anoBytes | TECHNOLOGY SOLUTIONS

Harden Linux against CIS Benchmarks with OpenSCAP

A repeatable workflow for Fedora, RHEL, and Ubuntu: measure a running server against a CIS Benchmark, remediate only what failed, and keep the before-and-after evidence.

OPENSCAP · SCAP SECURITY GUIDE · CIS BENCHMARKS · HOW-TO GUIDE

General educational guidance. Not affiliated with or endorsed by the Center for Internet Security. Test every command on a non-production system or a snapshot first. Profile names and rule details refer to ComplianceAsCode v0.1.81. Published 2026-08-05.

3 Linux platforms

Fedora 44 · RHEL 9 and 10 · Ubuntu Server 24.04 LTS

7 steps, install to re-scan

run it again later to confirm the system still passes

TOOLS

The three pieces you are running

The scanner, the rule content, and the profile are three separate things.

OpenSCAP (`oscap`), the scanner

It evaluates a system against SCAP content and produces results and a report. It ships in your distribution's repositories as `openscap-scanner`.

ComplianceAsCode / content, the rules

The open source project that writes the checks. Its built release archive is still named **SCAP Security Guide** (SSG) for historical reasons. Each release ships one **SCAP source data stream** per operating system, named `ssg-<product>-ds.xml` (for example `ssg-ubuntu2404-ds.xml`), containing every profile for that OS.

A CIS profile, inside the data stream

A profile selects and configures the rules that correspond to one CIS Benchmark level, for example "Level 1 - Server".

Alignment vs. certification

WORTH GETTING RIGHT IN CLIENT CONVERSATIONS

These profiles are community-maintained content that **aligns to** a specific CIS Benchmark and carries CIS trademark attribution. The project does not claim CIS certification, and neither does Red Hat. CIS's own certified assessor is CIS-CAT Pro. A clean SSG scan is strong evidence of alignment; it is not a CIS certificate. The alignment is tightest for RHEL and Ubuntu, whose profiles track benchmark versions released in the public CIS catalog. For Fedora it is looser: the Fedora profile tracks a CIS Community draft (the "CIS Fedora 40 Branch"), which is upstream working material rather than a released catalog benchmark.

MAP Distribution cheat-sheet

Where the scanner and content come from, the data stream to scan, and the benchmark each profile aligns to, per platform.

	FEDORA 44	RHEL 9 / 10	UBUNTU SERVER 24.04
Scanner install	`sudo dnf install openscap-scanner`	`sudo dnf install openscap-scanner`	`sudo apt install openscap-scanner`
Content source	`sudo dnf install scap-security-guide (current 0.1.81)`	`sudo dnf install scap-security-guide (Red Hat-supported)`	`upstream release zip (see Step 1)`
Data stream	`/usr/share/xml/scap/ssg/content/ssg-fedora-ds.xml`	`.../ssg-rhel9-ds.xml or ssg-rhel10-ds.xml`	`scap-security-guide-0.1.81/ssg-ubuntu2404-ds.xml`
CIS server profiles	`cis_server_l1, cis (L2, DRAFT)`	`cis_server_l1, cis (= L2 Server)`	`cis_level1_server, cis_level2_server`
Aligned benchmark	CIS Fedora 40 Branch (a CIS Community draft, not a released catalog benchmark)	RHEL 9 Benchmark v2.0.0 / RHEL 10 Benchmark v1.0.1 (what upstream 0.1.81 targets; dnf-shipped content follows your minor release, so confirm the version in the profile description)	Ubuntu 24.04 LTS Benchmark v1.0.0 (the version the v0.1.81 content targets; CIS has since published v2.0.0)

Scope limits and caveats:

Arch-based distributions are not covered

ComplianceAsCode ships SCAP content for dozens of operating systems, but none for Arch Linux, and the scanner itself is only in the AUR. There is no supported CIS SCAP content to scan against.

Ubuntu 26.04 LTS has no OS-matched content yet

No `ubuntu2604` data stream exists in v0.1.81. The newest Ubuntu content is `ubuntu2404`, so this guide scans 24.04. Scanning 26.04 with 24.04 content is an unsupported combination and produces misleading results.

On RHEL, use the dnf-shipped content, not the upstream zip

Red Hat officially supports the SCAP Security Guide versions shipped with each minor release (Red Hat KB 6337261) and warns that content is not always backward compatible. The upstream zip is for distributions whose packaged content is missing or stale, which is exactly Ubuntu's situation.

