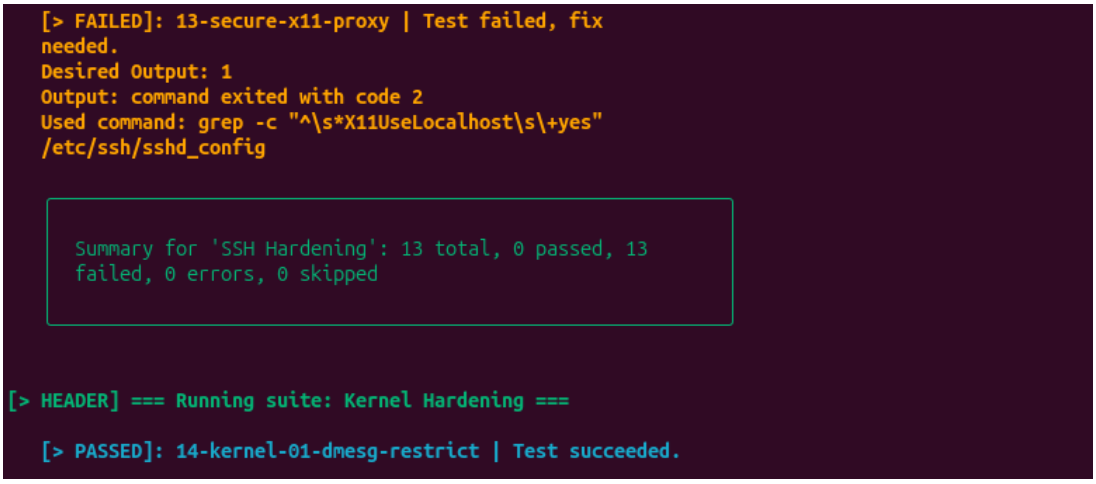

Figure 2: Intermediary feedback sections.

During the execution phase, the framework provides real-time feedback concerning the hardening domains adhering to *FR-06 (Human-Centric Auditability)*, as seen in Figure 2.

By executing in mode `audit`, the framework demonstrates its ability to verify system state against a declarative security policy without altering the host configuration, fulfilling the requirement *Audit Mode (FR-02)*.

7.5.3 Metric 2: Remediation Efficacy (FR-03)

To evaluate the framework's capacity for automated enforcement, the system was executed in `fix` mode. This phase translates the declarative security policies into active system modifications.

```
./hardener fix -p . -s high
[> FIX] In this mode, according to your audit results, automated fixes will be applied.

[...]

Successfully saved audit report to /mnt/shared/reports

[> SUMMARY] Check pass rate: 24.56%
[> SUMMARY] Fix applied rate: 67.44%
```

The execution yielded an automated remediation rate of 67.44%. The deviation from a complete 100% remediation rate is not indicative of execution failure, but rather the result of two distinct architectural constraints enforced by the framework.

First, a deliberate safety mechanism restricts the automation of high-risk configuration changes ("Manual Action Required", see explanation in Section 6).

Second, the remediation phase encountered operational blockers related to environmental idempotency.
As documented in the generated execution reports (Appendix F), certain fixes failed to apply because the targeted configuration files or directories (e.g., `/etc/audit/rules.d/`) did not exist on the unhardened baseline system. C Consequently, the framework's state-tracking mechanism aborted the remediation for these specific controls, as it cannot generate a valid pre-remediation backup (delta) of a non-existent file.

The framework achieves idempotency (*FR-07*) by executing a diagnostic audit immediately prior to any remediation; the enforcement logic *only triggers a fix if the initial RunCheck fails*. Verification confirms that repeated execution on a compliant system resulted in zero system modifications and zero write operations, ensuring configuration stability and preventing unnecessary state drift.

7.5.4 Lessons Learned and Future Architectural Considerations

The handling of non-existent configuration files presents a fundamental challenge in policy automation. While embedding file-creation logic (e.g., utilizing `mkdir -p` and `touch`) directly into the remediation commands ensures higher

automated compliance rates, it introduces ambiguity regarding the system's intended operational state prior to the intervention.

A prospective architectural refinement involves the implementation of *per-suite prerequisites*. Rather than delegating structural safety checks to individual remediation commands, the framework could define mandatory environmental preconditions at the suite level. This formalizes a critical operational boundary: determining whether a hardening framework should actively provision missing system structures or strictly operate upon pre-existing configurations, thereby leaving initial service provisioning to dedicated configuration management tools.

7.5.5 Metric 3: Operational Safety and Rollback (FR-04, FR-05)

To assess the framework's capability to safely revert system alterations, the rollback mechanism was invoked to reverse the atomic transaction created during the remediation phase.

```
./hardener rollback -t 2026-02-23T09:37:05Z
[> ROLLBACK] In this mode, your system will be rolled back to an older version...
[> INFO] Rolling back entire system to '2026-02-23T09:37:05Z'
[> INFO] Loaded 42 runs from /mnt/shared/runs.json
[...]

./hardener audit -p . -s high
[> SUMMARY] Check pass rate: 26.32%
[> SUMMARY] Fix applied rate: 0.00%
```

A subsequent diagnostic audit revealed a post-rollback pass rate of 26.32%.
While the framework successfully restored the targeted file contents and their explicit permissions, this metric represents a minor "compliance creep" relative to the pre-hardened baseline (22.81%). This variance empirically demonstrates the architectural distinction between *File-State Reversion* and *System-State Reversion*, driven by two specific environmental factors:

1. **Package Lifecycle Persistence:** During remediation, the framework actively installed missing dependencies (e.g., `openssh-server` for SSH hardening). The rollback mechanism restored the configuration files (e.g., `/etc/ssh/sshd_config`) but intentionally did not uninstall the binary packages. Consequently, default secure configurations native to the installed packages (such as `UsePAM yes`) remained active, inadvertently satisfying related compliance checks.
2. **Volatile Memory Retention:** Certain network hardening controls inject values directly into the live kernel memory (e.g., via `sysctl -w`). Reverting the persistent configuration file on disk and issuing a standard subsystem reload (`sysctl --system`) does not actively flush modified values from RAM unless the baseline configuration file explicitly defined the unsecure state. Therefore, the live kernel retains the secure parameter until a full system reboot occurs.

7.5.6 Lessons Learned and Future Architectural Considerations (Rollback)

These findings highlight a critical boundary in automated configuration management: achieving absolute state restoration strictly through file deltas is environmentally constrained. Reversing text strings and file permissions provides high integrity for static configurations but cannot account for volatile runtime states or secondary system artifacts like package installations. Future iterations of such frameworks must either incorporate deep package lifecycle tracking and mandate system reboots to flush kernel memory, or explicitly define their operational scope as complementary to hypervisor-level snapshotting mechanisms to guarantee absolute environmental parity.

8 Conclusion and Future Work

8.1 Conclusion

This short study validated that a "Documentation-as-Code" approach can successfully resolve the operational dichotomy between audit transparency and automated enforcement. The developed artifact, *Hardener*, eliminates the "black-box" nature of legacy compliance tools by embedding execution logic directly within human-readable Markdown. The evaluation confirmed that the framework satisfies all defined functional and non-functional requirements, delivering verifiable audits, safe automated remediation, and transactional state rollbacks without relying on external dependencies or complex XML schemas.

8.2 Limitations and Lessons Learned

Evaluation revealed that granular per-check environment validation was redundant; this logic was successfully consolidated into suite-level gatekeepers (`00_README.md`). Prototype constraints include silent failures during metadata ingestion, necessitating future schema enforcement, and the inherent inability of file-based rollbacks to revert package installations or volatile kernel memory.

8.3 Future Work

Future iterations will focus on:

- **Validation**: Implementing strict parsers and dedicated error indicators to eliminate silent ingestion failures.
- **Windows Support**: Extending the engine to address the unique architectural complexity of GPO and Registry integration, which exceeds standard command-line execution.

Hardener demonstrates that prioritizing transparency, reversibility, and portability provides a highly effective, low-complexity alternative to traditional compliance frameworks.

ERNW
providing security.

9 References

[Ahm+26] Naqvi Ahmed and Josephs. *Security Hardening Using FABRIC: Implementing a Unified Compliance Aggregator for Linux Servers*. 2026. URL: https://arxiv.org/html/2601.00909v1 (visited on Mar. 14, 2026).

[Ben25] Y. Benabderrezak. *Full Lab to Master Lynis*. 2025. URL: https://www.researchgate.net/publication/392194025_Full_Lab_to_Master_Lynis (visited on Mar. 9, 2026).

[Bey25] BeyondTrust. *What is systems hardening?* 2025. URL: https://www.beyondtrust.com/resources/glossary/systems-hardening (visited on Mar. 9, 2026).

[BSI24] BSI (Bundesamt für Sicherheit in der Informationstechnik). *Hardening Guideline*. 2024. URL: https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Cyber-Security/SiSyPHuS/AP11/Hardening_Guideline.pdf?__blob=publicationFile&v=3 (visited on Mar. 9, 2026).

[Cen25] Center for Internet Security (CIS). *CIS Ubuntu Linux Benchmark*. 2025. URL: https://www.cisecurity.org/benchmark/ubuntu_linux (visited on Mar. 9, 2026).

[CISnd] CISOfy. *Lynis Enterprise — FAQ*. n.d. URL: https://cisofy.com/lynis-enterprise/faq/ (visited on Mar. 9, 2026).

[Ded24] Dedoimedo. *Lynis — Robust security audit tool, but is it for Linux home users?* 2024. URL: https://www.dedoimedo.com/computers/lynis.html (visited on Mar. 9, 2026).

[Def25] Defense Information Systems Agency (DISA). *Security Technical Implementation Guides (STIGs)*. 2025. URL: https://www.cyber.mil/stigs (visited on Mar. 9, 2026).

[Fra06] Neil Fraser. *Diff Match Patch*. 2006. URL: https://github.com/google/diff-match-patch (visited on Mar. 14, 2026).

[Has21] HashiCorp. *What is infrastructure as code and why is it important?* 2021. URL: https://www.hashicorp.com/en/resources/what-is-infrastructure-as-code (visited on Mar. 9, 2026).

[24a] *HIPAA Security Rule / HHS Guidance*. U.S. Department of Health and Human Services. 2024. URL: https://www.hhs.gov/hipaa/for-professionals/security/index.html (visited on Mar. 9, 2026).

[HYP25] HYPR. *What is Systems Hardening?* 2025. URL: https://www.hypr.com/security-encyclopedia/hardening (visited on Mar. 9, 2026).

[Int25] Intel Corporation. *What is system hardening?* 2025. URL: https://www.intel.com/content/www/us/en/learn/what-is-system-hardening.html (visited on Mar. 9, 2026).

[24b] *ISA/IEC 62443 series – Security for industrial automation and control systems*. ISA / IEC. 2024. URL: https://www.isa.org/standards-and-publications/isa-standards/isa-iec-62443-series-of-standards (visited on Mar. 9, 2026).

[22] *ISO/IEC 27001:2022 – Information security management*. International Organization for Standardization (ISO). 2022. URL: https://www.iso.org/standard/27001 (visited on Mar. 9, 2026).

[Ita+18] Seyed M. R. Itani, Oliver Krancher, and Jens Dibbern. "Gap Analysis: A Review of Literature and Future Research Directions." In: *Proceedings of the International Conference on Information Systems (ICIS)*. 2018. URL: https://www.researchgate.net/publication/327879112_Gap_Analysis.

[Lei23] Tommi Leiritie. *Automated hardening of Linux infrastructure (Bachelor's thesis)*. 2023. URL: https://www.theseus.fi/bitstream/handle/10024/805686/Leiritie_Tommi.pdf?sequence=2 (visited on Mar. 9, 2026).

[Mic23] Microsoft Corporation. *Basic Cyber Hygiene Prevents 98% of Attacks*. 2023. URL: https://techcommunity.microsoft.com/blog/microsoft-security-blog/basic-cyber-hygiene-prevents-98-of-attacks/3926856 (visited on Mar. 9, 2026).

[Nat20] National Institute of Standards and Technology. *Security and Privacy Controls for Information Systems and Organizations (SP 800-53 Rev. 5)*. 2020. URL: https://csrc.nist.gov/pubs/sp/800/53/r5/upd1/final (visited on Mar. 14, 2026).

[Nat05] National Institute of Standards and Technology (NIST). *OVAL Tutorial 1 - Overview*. 2005. URL: https://csrc.nist.gov/CSRC/media/Projects/Security-Content-Automation-Protocol/documents/docs/conference%20presentations/workshops/oval%20tutorial%201%20-%20overview.pdf (visited on Mar. 9, 2026).

[NIS11] NIST. *Guide for Security-Focused Configuration Management of Information Systems (SP 800-128)*. Tech. rep. National Institute of Standards and Technology, 2011. URL: https://csrc.nist.gov/publications/detail/sp/800-128/final (visited on Mar. 9, 2026).

[NIS20] NIST. *NIST Special Publication 800-53, Rev. 5: Security and Privacy Controls*. Tech. rep. National Institute of Standards and Technology, 2020. URL: https://csrc.nist.gov/publications/detail/sp/800-53/rev-5/final (visited on Mar. 9, 2026).

[NVI20] NVISO. *Windows Server Hardening with PowerShell DSC*. 2020. URL: https://blog.nviso.eu/2020/03/03/windows-server-hardening-with-powershell-dsc/ (visited on Mar. 9, 2026).

[24c] *Payment Card Industry Data Security Standard (PCI DSS)*. PCI Security Standards Council. 2024. URL: https://www.pcisecuritystandards.org/standards (visited on Mar. 9, 2026).

[Red25] Red Hat. *Red Hat Enterprise Linux 8 Security Hardening*. 2025. URL: https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/8/pdf/security_hardening/Red_Hat_Enterprise_Linux-8-Security_hardening-en-US.pdf (visited on Mar. 9, 2026).

[Tev21] D. A. Tevault. *Mastering Linux Security and Hardening*. 2021. URL: https://www.packtpub.com/en-de/product/mastering-linux-security-and-hardening-9781837632626/ (visited on Mar. 9, 2026).

[Wan+24] W. Wang et al. "A Survey of Major Cybersecurity Compliance Frameworks." In: (2024). survey comparing SOC2, GDPR, PCI DSS, HIPAA, CIS, NIST, CMMC. URL: https://cake.fiu.edu/Publications/Wang%2Bal-24-MC.A_Survey_of_Major_Cybersecurity_Compliance_Frameworks.published-downloaded.pdf (visited on Mar. 9, 2026).

[Web25] WebAsha Technologies. *What is Lynis in Linux and How to Use It for Security Auditing and System Hardening*. 2025. URL: https://www.webasha.com/blog/what-is-lynis-in-linux-and-how-to-use-it-for-security-auditing-and-system-hardening (visited on Mar. 9, 2026).

[Wri24] Write the Docs. *Docs as Code*. 2024. URL: https://www.writethedocs.org/guide/docs-as-code/ (visited on Mar. 14, 2026).

A Appendix A: List of Abbreviations & Technical Words

Abbreviations and technical words in order of encounter.

Abbreviation	Definition
BIOS	Basic Input/Output System
CIS	Center for Internet Security, see Section 3
CLI	Command-Line Interface
CM	Configuration Management, see Section 3
DISA	Defense Information Systems Agency, supporting the DoD
DoD	Department of Defense (USA), see Section 3
HIPAA	Health Insurance Portability and Accountability Act, see Section 3
IaC	Infrastructure as Code
IEC	International Electrotechnical Commission
ISA	International Society of Automation
ISO	International Organization for Standardization
MFA	Multi-Factor Authentication
NIST	National Institute of Standards and Technology, see Section 3
OS	Operating System
OT	Operational Technology
OVAL	Open Vulnerability and Assessment Language
PCI-DSS	Payment Card Industry Data Security Standard, see Section 3
(PowerShell) DSC	(PowerShell) Desired State Configuration
SCAP	Security Content Automation Protocol
SSG	Scap Security Guide
STIG	Security Technical Implementation Guides and Security Requirements Guides, see Section 3
UEFI	Unified Extensible Firmware Interface
Unix	A family of operating systems
XCCDF	Extensible Configuration Checklist Description Format
XML	Extensible Markup Language
YAML	YAML Ain't Markup Language

B Appendix B: Used Security Controls

SSH.md

```
---
title: SSH Hardening
label: "80-ssh-hardening"
template:
    type: section
    id: 80
lang: "en-US"
checksuites:
  - id: "01-ssh-full-stack-installed" # CIS Rule SV-270665r1067133
    description: "Verify that the complete OpenSSH stack (client, server, and sftp-server)
↪  is installed."
    command: |
        dpkg -l | grep -E '^ii\s+openssh-(client|server|sftp-server)' | wc -l
    expected: 3
    sudo: false
    fix: "sudo apt update && sudo apt install -y openssh-client openssh-server openssh-sft
↪ p-server"
    fix_sudo: true
    affected_file: "/var/lib/dpkg/status"
    post_action: 'None'
    security_level: 'baseline'
    arch: [x86_64, aarch64]
    risk_level: "low"
    risk_desc: "Ensures all necessary SSH components are available. Installing the server
↪ enables remote access capability."

  - id: "02-disable-root-login" # MISSING in CIS
    description: "The root user MUST NOT be allowed to log in directly via SSH to enforce
↪ accountability."
    command: 'grep -c "^\s*PermitRootLogin\s\+no" /etc/ssh/sshd_config'
    expected: 1
    sudo: true
    fix: 'sed -i "s/^\s*#\?\s*PermitRootLogin.*/PermitRootLogin no/" /etc/ssh/sshd_config'
    fix_sudo: true
    affected_file: '/etc/ssh/sshd_config'
    post_action: 'File changed. Please restart SSH manually with sudo systemctl restart ss
↪ hd.'
    security_level: 'baseline'
    arch: [x86_64, aarch64]
    risk_level: "low"
```

```
    risk_desc: "Blocks direct root access. Use sudo for administrative tasks."

  - id: "03-disable-password-auth" # MISSING in CIS
    description: "Password authentication MUST be disabled in favor of SSH Key-based authe
↪ ntication."
    command: 'grep -c "^\s*PasswordAuthentication\s\+no" /etc/ssh/sshd_config'
    expected: 1
    sudo: true
    fix: 'sed -i "s/^\s*#\?\s*PasswordAuthentication.*/PasswordAuthentication no/" /etc/ss
↪ h/sshd_config'
    fix_sudo: true
    affected_file: '/etc/ssh/sshd_config'
    post_action: 'File changed. Restart SSH. WARNING: Ensure functioning SSH keys exist BE
↪ FORE applying.'
    security_level: 'high'
    arch: [x86_64, aarch64]
    risk_level: "high"
    risk_desc: "May lock out users without keys."

  - id: "04-max-auth-tries" # MISSING in CIS
    description: "Limit maximum authentication attempts to 3 to slow down brute-force atta
↪ cks."
    command: 'grep -E -i "^\s*MaxAuthTries\s+(3|2|1)" /etc/ssh/sshd_config | wc -l'
    expected: 1
    sudo: true
    fix: 'sed -i "s/^\s*#\?\s*MaxAuthTries.*/MaxAuthTries 3/" /etc/ssh/sshd_config'
    fix_sudo: true
    affected_file: '/etc/ssh/sshd_config'
    post_action: 'File changed. Please restart SSH manually.'
    security_level: 'baseline'
    arch: [x86_64, aarch64]
    risk_level: "low"
    risk_desc: "Low risk. May inconvenience users with poor typing."

  - id: "05-use-pubkey-auth" # CIS Rule SV-270722r1067130
    description: "Public key authentication MUST be enabled to support MFA/Key-based acces
↪ s."
    command: 'grep -c "^\s*PubkeyAuthentication\s\+yes" /etc/ssh/sshd_config'
    expected: 1
    sudo: true
    fix: 'sed -i "s/^\s*#\?\s*PubkeyAuthentication.*/PubkeyAuthentication yes/" /etc/ssh/s
↪ shd_config'
    fix_sudo: true
    affected_file: '/etc/ssh/sshd_config'
```

```
    post_action: 'File changed. Please restart SSH manually.'
    security_level: 'baseline'
    arch: [x86_64, aarch64]
    risk_level: "low"
    risk_desc: "Standard requirement for secure authentication."

  - id: "06-disable-empty-passwords" # CIS Rule SV-270717r1067177
    description: "Accounts with empty passwords MUST NOT be allowed to log in."
    command: 'grep -c "^\s*PermitEmptyPasswords\s\+no" /etc/ssh/sshd_config'
    expected: 1
    sudo: true
    fix: 'sed -i "s/^\s*#\?\s*PermitEmptyPasswords.*/PermitEmptyPasswords no/" /etc/ssh/ss
↪ hd_config'
    fix_sudo: true
    affected_file: '/etc/ssh/sshd_config'
    post_action: 'File changed. Please restart SSH manually.'
    security_level: 'baseline'
    arch: [x86_64, aarch64]
    risk_level: "low"
    risk_desc: "Critical security control with no negative impact on legitimate users."

  - id: "07-enable-pam" # CIS Rule SV-270741r1066712
    description: "UsePAM MUST be enabled to allow strong authentication modules (MFA, Limi
↪ ts)."
    command: 'grep -c "^\s*UsePAM\s\+yes" /etc/ssh/sshd_config'
    expected: 1
    sudo: true
    fix: 'sed -i "s/^\s*#\?\s*UsePAM.*/UsePAM yes/" /etc/ssh/sshd_config'
    fix_sudo: true
    affected_file: '/etc/ssh/sshd_config'
    post_action: 'File changed. Please restart SSH manually.'
    security_level: 'baseline'
    arch: [x86_64, aarch64]
    risk_level: "medium"
    risk_desc: "Ensure PAM is correctly configured on the system to avoid authentication l
↪ oops."

  - id: "08-idle-timeout" # CIS Rule SV-270743r1066718
    description: "Idle sessions MUST be terminated after 10 minutes (600s) to prevent hija
↪ cking."
    command: 'grep -c "^\s*ClientAliveInterval\s\+600" /etc/ssh/sshd_config'
    expected: 1
    sudo: true
    fix: 'sed -i "s/^\s*#\?\s*ClientAliveInterval.*/ClientAliveInterval 600/" /etc/ssh/ssh
```

```
↪ d_config && sed -i "s/^\s*#\?\s*ClientAliveCountMax.*/ClientAliveCountMax 1/" /etc/ssh/
↪ sshd_config'
    fix_sudo: true
    affected_file: '/etc/ssh/sshd_config'
    post_action: 'File changed. Please restart SSH manually.'
    security_level: 'baseline'
    arch: [x86_64, aarch64]
    risk_level: "low"
    risk_desc: "May disconnect inactive users, requiring them to reconnect."

  - id: "09-disable-x11-forwarding" # CIS Rule SV-270708r1066613
    description: "X11 Forwarding MUST be disabled to prevent GUI-based session hijacking."
    command: 'grep -c "^\s*X11Forwarding\s\+no" /etc/ssh/sshd_config'
    expected: 1
    sudo: true
    fix: 'sed -i "s/^\s*#\?\s*X11Forwarding.*/X11Forwarding no/" /etc/ssh/sshd_config'
    fix_sudo: true
    affected_file: '/etc/ssh/sshd_config'
    post_action: 'File changed. Please restart SSH manually.'
    security_level: 'baseline'
    arch: [x86_64, aarch64]
    risk_level: "low"
    risk_desc: "Breaks remote GUI applications forwarded over SSH."

  - id: "10-ssh-ciphers-fips-compliant" # CIS Rule SV-270667r1067107
    description: "Configure the SSH server to implement only FIPS-approved ciphers (AES-GC
↪ M and AES-CTR)."
    command: 'grep -E "^Ciphers\s+aes256-gcm@openssh.com,aes128-gcm@openssh.com,aes256-ctr
↪ ,aes128-ctr$" /etc/ssh/sshd_config | wc -l'
    expected: 1
    sudo: true
    fix: 'sed -i "/^Ciphers/d" /etc/ssh/sshd_config && echo "Ciphers aes256-gcm@openssh.co
↪ m,aes128-gcm@openssh.com,aes256-ctr,aes128-ctr" | sudo tee -a /etc/ssh/sshd_config'
    fix_sudo: true
    affected_file: '/etc/ssh/sshd_config'
    post_action: 'File changed. Please restart SSH manually with: sudo systemctl restart s
↪ shd.service'
    security_level: 'high'
    arch: [x86_64, aarch64]
    risk_level: "high"
    risk_desc: "Disables non-FIPS ciphers like ChaCha20. May impact performance on CPUs wi
↪ thout AES-NI hardware acceleration or connectivity for legacy clients."

  - id: "11-strong-macs" # CIS Rule SV-270668r1067110
```

```
    description: "Configure SSH to use only strong Message Authentication Codes (MACs)."
    command: 'grep -c "^\s*MACs\s\+hmac-sha2-512-etm@openssh.com" /etc/ssh/sshd_config'
    expected: 1
    sudo: true
    fix: 'sed -i "/^\s*MACs/d" /etc/ssh/sshd_config && echo "MACs hmac-sha2-512-etm@openss
↪ h.com,hmac-sha2-256-etm@openssh.com,hmac-sha2-512,hmac-sha2-256" | sudo tee -a /etc/ssh
↪ /sshd_config'
    fix_sudo: true
    affected_file: '/etc/ssh/sshd_config'
    post_action: 'File changed. Please restart SSH manually.'
    security_level: 'high'
    arch: [x86_64, aarch64]
    risk_level: "high"
    risk_desc: "Incompatible with legacy SSH clients."

  - id: "12-strong-kex" # CIS Rule SV-270669r1067112
    description: "Configure SSH to use only strong Key Exchange (KEX) algorithms."
    command: 'grep -c "^\s*KexAlgorithms\s\+ecdh-sha2-nistp521" /etc/ssh/sshd_config'
    expected: 1
    sudo: true
    fix: 'sed -i "/^\s*KexAlgorithms/d" /etc/ssh/sshd_config && echo "KexAlgorithms ecdh-s
↪ ha2-nistp521,ecdh-sha2-nistp384,ecdh-sha2-nistp256,diffie-hellman-group-exchange-sha256
↪ ,diffie-hellman-group16-sha512,diffie-hellman-group14-sha256" | sudo tee -a /etc/ssh/ss
↪ hd_config'
    fix_sudo: true
    affected_file: '/etc/ssh/sshd_config'
    post_action: 'File changed. Please restart SSH manually.'
    security_level: 'high'
    arch: [x86_64, aarch64]
    risk_level: "high"
    risk_desc: "Strict KEX limits may block older clients or specific automation tools."

  - id: "13-secure-x11-proxy" # CIS Rule SV-270709r1066616
    description: "Force X11 forwarding to bind to the localhost address to prevent remote
↪ exposure."
    command: 'grep -c "^\s*X11UseLocalhost\s\+yes" /etc/ssh/sshd_config'
    expected: 1
    sudo: true
    fix: 'sed -i "s/^\s*#\?\s*X11UseLocalhost.*/X11UseLocalhost yes/" /etc/ssh/sshd_config
↪ '
    fix_sudo: true
    affected_file: '/etc/ssh/sshd_config'
    post_action: 'File changed. Please restart SSH manually.'
    security_level: 'baseline'
```

```
    arch: [x86_64, aarch64]
    risk_level: "low"
    risk_desc: "Standard security practice with minimal impact."
---
```