PREP Prerequisites

- A test system or a snapshot first. Remediation rewrites system configuration and there is no automated undo (Step 6 covers this).
- Root access via `sudo`, and a working **non-root administrative account**. The CIS profiles disable SSH root login, so if you administer the box as root over SSH you will lose that path when `sshd` reloads.
- About 2.5 GB of free space on Ubuntu for the content archive and its extracted contents: the download is about 175 MB, and it extracts to over 2 GB.

01 Install the scanner and the content

Fedora 44 ships the current upstream content in its own repositories:

```
sudo dnf install openscap-scanner scap-security-guide
ls /usr/share/xml/scap/ssg/content/ | grep fedora
```

RHEL 9 and 10 use the same packages, with Red Hat-supported content:

```
sudo dnf install openscap-scanner scap-security-guide
ls /usr/share/xml/scap/ssg/content/ | grep rhel
```

Ubuntu 24.04 needs the scanner from the archive but the content from upstream. The archive's `ssg-*` packages are stuck at 0.1.71 (December 2023), and that build predates the 24.04 product entirely: it contains no `ssg-ubuntu2404-ds.xml`, so the packaged content cannot scan the OS it ships on. Download the release archive instead, and verify its checksum:

```
sudo apt update && sudo apt install openscap-scanner unzip
wget https://github.com/ComplianceAsCode/content/releases/download/v0.1.81/scap-security-guide-0.1.81.zip
wget https://github.com/ComplianceAsCode/content/releases/download/v0.1.81/scap-security-guide-0.1.81.zip.sha512
sha512sum -c scap-security-guide-0.1.81.zip.sha512
unzip scap-security-guide-0.1.81.zip
cd scap-security-guide-0.1.81
```

The zip extracts to a single `scap-security-guide-0.1.81/` directory with every `ssg-<product>-ds.xml` at the top level, plus prebuilt `bash/` and `ansible/` remediation for whole profiles and human-readable HTML `guides/`. Check the releases page for a newer version before you start; the project releases several times a year, and the commands only need the version string updated.

02 List the profiles and pick one

```
# Ubuntu (inside the extracted directory)
oscap info ssg-ubuntu2404-ds.xml

# Fedora / RHEL
oscap info /usr/share/xml/scap/ssg/content/ssg-rhel9-ds.xml
```

The output lists every profile with its title and full ID. The CIS profile IDs follow the pattern `xccdf_org.ssgproject.content_profile_<name>`:

DISTRIBUTION	PROFILE ID SUFFIX	CIS BENCHMARK TARGET
Ubuntu 24.04	`cis_level1_server`	Level 1 - Server
Ubuntu 24.04	`cis_level2_server`	Level 2 - Server
Ubuntu 24.04	`cis_level1_workstation` / `cis_level2_workstation`	Workstation levels
RHEL 9 / 10	`cis_server_l1`	Level 1 - Server
RHEL 9 / 10	`cis`	Level 2 - Server (its ID is the bare `cis`)
RHEL 9 / 10	`cis_workstation_l1` / `cis_workstation_l2`	Workstation levels

DISTRIBUTION	PROFILE ID SUFFIX	CIS BENCHMARK TARGET
Fedora	`cis_server_l1`, `cis`, `cis_workstation_l1`, `cis_workstation_l2`	DRAFT profiles

Which one to pick. Level 1 is the baseline CIS considers practical and prudent, with a clear security benefit and limited breakage risk. Level 2 adds defense in depth for environments where security is paramount, and it can reduce functionality or performance. Start with Level 1 - Server on a server. The Fedora profiles describe themselves as draft, experimental, and maintained best-effort against the CIS Fedora 40 Branch. They are fine for a personal baseline, but do not present a Fedora scan to a client as a CIS Benchmark result. The same data streams also carry non-CIS profiles (DISA STIG, ANSSI, PCI DSS, depending on product), which is useful when a client's framework is something other than CIS.

03 Scan the system

Run the evaluation as root so every check can read the files it needs; an unprivileged scan degrades into `error` and `unknown` results for root-only checks. One command produces both machine-readable results and an HTML report:

```
# Ubuntu 24.04, CIS Level 1 Server
sudo oscap xccdf eval \
  --profile xccdf_org.ssgproject.content_profile_cis_level1_server \
  --results results.xml \
  --report report.html \
  ssg-ubuntu2404-ds.xml

# RHEL 9 example (the suffix shorthand also works: --profile cis_server_l1)
sudo oscap xccdf eval \
  --profile cis_server_l1 \
  --results results.xml \
  --report report.html \
  /usr/share/xml/scap/ssg/content/ssg-rhel9-ds.xml
```

The exit code is expected to be non-zero. `oscap` returns **0** only when every rule passes, **2** when at least one rule failed or came back unknown, and **1** on an actual evaluation error. On a first scan, 2 is the normal outcome; in scripts or CI, treat it as "scan completed with findings", not as a command failure.