Secure Shell (SSH) is the primary remote administration protocol and, as such, represents a critical attack surface. Hardening the SSH daemon (`sshd`) MUST be a priority to prevent unauthorized access, brute-force attacks, and cryptographic compromise.

Ensure Full Installation

To maintain system integrity and the full functionality of security protocols, the complete OpenSSH stack—including the client, server, and SFTP server components—must be present. The following packages MUST be installed on the system:

```
openssh-client openssh-server openssh-sftp-server
```

Disable Direct Root Login

Direct root login over SSH prevents accountability, as the identity of the human administrator is obscured. The administrator MUST use an unprivileged account and escalate privileges using `sudo`. The following directive MUST be set in `/etc/ssh/sshd_config`:

```
PermitRootLogin no
```

Disable Password Authentication

Password-based authentication is vulnerable to brute-force attacks and keylogging. Authentication MUST rely exclusively on **SSH Public Key Authentication**, where the client holds the private key (often protected by a strong passphrase). This setting requires that every legitimate user has a functioning SSH key pair *before* implementation.

```
PasswordAuthentication no
```

Limit Maximum Authentication Attempts

Limiting the number of authentication attempts per connection slows down automated brute-force attacks and reduces the efficiency of credential stuffing. The following directive MUST be configured to a value of 3 or less:

```
MaxAuthTries 3
```

Enable Public Key Authentication

To support modern Multi-Factor Authentication (MFA) and secure, key-based access patterns, public key authentication must be explicitly enabled. Ensure the following directive is active:

```
PubkeyAuthentication yes
```

Disable Accounts with Empty Passwords

Allowing remote access to accounts without passwords represents a critical security failure. The SSH daemon MUST be configured to reject any login attempt that does not provide a valid credential. Ensure the following directive is set:

```
PermitEmptyPasswords no
```

Enable PAM

Pluggable Authentication Modules (PAM) allow the SSH daemon to utilize system-wide security policies, including account locking, password complexity, and MFA modules. The following directive MUST be enabled:

```
UsePAM yes
```

Terminate Idle Sessions in Proper Time

Unattended, open SSH sessions are a target for session hijacking. The daemon should be configured to terminate connections that have been inactive for more than 10 minutes (600 seconds). The following directives MUST be configured:

```
ClientAliveInterval 600
ClientAliveCountMax 1
```

Disable X11 Forwarding

X11 Forwarding allows remote GUI applications to be displayed locally but can be exploited to hijack local X11 sessions. Unless strictly required for administrative GUI tools, it should be disabled. The following directive MUST be set:

```
X11Forwarding no
```

Use only FIPS-approved Ciphers

To ensure the confidentiality of data in transit, the SSH server must only utilize NIST FIPS-validated cryptographic ciphers (specifically AES-GCM and AES-CTR). The `Ciphers` list MUST be restricted to:

```
aes256-gcm@openssh.com,aes128-gcm@openssh.com,aes256-ctr,aes128-ctr
```

Use only Strong MACs

Message Authentication Codes (MACs) ensure the integrity of the data stream. Only strong, HMAC-SHA2 based algorithms should be permitted to prevent packet injection and tampering. The `MACs` list SHOULD include:

```
hmac-sha2-512-etm@openssh.com,hmac-sha2-256-etm@openssh.com,hmac-sha2-512,hmac-sha2-256
```

Use only Strong KEX Algorithms

Key Exchange (KEX) algorithms determine how the shared secret is established. Legacy algorithms are vulnerable to modern compute-based attacks; therefore, only Elliptic Curve or strong Diffie-Hellman exchanges should be used. The `KexAlgorithms` list SHOULD include:

```
ecdh-sha2-nistp521,ecdh-sha2-nistp384,ecdh-sha2-nistp256
```

Force X11 Forwarding

If X11 Forwarding is enabled, the server must force the X11 proxy to bind only to the loopback (localhost) address to prevent the X11 port from being exposed to the network. The following directive MUST be enabled:

```
X11UseLocalhost yes
```