04 Read the report

Open `report.html` in a browser (`scp` it off a headless server). It shows a compliance score, a rule-by-rule pass/fail table, and for each rule the rationale, the exact check, and the remediation it would apply. Three result types deserve attention beyond pass and fail:

`notapplicable`
The rule does not apply to this system (wrong platform, package absent). Normal.

`notchecked`
The rule has no automated check and needs manual review.

`error`
The check could not run, most often a scan executed without root.

Skim the failed rules before remediating anything. This is where you learn what the profile is about to change, and where you decide which rules you do not want (Step 6 shows how to skip them).

05 Generate remediation for what failed

You can generate fixes two ways:

- **From the profile (`--profile` against the data stream)**
  The script tries to fix **every rule in the profile**, whether or not your system already passes it. This is the "bring a fresh build to baseline" option, and it is what the prebuilt scripts in the release's `bash/` directory are.

- **From your scan results (`--result-id` against `results.xml`)**
  The script fixes **only the rules that failed on this system**. That makes it shorter and easier to review, and it is the right default on an existing server.

Generate the results-based script:

```
# Find the TestResult ID inside your results file
oscap info results.xml
# It follows the pattern xccdf_org.open-scap_testresult_<profile-id>

oscap xccdf generate fix \
  --fix-type bash \
  --result-id xccdf_org.open-scap_testresult_xccdf_org.ssgproject.content_profile_cis_level1_server \
  results.xml > cis-remediation.sh
```

For the whole-profile variant, swap in `--profile cis_level1_server ssg-ubuntu2404-ds.xml` in place of the `--result-id` and results pair.

Some rules have no automated fix

`EXPECT: FIX FOR THIS RULE '...' IS MISSING!`

Repartitioning (`partition_for_tmp` and related rules) and the GRUB bootloader password (`grub2_password`) are the classic examples, since no script should repartition a live disk or invent a password for you. Those rules stay failed until you address them by hand. On a cloud image with a single root volume, the partition rules may be a documented accepted risk rather than a fix.

06 Review before you apply

Read the generated script before running it. It is plain bash, one commented block per rule. There is no automated way back: Red Hat states that it provides no automated method to revert security-hardening remediations (KB 7032665), so your only rollback is the snapshot you take before you apply. The risky changes to check for, all confirmed against the v0.1.81 Ubuntu CIS Level 1 Server profile:

- **The fix can remove your firewall tool**
  CIS requires a single firewall utility. The CIS profile sets its firewall variable (`var_network_filtering_service`) to **nftables**, and the generated fix for `package_ufw_removed` runs `apt-get remove -y ufw` when the variable is not set to ufw. If your ruleset lives in ufw (as on a stock hardened Ubuntu), either re-express it in nftables first or skip those rules.

- **SSH root login goes away and weak crypto is disabled**
  `sshd_disable_root_login` plus the strong cipher, KEX, and MAC lists take effect when `sshd` reloads. Key-based auth for normal users is untouched (no CIS rule disables `PubkeyAuthentication`), but log in with your non-root account and keep one working session open until you have verified a fresh login.

- **PAM is rewritten through pam-auth-update on Ubuntu**

  The fixes install `pam-configs` profiles (password quality, faillock, and related) and apply them noninteractively. If conflicting profiles are selected you can see `Incompatible PAM profiles selected` messages from `pam-auth-update` itself. After remediation, verify you can still `sudo` and log in before closing your root session.

- **Skipping a rule is legitimate**

  Delete the rule's block from the script, or build a tailored profile: `autotailor` (in `openscap-utils`) writes a tailoring file you pass to `oscap` with `--tailoring-file`, which is the auditable way to record "we deviate from 4.3.2 because...". The SCAP Workbench GUI does the same job, though it is not currently packaged for Fedora 44.

Then apply:

```
# 1. Snapshot or image the machine (cloud snapshot, VM snapshot, or backup)
# 2. Keep a root shell open in a second terminal until you are done
# 3. Apply
chmod +x cis-remediation.sh
sudo bash ./cis-remediation.sh 2>&1 | tee remediation.log
```

The script must run as root and only on the OS it was generated for. Most rules carry a platform gate that prints `Remediation is not applicable, nothing was done` on a mismatch, but a sizable minority carry no gate at all (in the v0.1.81 Ubuntu Level 1 script, roughly a quarter of the fix blocks) and would run wherever the script is executed, so treat the gate as a convenience rather than a guarantee. There is also `sudo oscap xccdf eval --remediate ...`, which fixes failures during the scan itself in one shot. Red Hat's documentation warns that, used carelessly, it can leave a system non-functional, so on a live server the generate-review-apply loop above is the defensible process. Keep `--remediate` for disposable builds.

07 Reboot and re-scan

Some fixes only take effect after a reboot: kernel module blacklists (`kernel_module_*_disabled` rules are flagged `Reboot: true`), GRUB command-line changes such as enabling AppArmor, and anything read once at boot, so re-scan only after the machine has come back up:

```
sudo reboot
# log back in with your non-root account, then return to the extracted directory:
cd scap-security-guide-0.1.81
sudo oscap xccdf eval \
  --profile xccdf_org.ssgproject.content_profile_cis_level1_server \
  --results results-after.xml \
  --report report-after.html \
  ssg-ubuntu2404-ds.xml
```

Compare `report-after.html` with the first report. Remaining failures fall into three buckets: rules with no automated fix (Step 5), rules you deliberately skipped or tailored out (document them), and genuine leftovers to fix by hand. Keep both reports. A before-and-after pair is the evidence an auditor or a client wants to see.

ANSIBLE Optional: the Ansible route

For fleets, drift correction, or anywhere you already run configuration management, generate a playbook instead of a shell script:

```
oscap xccdf generate fix \
  --fix-type ansible \
  --profile cis_level1_server \
  ssg-ubuntu2404-ds.xml > cis-playbook.yml

# Dry run first, then apply
ansible-playbook -i "localhost," -c local --check cis-playbook.yml
sudo ansible-playbook -i "localhost," -c local cis-playbook.yml
```

The playbooks are idempotent, which makes them safer to re-run than the bash script. Two requirements: Ansible 2.9 or newer, and the `community.general` and `ansible.posix` collections, so install the full `ansible` package (or add those two collections to `ansible-core` with `ansible-galaxy`). Ansible coverage of rules can lag bash coverage, so a bash-remediated system and an Ansible-remediated system may not land on identical scores.

HELP Troubleshooting

- **`oscap` exited with code 2**
  The scan worked and found failures. Only exit code 1 is an error.

- **Many `error` results**
  You ran the scan without `sudo`. Re-run as root.

- **Ubuntu's packaged `ssg-*` content has no 24.04 data stream**
  Correct, and why Step 1 downloads the upstream zip. The archive packages are at 0.1.71, which predates the `ubuntu2404` product.

- **The report shows `notchecked` for rules I care about**
  Those rules ship no automated check. The HTML guide for your profile (in the release `guides/` directory) describes the manual verification.

- **A remediated rule still fails after re-scan**
  Check whether it is flagged `Reboot: true` in the report and whether you rebooted, then check `remediation.log` for its block, which may have printed a missing-fix or not-applicable message.

- **I locked myself out of SSH**
  Console access, or your still-open root session, is the way back in: re-enable what you need in `/etc/ssh/sshd_config.d/`, or restore the snapshot. This is why Step 6 says to keep a session open and verify a fresh login before disconnecting.

ROLLBACK How to roll back

There is no automated revert. The supported path back is the snapshot or image you took in Step 6. For a single unwanted change, the remediation script itself tells you what it did: find the rule's block in `cis-remediation.sh` or `remediation.log`, see which file it edited or package it removed, and reverse that one change by hand. For example,

run `sudo apt install ufw` and re-enable it if you decide to keep ufw as your firewall, then set the profile's firewall variable accordingly in a tailoring file so the next scan agrees.

DONE What you achieved

- Your server is measured against a named, versioned CIS Benchmark (Ubuntu 24.04 v1.0.0, RHEL 9 v2.0.0, or RHEL 10 v1.0.1, as your content version targets) with a scanner and content you can pin and re-run.
- You remediated failures with generated code that you reviewed rather than by hand, and skipped and documented the exceptions in a tailoring file.
- You hold before-and-after reports as audit evidence, and a repeatable loop (scan, remediate, reboot, re-scan) for drift checks.

REF References

ComplianceAsCode / content (the SCAP Security Guide source) · `github.com/ComplianceAsCode/content`

ComplianceAsCode release v0.1.81 · `github.com/ComplianceAsCode/content/releases/tag/v0.1.81`

OpenSCAP project and user manual · `open-scap.org` · `static.open-scap.org/openscap-1.3/oscap_user_manual.html`

Red Hat: Supported versions of the SCAP Security Guide in RHEL (KB 6337261); why remediations cannot be automatically reverted (KB 7032665) · `access.redhat.com`

CIS Benchmarks · `cisecurity.org/cis-benchmarks`