`Kernel.md`

```
---
title: Kernel Hardening
label: "81-kernel-hardening"
template:
    type: section
    id: 81
lang: "en-US"
checksuites:
  # --- General Kernel Self-Protection ---
  - id: "14-kernel-01-dmesg-restrict" # CIS Rule SV-270749r1067179
    description: "Restrict access to the kernel message buffer (dmesg) to root only to pre
↪ vent information leakage."
    command: "sysctl -n kernel.dmesg_restrict"
    expected: 1
    sudo: false
    fix: |
      sudo mkdir -p /etc/sysctl.d/
      sudo touch /etc/sysctl.d/10-kernel-hardening.conf
      echo 'kernel.dmesg_restrict=1' | sudo tee -a /etc/sysctl.d/10-kernel-hardening.conf
    fix_sudo: true
    affected_file: '/etc/sysctl.d/10-kernel-hardening.conf'
    post_action: "Run 'sudo sysctl -p /etc/sysctl.d/10-kernel-hardening.conf' to apply cha
↪ nges."
    security_level: 'baseline'
    arch: [x86_64, aarch64]
    risk_level: "low"
    risk_desc: "Prevents non-root users from viewing kernel logs. May impact debugging too
↪ ls running as user."
```

```
  - id: "15-kernel-02-randomize-va-space" # CIS Rule SV-270772r1066805
    description: "Enable Address Space Layout Randomization (ASLR) to prevent buffer overf
↪ low exploits."
    command: "sysctl -n kernel.randomize_va_space"
    expected: 2
    sudo: false
    fix: |
      sudo mkdir -p /etc/sysctl.d/
      sudo touch /etc/sysctl.d/10-kernel-hardening.conf
      echo 'kernel.randomize_va_space=2' | sudo tee -a /etc/sysctl.d/10-kernel-hardening.c
↪ onf
    fix_sudo: true
    affected_file: '/etc/sysctl.d/10-kernel-hardening.conf'
    post_action: "Run 'sudo sysctl -p /etc/sysctl.d/10-kernel-hardening.conf' to apply cha
↪ nges."
    security_level: 'baseline'
    arch: [x86_64, aarch64]
    risk_level: "low"
    risk_desc: "Standard security feature on modern Linux."

  - id: "16-kernel-03-disable-core-dumps" # MISSING in CIS
    description: "Restrict core dumps (fs.suid_dumpable) to prevent sensitive memory from
↪ being written to disk during crashes."
    command: "sysctl -n fs.suid_dumpable"
    expected: 0
    sudo: false
    fix: |
      sudo mkdir -p /etc/sysctl.d/
      sudo touch /etc/sysctl.d/10-kernel-hardening.conf
      echo 'fs.suid_dumpable=0' | sudo tee -a /etc/sysctl.d/10-kernel-hardening.conf
    fix_sudo: true
    affected_file: '/etc/sysctl.d/10-kernel-hardening.conf'
    post_action: "Run 'sudo sysctl -p /etc/sysctl.d/10-kernel-hardening.conf' to apply cha
↪ nges."
    security_level: 'high'
    arch: [x86_64, aarch64]
    risk_level: "medium"
    risk_desc: "Inhibits debugging of crashing services. Developers may need this re-enabl
↪ ed temporarily."

  - id: "17-kernel-04-yama-ptrace-scope" # MISSING in CIS
    description: "Restrict ptrace to prevent unprivileged processes from inspecting siblin
↪ g processes."
```

```
    command: "sysctl -n kernel.yama.ptrace_scope"
    expected: 1
    sudo: false
    fix: |
      sudo mkdir -p /etc/sysctl.d/
      sudo touch /etc/sysctl.d/10-kernel-hardening.conf
      echo 'kernel.yama.ptrace_scope=1' | sudo tee /etc/sysctl.d/10-kernel-hardening.conf
    fix_sudo: true
    affected_file: '/etc/sysctl.d/10-kernel-hardening.conf'
    post_action: "Run 'sudo sysctl -p /etc/sysctl.d/10-kernel-hardening.conf'."
    security_level: 'baseline'
    arch: [x86_64, aarch64]
    risk_level: "medium"
    risk_desc: "May interfere with gdb/strace for non-root users."

  - id: "18-kernel-05-kptr-restrict" # MISSING in CIS
    description: "Hide kernel symbols/addresses from unprivileged users to mitigate KASLR
↪ bypass attacks."
    command: "sysctl -n kernel.kptr_restrict"
    expected: 2
    sudo: false
    fix: |
      sudo mkdir -p /etc/sysctl.d/
      sudo touch /etc/sysctl.d/10-kernel-hardening.conf
      echo 'kernel.kptr_restrict=2' | sudo tee -a /etc/sysctl.d/10-kernel-hardening.conf
    fix_sudo: true
    affected_file: '/etc/sysctl.d/10-kernel-hardening.conf'
    post_action: "Run 'sudo sysctl -p /etc/sysctl.d/10-kernel-hardening.conf'."
    security_level: 'high'
    arch: [x86_64, aarch64]
    risk_level: "low"
    risk_desc: "Essential for protecting against kernel exploits; minimal impact on normal
↪   operations."

  # --- eBPF & Sandboxing ---

  - id: "19-kernel-06-ebpf-unprivileged-disabled" # MISSING in CIS
    description: "Disable unprivileged eBPF to reduce the kernel attack surface."
    command: "sysctl -n kernel.unprivileged_bpf_disabled"
    expected: 1
    sudo: false
    fix: |
      sudo mkdir -p /etc/sysctl.d/
      sudo touch /etc/sysctl.d/10-kernel-hardening.conf
```

```
      echo 'kernel.unprivileged_bpf_disabled=1' | sudo tee -a /etc/sysctl.d/10-kernel-hard
↪ ening.conf
    fix_sudo: true
    affected_file: '/etc/sysctl.d/10-kernel-hardening.conf'
    post_action: "Run 'sudo sysctl -p /etc/sysctl.d/10-kernel-hardening.conf'."
    security_level: 'high'
    arch: [x86_64, aarch64]
    risk_level: "medium"
    risk_desc: "Breaks non-root performance tracing and specific network filters."

  - id: "20-kernel-07-ebpf-jit-hardening" # MISSING in CIS
    description: "Enable eBPF JIT hardening to mitigate JIT spraying and information leaks
↪ ."
    command: "sysctl -n net.core.bpf_jit_harden"
    expected: 2
    sudo: false
    fix: |
      sudo mkdir -p /etc/sysctl.d/
      sudo touch /etc/sysctl.d/10-kernel-hardening.conf
      echo 'net.core.bpf_jit_harden=2' | sudo tee -a /etc/sysctl.d/10-kernel-hardening.con
↪ f
    fix_sudo: true
    affected_file: '/etc/sysctl.d/10-kernel-hardening.conf'
    post_action: "Run 'sudo sysctl -p /etc/sysctl.d/10-kernel-hardening.conf'."
    security_level: 'baseline'
    arch: [x86_64, aarch64]
    risk_level: "low"
    risk_desc: "Minimal performance overhead for JIT compilation."

  - id: "21-kernel-08-userns-clone-disabled" # MISSING in CIS
    description: "Disable unprivileged user namespace creation to limit attack surface."
    command: "sysctl -n kernel.unprivileged_userns_clone"
    expected: 0
    sudo: false
    fix: |
      sudo mkdir -p /etc/sysctl.d/
      sudo touch /etc/sysctl.d/10-kernel-hardening.conf
      echo 'kernel.unprivileged_userns_clone=0' | sudo tee -a /etc/sysctl.d/10-kernel-hard
↪ ening.conf
    fix_sudo: true
    affected_file: '/etc/sysctl.d/10-kernel-hardening.conf'
    post_action: "Run 'sudo sysctl -p /etc/sysctl.d/10-kernel-hardening.conf'."
    security_level: 'high'
    arch: [x86_64, aarch64]
```

```
    risk_level: "high"
    risk_desc: "CRITICAL: Breaks rootless containers (Podman), Flatpaks, and browser sandb
↪ oxing."

  # --- Protocol & Filesystem Hardening ---

  - id: "22-kernel-09-protected-fifos" # MISSING in CIS
    description: "Enable strict FIFO protection to prevent race conditions in named pipes.
↪ "
    command: "sysctl -n fs.protected_fifos"
    expected: 2
    sudo: false
    fix: |
      sudo mkdir -p /etc/sysctl.d/
      sudo touch /etc/sysctl.d/10-kernel-hardening.conf
      echo 'fs.protected_fifos=2' | sudo tee -a /etc/sysctl.d/10-kernel-hardening.conf
    fix_sudo: true
    affected_file: '/etc/sysctl.d/10-kernel-hardening.conf'
    post_action: "Run 'sudo sysctl -p /etc/sysctl.d/10-kernel-hardening.conf'."
    security_level: 'high'
    arch: [x86_64, aarch64]
    risk_level: "low"
    risk_desc: "Prevents a common class of local privilege escalation."

  - id: "23-mod-01-blacklist-dccp" # MISSING in CIS
    description: "Blacklist the DCCP protocol module to reduce the network attack surface.
↪ "
    command: |
      grep -r 'blacklist dccp' /etc/modprobe.d/ | wc -l
    expected: 1
    sudo: false
    fix: |
      sudo mkdir -p /etc/modprobe.d/
      sudo touch /etc/modprobe.d/blacklist-dccp.conf
      echo 'blacklist dccp' | sudo tee /etc/modprobe.d/blacklist-dccp.conf
    fix_sudo: true
    affected_file: '/etc/modprobe.d/blacklist-dccp.conf'
    post_action: "Run 'sudo modprobe -r dccp' and update initramfs."
    security_level: 'baseline'
    arch: [x86_64, aarch64]
    risk_level: "low"
    risk_desc: "Only impacts applications using the Datagram Congestion Control Protocol."

  - id: "24-mod-02-blacklist-sctp" # MISSING in CIS
```

```
    description: "Blacklist the SCTP protocol module to reduce the network attack surface.
↪ "
    command: |
      grep -r 'blacklist sctp' /etc/modprobe.d/ | wc -l
    expected: 1
    sudo: false
    fix: |
      sudo mkdir -p /etc/modprobe.d/
      sudo touch /etc/modprobe.d/blacklist-sctp.conf
      echo 'blacklist sctp' | sudo tee /etc/modprobe.d/blacklist-sctp.conf
    fix_sudo: true
    affected_file: '/etc/modprobe.d/blacklist-sctp.conf'
    post_action: "Run 'sudo modprobe -r sctp' and update initramfs."
    security_level: 'baseline'
    arch: [x86_64, aarch64]
    risk_level: "low"
    risk_desc: "Disables Stream Control Transmission Protocol."

  # --- Network Stack Hardening ---

  - id: "25-net-01-syn-cookies" # CIS Rule SV-270753r1066748
    description: "Enable TCP SYN cookies to mitigate SYN flood Denial of Service (DoS) att
↪ acks."
    command: "sysctl -n net.ipv4.tcp_syncookies"
    expected: 1
    sudo: false
    fix: |
      sudo mkdir -p /etc/sysctl.d/
      sudo touch /etc/sysctl.d/90-network-hardening.conf
      echo 'net.ipv4.tcp_syncookies=1' | sudo tee -a /etc/sysctl.d/90-network-hardening.co
↪ nf
    fix_sudo: true
    affected_file: '/etc/sysctl.d/90-network-hardening.conf'
    post_action: "Run 'sudo sysctl -p /etc/sysctl.d/90-network-hardening.conf' to apply ch
↪ anges."
    security_level: 'baseline'
    arch: [x86_64, aarch64]
    risk_level: "low"
    risk_desc: "Activates only under high load/attack. No impact on normal traffic."

  - id: "26-net-02-rp-filter" # MISSING in CIS
    description: "Enable Reverse Path Filtering (Anti-spoofing) to validate source address
↪ es."
    command: "sysctl -n net.ipv4.conf.all.rp_filter"
```

```
    expected: 1
    sudo: false
    fix: "echo 'net.ipv4.conf.all.rp_filter=1' | sudo tee -a /etc/sysctl.d/90-network-hard
↪ ening.conf"
    fix_sudo: true
    affected_file: '/etc/sysctl.d/90-network-hardening.conf'
    post_action: "Run 'sudo sysctl -p /etc/sysctl.d/90-network-hardening.conf' to apply ch
↪ anges."
    security_level: 'baseline'
    arch: [x86_64, aarch64]
    risk_level: "medium"
    risk_desc: "Strict mode (1) may drop packets in asymmetric routing environments. Use l
↪ oose mode (2) if connectivity fails."

  - id: "27-net-03-accept-redirects" # MISSING in CIS
    description: "Disable acceptance of ICMP redirects to prevent routing table manipulati
↪ on (MITM)."
    command: "sysctl -n net.ipv4.conf.all.accept_redirects"
    expected: 0
    sudo: false
    fix: "echo 'net.ipv4.conf.all.accept_redirects=0' | sudo tee -a /etc/sysctl.d/90-netwo
↪ rk-hardening.conf"
    fix_sudo: true
    affected_file: '/etc/sysctl.d/90-network-hardening.conf'
    post_action: "Run 'sudo sysctl -p /etc/sysctl.d/90-network-hardening.conf' to apply ch
↪ anges."
    security_level: 'high'
    arch: [x86_64, aarch64]
    risk_level: "low"
    risk_desc: "Safe to disable unless the network relies on dynamic legacy routing update
↪ s."

  - id: "28-net-04-source-route" # MISSING in CIS
    description: "Disable acceptance of packets with source routing options."
    command: "sysctl -n net.ipv4.conf.all.accept_source_route"
    expected: 0
    sudo: false
    fix: "echo 'net.ipv4.conf.all.accept_source_route=0' | sudo tee -a /etc/sysctl.d/90-ne
↪ twork-hardening.conf"
    fix_sudo: true
    affected_file: '/etc/sysctl.d/90-network-hardening.conf'
    post_action: "Run 'sudo sysctl -p /etc/sysctl.d/90-network-hardening.conf' to apply ch
↪ anges."
    security_level: 'baseline'
```

```
    arch: [x86_64, aarch64]
    risk_level: "low"
    risk_desc: "Source routing is deprecated and rarely used; disabling has no operational
↪   risk."

  - id: "29-net-05-bogus-icmp" # MISSING in CIS
    description: "Ignore ICMP packets with bogus error responses to reduce log noise and p
↪ otential DoS."
    command: "sysctl -n net.ipv4.icmp_ignore_bogus_error_responses"
    expected: 1
    sudo: false
    fix: "echo 'net.ipv4.icmp_ignore_bogus_error_responses=1' | sudo tee -a /etc/sysctl.d/
↪ 90-network-hardening.conf"
    fix_sudo: true
    affected_file: '/etc/sysctl.d/90-network-hardening.conf'
    post_action: "Run 'sudo sysctl -p /etc/sysctl.d/90-network-hardening.conf' to apply ch
↪ anges."
    security_level: 'baseline'
    arch: [x86_64, aarch64]
    risk_level: "low"
    risk_desc: "Pure security measure; no impact on legitimate traffic."

  - id: "30-net-06-ipv4-ip_forward-disabled" # MISSING in CIS
    description: "Ensure IP forwarding is disabled."
    command: "sysctl -n net.ipv4.ip_forward"
    expected: 0
    sudo: true
    fix: |
      sysctl -w net.ipv4.ip_forward=0
      sed -i "/net.ipv4.ip_forward/d" /etc/sysctl.conf
      echo "net.ipv4.ip_forward = 0" >> /etc/sysctl.conf
    fix_sudo: true
    affected_file: '/etc/sysctl.conf'
    post_action: 'None'
    security_level: 'baseline'
    arch: [x86_64, aarch64]
    risk_level: "medium"
    risk_desc: "Prevents routing packets between network interfaces."

  - id: "31-net-07-ipv4-conf-all-send_redirects-disabled" # MISSING in CIS
    description: "Disable ICMP send_redirects."
    command: "sysctl -n net.ipv4.conf.all.send_redirects"
    expected: 0
    sudo: true
```

```
    fix: |
      sysctl -w net.ipv4.conf.all.send_redirects=0
      sed -i "/net.ipv4.conf.all.send_redirects/d" /etc/sysctl.conf
      echo "net.ipv4.conf.all.send_redirects = 0" >> /etc/sysctl.conf
    fix_sudo: true
    affected_file: '/etc/sysctl.conf'
    post_action: 'None'
    security_level: 'baseline'
    arch: [x86_64, aarch64]
    risk_level: "low"
    risk_desc: "Safe security change for standard operations."

  # --- Module & Crypto Hardening ---

  - id: "32-mod-01-disable-usb-storage" # CIS Rule SV-270718r1067128
    description: "Prevent the 'usb-storage' kernel module from loading."
    command: |
      grep -r 'install usb-storage /bin/true' /etc/modprobe.d/ | wc -l
    expected: 1
    sudo: false
    fix: |
      echo 'install usb-storage /bin/true' | sudo tee /etc/modprobe.d/blacklist-usb-storag
↪ e.conf
    fix_sudo: true
    affected_file: '/etc/modprobe.d/blacklist-usb-storage.conf'
    post_action: "Reboot required to unload active modules."
    security_level: 'high'
    arch: [x86_64, aarch64]
    risk_level: "medium"
    risk_desc: "Disables ALL USB mass storage devices."

  - id: "33-crypto-01-fips-mode" # CIS Rule SV-270744r1066721
    description: "Ensure the OS implements NIST FIPS-validated cryptography (Auditing only
↪ )."
    command: "cat /proc/sys/crypto/fips_enabled"
    expected: 1
    sudo: false
    fix: "Manual action required: Enabling FIPS requires installing certified packages (e.
↪ g., ubuntu-fips) and modifying the bootloader. Automated fixing is unsafe."
    fix_sudo: false
    affected_file: '/boot/grub/grub.cfg'
    post_action: "N/A"
    security_level: 'high'
    arch: [x86_64, aarch64]
```

```
    risk_level: "high"
    risk_desc: "Enabling FIPS often disables non-compliant crypto (e.g., certain SSH ciphe
↪ rs), potentially locking out remote access."
---
...
```

Filesystem.md

```
---
title: Filesystem Hardening
label: "82-filesystem-hardening"
template:
    type: section
    id: 82
lang: "en-US"
checksuites:
  # --- File Permissions & Ownership ---
  - id: "34-fs-01-shadow-perms" # MISSING in CIS
    description: "Ensure /etc/shadow permissions are 640 and owned by root:shadow."
    command: |
      stat -c '%a %U:%G' /etc/shadow | grep -c '640 root:shadow'
    expected: 1
    sudo: true
    fix: "chown root:shadow /etc/shadow && chmod 640 /etc/shadow"
    fix_sudo: true
    affected_file: '/etc/shadow'
    post_action: 'None'
    security_level: 'baseline'
    arch: [x86_64, aarch64, amd64]
    risk_level: "low"
    risk_desc: "Standard security control. No impact on system function."

  - id: "35-fs-02-passwd-perms" # MISSING in CIS
    description: "Ensure /etc/passwd permissions are 644 and owned by root:root."
    command: |
      stat -c '%a %U:%G' /etc/passwd | grep -c '644 root:root'
    expected: 1
    sudo: true
    fix: "chown root:root /etc/passwd && chmod 644 /etc/passwd"
    fix_sudo: true
    affected_file: '/etc/passwd'
    post_action: 'None'
    security_level: 'baseline'
    arch: [x86_64, aarch64, amd64]
```

```
    risk_level: "low"
    risk_desc: "Standard security control."

  - id: "36-fs-03-group-perms" # MISSING in CIS
    description: "Ensure /etc/group permissions are 644 and owned by root:root."
    command: |
      stat -c '%a %U:%G' /etc/group | grep -c '644 root:root'
    expected: 1
    sudo: true
    fix: "chown root:root /etc/group && chmod 644 /etc/group"
    fix_sudo: true
    affected_file: '/etc/group'
    post_action: 'None'
    security_level: 'baseline'
    arch: [x86_64, aarch64, amd64]
    risk_level: "low"
    risk_desc: "Standard security control."

  - id: "37-fs-04-bootloader-password" # CIS Rule SV-270675r1066514
    description: "Verify the bootloader requires a password for single-user mode."
    command: |
      grep -i '^password_pbkdf2' /boot/grub/grub.cfg | wc -l
    expected: 1
    sudo: true
    fix: |
      Manual action required: Run 'grub-mkpasswd-pbkdf2' and configure /etc/grub.d/00_head
↪ er.
    fix_sudo: false
    affected_file: '/boot/grub/grub.cfg'
    post_action: 'None'
    security_level: 'high'
    arch: [x86_64, aarch64, amd64]
    risk_level: "high"
    risk_desc: "Misconfiguration can render the system unbootable or lock out administrato
↪ rs."

  - id: "38-fs-05-sticky-bit" # CIS Rule SV-270750r1066739
    description: "Ensure the Sticky Bit is set on all world-writable directories."
    command: |
      find / -xdev -type d \( -perm -0002 -a ! -perm -1000 \) | wc -l
    expected: 0
    sudo: true
    fix: |
      find / -xdev -type d \( -perm -0002 -a ! -perm -1000 \) -exec chmod +t {} \;
```

```
    fix_sudo: true
    affected_file: 'Multiple Directories'
    post_action: 'None'
    security_level: 'baseline'
    arch: [x86_64, aarch64, amd64]
    risk_level: "low"
    risk_desc: "Prevents users from deleting each other's files in shared directories."

  # --- Authentication & Password Policy ---

  - id: "39-auth-01-pass-encryption" # CIS Rule SV-270739r1067124
    description: "Ensure passwords use SHA512 hashing."
    command: |
      grep -E '^ENCRYPT_METHOD SHA512' /etc/login.defs | wc -l
    expected: 1
    sudo: false
    fix: |
      sed -i 's/^ENCRYPT_METHOD.*/ENCRYPT_METHOD SHA512/' /etc/login.defs
    fix_sudo: true
    affected_file: '/etc/login.defs'
    post_action: 'Only affects new passwords.'
    security_level: 'baseline'
    arch: [x86_64, aarch64, amd64]
    risk_level: "low"
    risk_desc: "Ensures compliance for new accounts."

  - id: "40-auth-02-pass-complexity" # CIS Rule SV-270732r1066685
    description: "Ensure password complexity is enforced (minlen=14)."
    command: |
      grep -E 'minlen=14' /etc/security/pwquality.conf | wc -l
    expected: 1
    sudo: true
    fix: |
      echo 'minlen=14' | sudo tee -a /etc/security/pwquality.conf
    fix_sudo: true
    affected_file: '/etc/security/pwquality.conf'
    post_action: 'None'
    security_level: 'baseline'
    arch: [x86_64, aarch64, amd64]
    risk_level: "medium"
    risk_desc: "Does not force reset of existing passwords."

  - id: "41-auth-03-pass-lifetime" # CIS Rule SV-270731r1066682
    description: "Ensure maximum password days is 60 or less."
```

```
    command: |
      grep '^PASS_MAX_DAYS' /etc/login.defs | awk '{if($2<=60) print 1; else print 0}'
    expected: 1
    sudo: false
    fix: |
      sed -i 's/^PASS_MAX_DAYS.*/PASS_MAX_DAYS 60/' /etc/login.defs
    fix_sudo: true
    affected_file: '/etc/login.defs'
    post_action: 'None'
    security_level: 'baseline'
    arch: [x86_64, aarch64, amd64]
    risk_level: "medium"
    risk_desc: "Existing users must be updated manually via 'chage'."

  - id: "42-auth-04-cached-auth" # CIS Rule SV-270734r1066691
    description: "Prohibit PAM from using cached credentials for more than 1 day (SSSD)."
    command: |
      grep -r 'offline_credentials_expiration' /etc/sssd/ | wc -l
    expected: 1
    sudo: true
    fix: "Manual action required: Configure offline_credentials_expiration = 1 in /etc/sss
↪ d/sssd.conf"
    fix_sudo: false
    affected_file: '/etc/sssd/sssd.conf'
    post_action: 'None'
    security_level: 'high'
    arch: [x86_64, aarch64]
    risk_level: "medium"
    risk_desc: "If the domain controller is unreachable for >1 day, users will be locked o
↪ ut of the laptop/server."

  - id: "43-auth-05-root-lock" # CIS Rule SV-270724r1066661
    description: "Ensure the root account is locked."
    command: |
      sudo passwd -S root | grep -c 'L'
    expected: 1
    sudo: true
    fix: "sudo passwd -l root"
    fix_sudo: true
    affected_file: '/etc/shadow'
    post_action: 'None'
    security_level: 'baseline'
    arch: [x86_64, aarch64, amd64]
    risk_level: "medium"
```

```
    risk_desc: "Requires a valid sudo user before applying."

  - id: "44-auth-06-umask-strict" # CIS Rule SV-270716r1066637
    description: "Define default UMASK (077)."
    command: |
      grep -i '^\s*umask' /etc/login.defs | grep -v '000' | grep -c '077'
    expected: 1
    sudo: false
    fix: |
      if grep -q "^UMASK" /etc/login.defs; then
        sudo sed -i "s/^UMASK.*/UMASK 077/" /etc/login.defs
      else
        echo "UMASK 077" | sudo tee -a /etc/login.defs
      fi
    fix_sudo: true
    affected_file: '/etc/login.defs'
    post_action: 'None. Affects home directories for new users.'
    security_level: 'high'
    arch: [x86_64, aarch64, amd64]
    risk_level: "medium"
    risk_desc: "Restrictive; may break shared group data directories."

  - id: "45-fs-07-find-suid" # MISSING in CIS
    description: "Ensure the 'find' binary does not have the SUID bit set."
    command: |
      stat -c '%a' $(which find) | grep -c '4'
    expected: 0
    sudo: false
    fix: |
      chmod u-s $(which find)
    fix_sudo: true
    affected_file: '/usr/bin/find'
    post_action: 'None'
    security_level: 'baseline'
    arch: [x86_64, aarch64, amd64]
    risk_level: "low"
    risk_desc: "SUID on 'find' is almost always a security vulnerability."
---
...
```

Services.md

```
---
title: Service Hardening
```

```
label: "83-services-hardening"
template:
    type: section
    id: 83
lang: "en-US"
checksuites:
  # --- Advanced Intrusion Detection Environment (AIDE) ---
  - id: "46-aide-package-installed" # CIS Rule SV-270649r1067136
    description: "Verify that the AIDE (Advanced Intrusion Detection Environment) package
↪ is installed for file integrity monitoring."
    command: |
      dpkg -l aide 2>/dev/null | grep -c '^ii'
    expected: 1
    sudo: false
    fix: "sudo apt install aide aide-common"
    fix_sudo: true
    affected_file: 'System package list'
    post_action: "Run 'sudo aideinit' to initialize the baseline database."
    security_level: 'baseline'
    arch: [x86_64, aarch64]
    risk_level: "low"
    risk_desc: "Installation is low risk. Initializing the database may take several minut
↪ es and consume CPU/IO."

  - id: "47-aide-conf-audit-binaries" # CIS Rule SV-270831r1066982
    description: "Ensure AIDE uses cryptographic mechanisms (SHA512) to protect the integr
↪ ity of audit tools (auditctl, auditd, ausearch, aureport, autrace, augenrules)."
    command: |
      grep -E '(/sbin/(audit|au))' /etc/aide/aide.conf | grep -c 'sha512'
    expected: 6
    sudo: true
    fix: |
      for bin in auditctl auditd ausearch aureport autrace augenrules; do
        if ! grep -q "/sbin/$bin" /etc/aide/aide.conf; then
          echo "/sbin/$bin p+i+n+u+g+s+b+acl+xattrs+sha512" | sudo tee -a /etc/aide/aide.c
↪ onf
        fi
      done
    fix_sudo: true
    affected_file: '/etc/aide/aide.conf'
    post_action: "Run 'sudo aide --update' to initialize the integrity database with the n
↪ ew definitions."
    security_level: 'high'
    arch: [x86_64, aarch64, amd64]
```

```
    risk_level: "low"
    risk_desc: "Critical for identifying if audit tools have been replaced or modified by
↪ attackers. Essential for ensuring the integrity of audit settings and reports."

  - id: "48-auditd-package-installed" # CIS Rule SV-270656r1067148
    description: "Verify that the auditd package, which provides the Linux Auditing Framew
↪ ork daemon, is installed on the system."
    command: |
      dpkg -l auditd 2>/dev/null | grep -c '^ii'
    expected: 1
    sudo: false
    fix: "sudo apt install auditd"
    fix_sudo: true
    affected_file: 'System package list'
    post_action: 'None'
    security_level: 'baseline'
    arch: [x86_64, aarch64]
    risk_level: "low"
    risk_desc: "Installing the auditd package is a prerequisite for security monitoring an
↪ d does not affect system stability."

  - id: "49-auditd-service-enabled" # CIS Rule SV-270657r1066460
    description: "Verify that the auditd service is enabled to ensure that the auditing fr
↪ amework starts automatically upon system boot."
    command: |
      systemctl is-enabled auditd | grep -c 'enabled'
    expected: 1
    sudo: false
    fix: "sudo systemctl enable auditd"
    fix_sudo: true
    affected_file: 'System service configuration'
    post_action: 'None'
    security_level: 'baseline'
    arch: [x86_64, aarch64, amd64]
    risk_level: "low"
    risk_desc: "Enabling the service is essential for persistent security monitoring and p
↪ oses a minimal risk."

  - id: "50-auditd-service-active" # CIS Rule SV-270657r1066460
    description: "Verify that the auditd service is currently running actively to ensure c
↪ ontinuous security event logging."
    command: |
      systemctl is-active auditd | grep -c 'active'
    expected: 1
```

```
    sudo: false
    fix: "sudo systemctl start auditd"
    fix_sudo: true
    affected_file: 'System service runtime'
    post_action: 'None'
    security_level: 'baseline'
    arch: [x86_64, aarch64, amd64]
    risk_level: "low"
    risk_desc: "Starting the service introduces a slight performance overhead, but the sec
↪ urity benefits are mandatory."

  - id: "51-audit-identity-file-changes" # CIS Rule SV-270684r1066541, SV-270686r1066547
    description: "Verify rules exist to monitor all write and attribute changes to critica
↪ l identity files (/etc/passwd, /etc/shadow, /etc/sudoers)."
    command: |
        grep -cE '^\s*-w\s+/etc/shadow\s+-p\s+wa' /etc/audit/rules.d/*.rules
    expected: 1
    sudo: false
    fix: "Manual action required: Add rules to monitor all critical identity files for wri
↪ te access (wa). See text for full list of mandatory rules."
    fix_sudo: true
    affected_file: '/etc/audit/rules.d/90-identity.rules'
    post_action: |
        Run 'sudo auditctl -R' to load the new ruleset immediately.
    security_level: 'baseline'
    arch: [x86_64, aarch64]
    risk_level: "low"
    risk_desc: "Incorrect audit rules may fail to log critical events, but will not affect
↪  system stability."

  - id: "52-audit-privilege-executables" # CIS Rule SV-270689r1066556
    description: "Verify rules exist to monitor the successful execution of privilege esca
↪ lation binaries (su and sudo)."
    command: |
        grep -cE '^\s*-a\s+always,exit\s+-F\s+path=/usr/bin/su' /etc/audit/rules.d/*.rules
    expected: 1
    sudo: false
    fix: "Manual action required: Add execution monitoring rules for /usr/bin/su and /usr/
↪ bin/sudo."
    fix_sudo: true
    affected_file: '/etc/audit/rules.d/90-identity.rules'
    post_action: |
        Run 'sudo auditctl -R' to load the new ruleset immediately.
    security_level: 'baseline'
```

```
    arch: [x86_64, aarch64]
    risk_level: "low"
    risk_desc: "Monitoring execution of binaries is low risk, but failure to monitor may a
↪ llow an attacker to escalate privileges without detection."

  - id: "53-audit-file-attribute-changes" # CIS Rule SV-270786r1068375
    description: "Verify the audit system generates records for successful/unsuccessful us
↪ e of 'chmod', 'fchmod', and 'fchmodat' syscalls."
    command: |
      sudo auditctl -l | grep -cE 'chmod,fchmod,fchmodat.*arch=b64'
    expected: 1
    sudo: true
    fix: |
      RULE_FILE="/etc/audit/rules.d/stig.rules"
      echo "-a always,exit -F arch=b32 -S chmod,fchmod,fchmodat -F auid>=1000 -F auid!=-1
↪ -k perm_chng" | sudo tee -a $RULE_FILE
      echo "-a always,exit -F arch=b64 -S chmod,fchmod,fchmodat -F auid>=1000 -F auid!=-1
↪ -k perm_chng" | sudo tee -a $RULE_FILE
    fix_sudo: true
    affected_file: '/etc/audit/rules.d/stig.rules'
    post_action: "Run 'sudo augenrules --load' to apply the new syscall monitoring rules."
    security_level: 'baseline'
    arch: [x86_64, aarch64, amd64]
    risk_level: "low"
    risk_desc: "Monitoring file permission changes is critical for detecting unauthorized
↪ privilege escalation or tampering."

  - id: "54-audit-file-deletion-activity" # CIS Rule SV-270809r1068388
    description: "Verify rules exist to monitor syscalls related to file and directory del
↪ etion (e.g., unlink, rmdir) across the filesystem."
    command: |
        grep -cE '^\s*-a\s+always,exit\s+-S\s+unlink' /etc/audit/rules.d/*.rules
    expected: 1
    sudo: false
    fix: "Manual action required: Add the mandatory syscall rules for file deletion/renami
↪ ng."
    fix_sudo: true
    affected_file: '/etc/audit/rules.d/91-system-access.rules'
    post_action: |
        Run 'sudo auditctl -R' to load the new ruleset immediately.
    security_level: 'baseline'
    arch: [x86_64, aarch64]
    risk_level: "low"
    risk_desc: "This monitoring is essential for forensic analysis after a file integrity
```

```
↪ violation."

  - id: "55-audit-kernel-module-loading"  # CIS Rule SV-270805r106838
    description: "Verify rules exist to monitor kernel integrity by logging attempts to lo
↪ ad or unload kernel modules (init_module, delete_module)."
    command: |
        grep -cE '^\s*-a\s+always,exit\s+-S\s+init_module' /etc/audit/rules.d/*.rules
    expected: 1
    sudo: false
    fix: "Manual action required: Add the syscall rules for kernel module management."
    fix_sudo: true
    affected_file: '/etc/audit/rules.d/91-system-access.rules'
    post_action: |
        Run 'sudo auditctl -R' to load the new ruleset immediately.
    security_level: 'high'
    arch: [x86_64, aarch64]
    risk_level: "low"
    risk_desc: "Monitoring kernel module activity is critical for detecting rootkits and p
↪ oses zero performance overhead."

  - id: "56-audit-immutable-rule-config" # MISSING in CIS
    description: "Verify that the mandatory rule to set the audit configuration to immutab
↪ le is the last line in the ruleset."
    command: |
      sudo tail -n 1 /etc/audit/rules.d/99-finalize.rules 2>/dev/null | grep -c '^-e 2'
    expected: 1
    sudo: true
    fix: |
      echo '-e 2' | sudo tee /etc/audit/rules.d/99-finalize.rules
    fix_sudo: true
    affected_file: '/etc/audit/rules.d/99-finalize.rules'
    post_action: 'A system reboot is MANDATORY to load the immutable rule and lock the con
↪ figuration.'
    security_level: 'high'
    arch: [x86_64, aarch64]
    risk_level: "high"
    risk_desc: "Once set to immutable, the audit rules cannot be changed or cleared withou
↪ t a system reboot. This creates a risk if the ruleset is flawed and causes a Denial of
↪ Service (DoS) due to excessive logging."

  - id: "57-audit-immutable-mode-runtime" # MISSING in CIS
    description: "Verify that the kernel's audit system is currently operating in immutabl
↪ e mode, preventing runtime tampering."
    command: |
```

```
      sudo auditctl -s | grep -c 'enabled 2'
    expected: 1
    sudo: true
    fix: "Manual action required: Reboot the system to ensure the immutable rule is loaded
↪   by the auditd service."
    fix_sudo: true
    affected_file: 'Kernel runtime status'
    post_action: 'None'
    security_level: 'high'
    arch: [x86_64, aarch64]
    risk_level: "low"
    risk_desc: "This is purely a verification check for the most critical audit security f
↪ eature."
---
...
```

C Appendix C: Lynis Analysis of Controls

SSH

Check ID	Control Name (STIG / CIS Rule)	Lynis Result	Observation
00	Full SSH Stack (SV-270665)	Fragmented	Checked service states ([UNSAFE], [FOUND]) scattered across the log, but failed to systematically verify the installation of the full client/server/sftp package stack as required by the STIG rule.
01	Disable Root Login (Non-CIS)	Missed	Undetected. Fails to identify one of the most critical access control vectors.
02	Disable Password Auth	Missed	Undetected. Does not enforce the transition to key-based authentication.
04	Max Auth Tries	Missed	Undetected. Missing brute-force mitigation checks.
05	Use Pubkey Auth (SV-270722)	Missed	Undetected.
06	Disable Empty Passwords (SV-270717)	Detected	Flagged as Accounts without password [OK]. However, it lacks cognitive transparency; it doesn't show the user the exact /etc/ssh/sshd_config parameter it checked.
07	Enable PAM (SV-270741)	Detected	Flagged as PAM modules [FOUND]. Mentioned multiple times, indicating repetitive checks without clear remediation context.
08	Set Idle Timeout (SV-270743)	Missed	Undetected.
09	Disable X11 Forwarding (SV-270708)	Missed	Undetected.
10	FIPS Ciphers (SV-270667)	Missed	Undetected. Fails to perform deep cryptographic policy validation.
11	Strong MACs (SV-270668)	Missed	Undetected.

Check ID	Control Name (STIG / CIS Rule)	Lynis Result	Observation
12	Strong KEX (SV-270669)	Missed	Undetected.
13	Secure X11 Proxy (SV-270709)	Missed	Undetected.

Kernel

Check ID	Control Name (STIG / CIS Rule)	Lynis Result	Observation
14	Restrict dmesg (SV-270749)	Detected ([OK])	Good cognitive transparency. Explicitly lists the sysctl parameter (kernel.dmesg_restrict) and the expected value (exp: 1).
15	ASLR / VA Space (SV-270772)	Detected ([OK])	Correctly identified the sysctl parameter and expected value.
16	Disable Core Dumps (Non-CIS)	Detected ([DIFFERENT])	Flagged the discrepancy in fs.suid_dumpable.
17	Yama Ptrace Scope (Non-CIS)	Detected ([OK])	Correctly identified and evaluated against a range of acceptable hardened values (exp: 1 2 3).
18	kptr_restrict (Non-CIS)	Detected ([DIFFERENT])	Flagged the discrepancy in kernel pointer exposure.
19	Disable unpriv eBPF (Non-CIS)	Detected ([DIFFERENT])	Flagged the discrepancy.
20	eBPF JIT Hardening (Non-CIS)	Detected ([DIFFERENT])	Flagged the discrepancy in net.core.bpf_jit_harden.
21	Disable UserNS Clone (Non-CIS)	Missed	Undetected. Missing a critical container-escape mitigation.
22	Protected FIFOs (Non-CIS)	Detected ([DIFFERENT])	Flagged the discrepancy in fs.protected_fifos.

Check ID	Control Name (STIG / CIS Rule)	Lynis Result	Observation
23	Blacklist DCCP (Non-CIS)	Partial (NETW-3200)	Vague recommendation. Asks the user to "Determine if protocol is really needed" rather than verifying a strict blacklist state.
24	Blacklist SCTP (Non-CIS)	Partial (NETW-3200)	Vague recommendation, same as DCCP.
25	SYN Cookies (SV-270753)	Detected ([OK])	Correctly identified the IPv4 parameter.
26	Reverse Path Filter (Non-CIS)	Detected ([DIFFERENT])	Flagged the discrepancy in IP spoofing protection.
27	Accept Redirects (Non-CIS)	Detected ([DIFFERENT])	Flagged the discrepancy in ICMP routing.
28	Source Route (Non-CIS)	Detected ([OK])	Correctly identified the IPv4 parameter.
29	Bogus ICMP (Non-CIS)	Detected ([OK])	Correctly identified the IPv4 parameter.
30	Disable IP Forwarding (Non-CIS)	Missed	Undetected. Critical network isolation check missed.
31	Send Redirects (Non-CIS)	Detected ([DIFFERENT])	Flagged the discrepancy in ICMP routing.
32	Disable USB Storage (SV-270718)	Detected ([NOT DISABLED])	Correctly evaluated the modprobe configuration status.
33	FIPS Mode (SV-270744)	Missed	Undetected. Missing core cryptographic state validation.

Filesystem

Check ID	Control Name (STIG / CIS Rule)	Lynis Result	Observation
34	Shadow Permissions (Non-CIS)	Missed	Undetected. Fails to verify permissions on the critical password hash file.
35	Passwd Permissions (Non-CIS)	Detected ([OK])	Checked under "File Permissions," but lacks Cognitive Transparency. It does not display the expected octal value (e.g., 0644), leaving the user guessing what the standard is.
36	Group Permissions (Non-CIS)	Detected ([OK])	Same issue as fs-02. Opaque evaluation criteria.
37	Bootloader Password (SV-270675)	Missed	False Positive Detection: Lynis checks the file permissions of /boot/grub/grub.cfg, but completely fails to inspect the file's contents for the required password_pbkdf2 directive.
38	Sticky Bit Usage (SV-270750)	Partial ([OK])	Incomplete scope. Lynis statically checks /tmp and /var/tmp, whereas the STIG standard requires a dynamic search of all world-writable directories.
39	Password Encryption (SV-270739)	Missed	Undetected. Fails to verify hashing algorithms (SHA512/yescrypt).
40	Password Complexity (SV-270732)	Detected (AUTH-9286)	Conflated with password age, but provides a direct link to an external article on minimum password length. Good educational value, but poor automated remediation.
41	Password Lifetime (SV-270731)	Detected (AUTH-9286)	Identifies missing max password age and provides documentation links.
42	Cached Auth (SV-270734)	Missed	Undetected. Missing SSSD credential expiration checks.
43	Root Account Lock (SV-270724)	Missed	Undetected. Fails to check if direct local root login is locked in /etc/shadow.
44	Strict Umask (SV-270716)	Partial ([NONE])	Checks shell profiles (bash.bashrc, profile) but misses the centralized /etc/login.defs enforcement mandated by the STIG.
45	Find SUID Bit (Non-CIS)	Missed	Undetected.

Services

Check ID	Control Name (STIG / CIS Rule)	Lynis Result	Observation
46	AIDE Installed (SV-270649)	Detected [[FOUND]]	Correctly identified the installation of the AIDE package.
47	Audit Binaries Conf (SV-270831)	Passed [[FOUND]]	Found the config file and suggested cryptographic improvements (FINT-4402), though it did not explicitly verify the presence of the six mandatory audit binaries.
48	Auditd Installed (SV-270656)	Detected [[ENABLED]]	Verified package installation.
49	Auditd Enabled (SV-270657)	Detected [[ENABLED]]	Verified that the service is configured to start on boot.
50	Auditd Active (SV-270657)	Detected [[FOUND]]	Confirmed via the presence of the active auditd log file.
51	Identity File Changes	Missed	Undetected. Fails to verify specific syscall monitoring for user/group modifications.
52	Privilege Executables	Missed	Undetected. Fails to verify monitoring for SUID/SGID binary execution.
53	File Attribute Changes	Missed	Undetected. Fails to verify monitoring for chmod/chown syscalls.
54	File Deletion Activity	Missed	Undetected.
55	Kernel Module Loading	Missed	Undetected.
56	Immutable Rule Config	Missed	Undetected. Fails to check for the -e 2 flag in the audit rules.
57	Immutable Mode Runtime	Missed	Undetected.

D Appendix D: Oscap Analysis of Controls

SSH

Check ID	Control Name (STIG / CIS Rule)	OpenSCAP Result	Observation
01	Full SSH Stack (SV-270665)	fail	Successfully evaluated; correctly identified missing server packages/services.
02	Disable Root Login (Non-CIS)	Undetected	While a general root login rule exists, a specific SSH-targeted control was missing in the utilized profile.
03	Disable Password Auth (Non-CIS)	Undetected	Explicit rule missing from the baseline configuration profile.
04	Max Auth Tries (Non-CIS)	Undetected	Explicit rule missing from the baseline configuration profile.
05	Use Pubkey Auth (SV-270722)	pass	Functionally executed; verified Ubuntu's secure default configuration.
06	Disable Empty Passwords (SV-270717)	pass	Functionally executed; verified Ubuntu's secure default configuration.
07	Enable PAM (SV-270741)	pass	Functionally executed; verified Ubuntu's secure default configuration.
08	Set Idle Timeout (SV-270743)	pass	Functionally executed; verified Ubuntu's secure default configuration.
09	Disable X11 Forwarding (SV-270708)	pass	Functionally executed; verified Ubuntu's secure default configuration.
10	FIPS Ciphers (SV-270667)	fail	Functionally executed; correctly identified non-FIPS compliant ciphers.
11	Strong MACs (SV-270668)	fail	Functionally executed; correctly identified non-FIPS compliant MACs.
12	Strong KEX (SV-270669)	pass	Functionally executed; verified Ubuntu's secure default configuration.
13	Secure X11 Proxy (SV-270709)	pass	Functionally executed; verified Ubuntu's secure default configuration.

Kernel

Check ID	Control Name (STIG / CIS Rule)	OpenSCAP Result	Observation
14	Restrict dmesg (SV-270749)	fail	Functionally executed; correctly identified misconfigured sysctl parameter.
15	ASLR / VA Space (SV-270772)	fail	Functionally executed; correctly identified misconfigured sysctl parameter.
16	Disable Core Dumps (Non-CIS)	Undetected	Missing from the Ubuntu 24.04 baseline profile despite its security relevance.
17	Yama Ptrace Scope (Non-CIS)	Undetected	Explicit rule missing from the baseline configuration profile.
18	kptr_restrict (Non-CIS)	Undetected	Explicit rule missing from the baseline configuration profile.
19	Disable unpriv eBPF (Non-CIS)	Undetected	Explicit rule missing; highlights the gap in contemporary attack surface reduction.
20	eBPF JIT Hardening (Non-CIS)	Undetected	Explicit rule missing from the baseline configuration profile.
21	Disable UserNS Clone (Non-CIS)	Undetected	Missing critical container-hardening control.
22	Protected FIFOs (Non-CIS)	Undetected	Explicit rule missing from the baseline configuration profile.
23	Blacklist DCCP (Non-CIS)	Undetected	Protocol blacklisting not covered in this profile.
24	Blacklist SCTP (Non-CIS)	Undetected	Protocol blacklisting not covered in this profile.
25	SYN Cookies (SV-270753)	fail	Functionally executed; correctly identified misconfigured sysctl parameter.
26	Reverse Path Filter (Non-CIS)	Undetected	Explicit rule missing from the baseline configuration profile.
27	Accept Redirects (Non-CIS)	Undetected	Explicit rule missing from the baseline configuration profile.
28	Source Route (Non-CIS)	Undetected	Explicit rule missing from the baseline configuration profile.

Check ID	Control Name (STIG / CIS Rule)	OpenSCAP Result	Observation
29	Bogus ICMP (Non-CIS)	Undetected	Explicit rule missing from the baseline configuration profile.
30	Disable IP Forwarding (Non-CIS)	Undetected	Explicit rule missing from the baseline configuration profile.
31	Send Redirects (Non-CIS)	Undetected	Explicit rule missing from the baseline configuration profile.
32	Disable USB Storage (SV-270718)	fail	Functionally executed; correctly identified missing modprobe blacklist.
33	FIPS Mode (SV-270744)	fail	Functionally executed; correctly identified missing FIPS enablement in `/proc`.

Filesystem

Check ID	Control Name (STIG / CIS Rule)	OpenSCAP Result	Observation
34	Shadow Permissions (Non-CIS)	Undetected	Explicit rule missing from the baseline configuration profile.
35	Passwd Permissions (Non-CIS)	Undetected	Explicit rule missing from the baseline configuration profile.
36	Group Permissions (Non-CIS)	Undetected	Explicit rule missing from the baseline configuration profile.
37	Bootloader Password (SV-270675)	fail	Functionally executed; correctly identified missing GRUB password.
38	Sticky Bit Usage (SV-270750)	pass	Functionally executed; verified dynamic directory permissions.

Check ID	Control Name (STIG / CIS Rule)	OpenSCAP Result	Observation
39	Password Encryption (SV-270739)	pass	Functionally executed; verified login.defs hashing configuration.
40	Password Complexity (SV-270732)	fail	Functionally executed; correctly identified weak PAM length requirements.
41	Password Lifetime (SV-270731)	fail	Functionally executed; correctly identified weak password age limits.
42	Cached Auth (SV-270734)	fail	Functionally executed; correctly identified missing SSSD expiration limits.
43	Root Account Lock (SV-270724)	pass	Functionally executed; verified root account lockout status.
44	Strict Umask (SV-270716)	fail	Functionally executed; correctly identified weak default UMASK.
45	Find SUID Bit (Non-CIS)	Undetected	Explicit rule for identifying all SUID binaries missing from the profile.

Services

Check ID	Control Name (STIG / CIS Rule)	OpenSCAP Result	Observation
46	AIDE Installed (SV-270649)	fail	Functionally executed; correctly identified missing AIDE package.
47	Audit Binaries Conf (SV-270831)	Undetected	Explicit rule for specific integrity monitoring of audit binaries missing from profile.
48	Auditd Installed (SV-270656)	fail	Functionally executed; correctly identified missing auditd subsystem.
49	Auditd Enabled (SV-270657)	notapplicable	Aborted. OVAL logic skips service enablement checks if the base package (Check 48) is missing.
50	Auditd Active (SV-270657)	Undetected	Verification of the live runtime state of the daemon was not present as a distinct control.

Check ID	Control Name (STIG / CIS Rule)	OpenSCAP Result	Observation
51	Identity File Changes (SV-270684)	notapplicable	Aborted. Dependent rule; skips evaluation because the auditd package is not installed.
52	Privilege Executables (SV-270689)	notapplicable	Aborted. Dependent rule; skips evaluation because the auditd package is not installed.
53	File Attribute Changes (SV-270786)	Partial / Unclear	Potential matches exist in XML logic, but lack of definitive execution prevents precise mapping.
54	File Deletion Activity (SV-270809)	Partial / Unclear	Potential matches exist in XML logic, but lack of definitive execution prevents precise mapping.
55	Kernel Module Loading (SV-270805)	Partial / Unclear	Potential matches exist in XML logic, but lack of definitive execution prevents precise mapping.
56	Immutable Rule Config (Non-CIS)	notapplicable	Aborted. Dependent rule; skips evaluation because the auditd package is not installed.
57	Immutable Mode Runtime (Non-CIS)	Undetected	Verification of the immutable runtime state was not identified in the profile logic.

E Appendix E: OpenSCAP Post-Remediation Report

The undetected controls were omitted here.

SSH

Check ID	Control Name (STIG / CIS Rule)	OpenSCAP Result	Observation
01	Full SSH Stack (SV-270665)	pass	Successfully remediated.
05	Use Pubkey Auth (SV-270722)	fail	Presented `pass` before remediation!
06	Disable Empty Passwords (SV-270717)	fail	Presented `pass` before remediation!
07	Enable PAM (SV-270741)	pass	Unchanged.
08	Set Idle Timeout (SV-270743)	fail	Presented `pass` before remediation!
09	Disable X11 Forwarding (SV-270708)	fail	Presented `pass` before remediation!
10	FIPS Ciphers (SV-270667)	pass	Successfully remediated.
11	Strong MACs (SV-270668)	pass	Successfully remediated.
12	Strong KEX (SV-270669)	fail	Presented `pass` before remediation!
13	Secure X11 Proxy (SV-270709)	fail	Presented `pass` before remediation!

Kernel

Check ID	Control Name (STIG / CIS Rule)	OpenSCAP Result	Observation
14	Restrict dmesg (SV-270749)	pass	Successfully remediated.
15	ASLR / VA Space (SV-270772)	pass	Successfully remediated.
25	SYN Cookies (SV-270753)	pass	Successfully remediated.
32	Disable USB Storage (SV-270718)	pass	Successfully remediated.
33	FIPS Mode (SV-270744)	fail	Unchanged.

Filesystem

Check ID	Control Name (STIG / CIS Rule)	OpenSCAP Result	Observation
37	Bootloader Password (SV-270675)	fail	Unchanged.
38	Sticky Bit Usage (SV-270750)	pass	Unchanged.
39	Password Encryption (SV-270739)	pass	Unchanged.
40	Password Complexity (SV-270732)	pass	Successfully remediated.
41	Password Lifetime (SV-270731)	pass	Successfully remediated.
42	Cached Auth (SV-270734)	pass	Successfully remediated.

Check ID	Control Name (STIG / CIS Rule)	OpenSCAP Result	Observation
43	Root Account Lock (SV-270724)	pass	Unchanged.
44	Strict Umask (SV-270716)	pass	Successfully remediated.

Services

Check ID	Control Name (STIG / CIS Rule)	OpenSCAP Result	Observation
46	AIDE Installed (SV-270649)	pass	Successfully remediated.
48	Auditd Installed (SV-270656)	pass	Successfully remediated.
49	Auditd Enabled (SV-270657)	pass	Successfully remediated, changed from `notapplicable`
51	Identity File Changes (SV-270684)	fail	Changed from `notapplicable`.
52	Privilege Executables (SV-270689)	notapplicable	Unchanged.
53	File Attribute Changes (SV-270786)	Partial / Unclear	Unchanged.
54	File Deletion Activity (SV-270809)	Partial / Unclear	Unchanged.
55	Kernel Module Loading (SV-270805)	Partial / Unclear	Unchanged.
56	Immutable Rule Config (Non-CIS)	notapplicable	Unchanged.

F Appendix F: Hardener Execution Report

The following tables document the execution lifecycle of the *Hardener* framework across the Ubuntu 25.10 test environment. It tracks the initial diagnostic audit, the automated remediation phase, and the subsequent post-remediation audit to evaluate actual compliance enforcement.

SSH Hardening

Check ID	Control Name	Initial Audit	Hardener Action (Fix Mode)	Post-Fix Audit	Analytical Notes
01	ssh-full-stack-installed	Failed	Fixed (apt install)	Passed	Successfully installed missing dependencies.
02	disable-root-login	Failed	Fixed (sed replacement)	Passed	
03	disable-password-auth	Failed	Fixed (sed replacement)	*Failed*	Regex matching issue; file output returned 2 matches instead of 1.
04	max-auth-tries	Failed	Fixed (sed replacement)	Passed	
05	use-pubkey-auth	Failed	Fixed (sed replacement)	Passed	
06	disable-empty-passwords	Failed	Fixed (sed replacement)	Passed	
07	enable-pam	Failed	*Already Passed*	Passed	
08	idle-timeout	Failed	Fixed (sed replacement)	Passed	

Check ID	Control Name	Initial Audit	Hardener Action (Fix Mode)	Post-Fix Audit	Analytical Notes
09	disable-x11-forwarding	Failed	Fixed (sed replacement)	*Failed*	Regex matching issue; file output returned 2 matches instead of 1.
10	ssh-ciphers-fips-compliant	Failed	Fixed (sed replacement)	Passed	
11	strong-macs	Failed	Fixed (sed replacement)	Passed	
12	strong-kex	Failed	Fixed (sed replacement)	Passed	
13	secure-x11-proxy	Failed	Fixed (sed replacement)	Passed	

Kernel Hardening

Check ID	Control Name	Initial Audit	Hardener Action (Fix Mode)	Post-Fix Audit	Analytical Notes
14	kernel-01-dmesg-restrict	Passed	*Already Passed*	Passed	
15	kernel-02-randomize-va-space	Passed	*Already Passed*	Passed	
16	kernel-03-disable-core-dumps	Failed	Fixed (sysctl write)	*Failed*	Disk modified, but live state requires reboot/reload.
17	kernel-04-yama-ptrace-scope	Passed	*Already Passed*	Passed	

Check ID	Control Name	Initial Audit	Hardener Action (Fix Mode)	Post-Fix Audit	Analytical Notes
18	kernel-05-kptr-restrict	Failed	Fixed (sysctl write)	*Failed*	Disk modified, but live state requires reboot/reload.
19	kernel-06-ebpf-unprivileged	Failed	Fixed (sysctl write)	*Failed*	Disk modified, but live state requires reboot/reload.
20	kernel-07-ebpf-jit-hardening	Failed	Fixed (sysctl write)	*Failed*	Disk modified, but live state requires reboot/reload.
21	kernel-08-userns-clone	Failed	Fixed (sysctl write)	*Failed*	Disk modified, but live state requires reboot/reload.
22	kernel-09-protected-fifos	Failed	Fixed (sysctl write)	*Failed*	Disk modified, but live state requires reboot/reload.
23	mod-01-blacklist-dccp	Failed	Fixed (file write)	Passed	
24	mod-02-blacklist-sctp	Failed	Fixed (file write)	Passed	
25	net-01-syn-cookies	Passed	*Already Passed*	Passed	
26	net-02-rp-filter	Failed	Fixed (sysctl write)	*Failed*	Disk modified, but live state requires reboot/reload.

Check ID	Control Name	Initial Audit	Hardener Action (Fix Mode)	Post-Fix Audit	Analytical Notes
27	net-03-accept-redirects	Failed	Fixed (sysctl write)	*Failed*	Disk modified, but live state requires reboot/reload.
28	net-04-source-route	Passed	*Already Passed*	Passed	
29	net-05-bogus-icmp	Passed	*Already Passed*	Passed	
30	net-06-ipv4-ip_forward	Passed	*Already Passed*	Passed	
31	net-07-ipv4-send_redirects	Failed	Fixed (sysctl write)	Passed	Reload successful.
32	mod-01-disable-usb-storage	Failed	Fixed (file write)	Passed	
33	crypto-01-fips-mode	Failed	Manual Intervention	Failed	Correctly bypassed unsafe boot-loader/crypto modification.

Filesystem & Authentication Hardening

Check ID	Control Name	Initial Audit	Hardener Action (Fix Mode)	Post-Fix Audit	Analytical Notes
34	fs-01-shadow-perms	Passed	*Already Passed*	Passed	
35	fs-02-passwd-perms	Passed	*Already Passed*	Passed	

Check ID	Control Name	Initial Audit	Hardener Action (Fix Mode)	Post-Fix Audit	Analytical Notes
36	fs-03-group-perms	Passed	*Already Passed*	Passed	
37	fs-04-bootloader-password	Failed	Manual Intervention	Failed	Correctly bypassed unsafe bootloader modification.
38	fs-05-sticky-bit	Passed	*Already Passed*	Passed	
39	auth-01-pass-encryption	Failed	Fixed (sed replacement)	Passed	
40	auth-02-pass-complexity	Failed	Fixed (file write)	Passed	
41	auth-03-pass-lifetime	Failed	Fixed (sed replacement)	Passed	
42	auth-04-cached-auth	Failed	Manual Intervention	Failed	Bypassed SSSD AD modification to prevent lockouts.
43	auth-05-root-lock	Passed	*Already Passed*	Passed	
44	auth-06-umask-strict	Failed	Fixed (sed replacement)	Passed	
45	fs-07-find-suid	Failed	Fixed (chmod u-s)	*Failed*	Fix applied `chmod`, but post-audit command syntax exited with error.

Service Hardening (Auditd & AIDE)

Check ID	Control Name	Initial Audit	Hardener Action (Fix Mode)	Post-Fix Audit	Analytical Notes
46	aide-package-installed	Failed	Error (File not found)	Failed	Safety mechanism aborted fix due to missing package list path.
47	aide-conf-audit-binaries	Failed	Error (File not found)	Failed	Safety mechanism aborted fix due to missing `/etc/aide/aide.conf`.
48	auditd-package-installed	Failed	Error (File not found)	Failed	Safety mechanism aborted fix due to missing package list.
49	auditd-service-enabled	Failed	Error (File not found)	Failed	Safety mechanism aborted fix due to missing service config.
50	auditd-service-active	Passed	*Already Passed*	Passed	
51	audit-identity-file-changes	Failed	Manual Intervention	Failed	Complex rule logic correctly flagged for manual implementation.

Check ID	Control Name	Initial Audit	Hardener Action (Fix Mode)	Post-Fix Audit	Analytical Notes
52	audit-privilege-executables	Failed	Manual Intervention	Failed	Complex rule logic correctly flagged for manual implementation.
53	audit-file-attribute-changes	Failed	Error (File not found)	Failed	Target rule file `/etc/audit/rules.d/stig.rules` did not exist.
54	audit-file-deletion	Failed	Manual Intervention	Failed	Complex rule logic correctly flagged for manual implementation.
55	audit-kernel-module-loading	Failed	Manual Intervention	Failed	Complex rule logic correctly flagged for manual implementation.
56	audit-immutable-rule-config	Failed	Error (File not found)	Failed	Target configuration file did not exist.
57	audit-immutable-mode-runtime	Failed	Manual Intervention	Failed	Requires manual system reboot.