

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.

[OPENS CAP](#) · [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 v0.1.82. Published 2026-09-06.

4 Linux versions

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

8 steps, install to re-scan

run it again later to confirm the system still passes

TOOLS The three components 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 is in your distribution's repositories as `openscap-scanner`. Fedora and RHEL 10 carry 1.4.4, and RHEL 9 carries 1.3.14. Ubuntu 24.04 has 1.3.9 from `universe`, which is community-maintained rather than covered by Canonical's standard support.

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 carries 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 is not certification

WHAT A CLEAN SCAN PROVES

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-CAT Pro is CIS's own assessment tool, and CIS certifies other vendors' products through its SecureSuite program; OpenSCAP with SSG content is neither, and holds no CIS certification.

A clean SSG scan is strong evidence of alignment. It is not a CIS certificate. Alignment is tightest for RHEL and Ubuntu, whose profiles track benchmark versions released in the public CIS catalog. The Fedora profile tracks a

CIS Community draft (the "CIS Fedora 40 Branch"), which is upstream working material rather than a released catalog benchmark.

Your score will not match CIS-CAT

SSG CHECKS ARE THE STRICTER SET

SSG writes one set of checks for both its STIG and CIS profiles, so they come out stricter than the benchmark's own: SSG wants a kernel module disabled in a configuration file where CIS accepts the module being absent, and a service masked where CIS accepts disabled. It reads `/etc/fstab` where CIS checks the running mount, and returns `notapplicable` for the desktop rules on a server where CIS returns pass (Red Hat KB 7139696). So when you hand a report to a client, say whether SSG or CIS-CAT produced it.

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	<code>sudo dnf install openscap-scanner openscap-utils</code>	<code>sudo dnf install openscap-scanner openscap-utils</code>	<code>sudo apt install openscap-scanner openscap-utils</code>
Content source	<code>sudo dnf install scap-security-guide</code> (check what dnf offers; Fedora 44 carried 0.1.81 when this was written)	<code>sudo dnf install scap-security-guide</code> (Red Hat-supported; RHEL 9 and 10 carried 0.1.81 when this was written)	upstream release zip (see Step 1)
Data stream	<code>/usr/share/xml/scap/ssg/content/ssg-fedora-ds.xml</code>	<code>/usr/share/xml/scap/ssg/content/ssg-rhel9-ds.xml</code> or <code>ssg-rhel10-ds.xml</code>	<code>scap-security-guide-0.1.82/ ssg-ubuntu2404-ds.xml</code>
CIS server profiles	<code>cis_server_l1</code> , <code>cis (= L2)</code> ; both DRAFT	<code>cis_server_l1</code> , <code>cis (= L2 Server)</code>	<code>cis_level1_server</code> , <code>cis_level2_server</code>
Aligned benchmark	CIS Fedora 40 Branch, a CIS Community draft	RHEL 9 Benchmark v2.0.0 (CIS has since published v3.0.0) / RHEL 10 Benchmark v1.0.1 (what upstream 0.1.82 targets; content from dnf follows your minor release, so confirm the version in the profile description)	Ubuntu 24.04 LTS Benchmark v1.0.0 (the version the v0.1.82 content targets; CIS has since published v2.0.0)

Limits, and what can go wrong:

Arch-based distributions are not covered

ComplianceAsCode publishes 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 scanner content yet

CIS published the Ubuntu 26.04 LTS Benchmark v1.0.0 on 26 August 2026, and no SCAP content implements it. v0.1.82 added a `ubuntu2604` product whose data stream carries no usable profile: `oscap info ssg-ubuntu2604-ds.xml` lists none, and the file is a 116 KB stub against 12 MB for `ubuntu2404` . There is nothing to scan a 26.04

server against. The newest Ubuntu CIS content is `ubuntu2404`, which this guide uses. Scanning 26.04 with 24.04 content is an unsupported combination and produces misleading results.

On RHEL, take the content from dnf

Red Hat officially supports the SCAP Security Guide versions that come 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).
- A console you can reach without SSH, such as a cloud serial console or the hypervisor's console. You cannot arrange one after you are locked out.
- 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 once `sshd` reloads (Step 7).
- Somewhere to put your firewall rules. On Ubuntu the CIS profile switches the host from `ufw` to `nftables` and can leave it with nothing filtering, so read Step 6 before you apply anything.
- About 2.5 GB of free space on Ubuntu for the content archive and its extracted contents: the download is about 183 MB, and it extracts to over 2 GB.

01 Install the scanner and the content

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

```
sudo dnf install openscap-scanner openscap-utils 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 openscap-utils 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), which predates the 24.04 product and carries no `ssg-ubuntu2404-ds.xml`. Download the release archive instead, and check it against its published SHA-512:

```
sudo apt update && sudo apt install openscap-scanner openscap-utils unzip
wget https://github.com/ComplianceAsCode/content/releases/download/v0.1.82/scap-security-guide-0.1.82.zip
wget \
  https://github.com/ComplianceAsCode/content/releases/download/v0.1.82/scap-security-guide-0.1.82.zip.sha512
sha512sum -c scap-security-guide-0.1.82.zip.sha512
unzip scap-security-guide-0.1.82.zip
cd scap-security-guide-0.1.82
```

The zip extracts to a single `scap-security-guide-0.1.82/` 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/`. The archive and its checksum come from the same release, so `sha512sum -c` catches a truncated or corrupted download rather than a substituted one; the project publishes no detached signature.

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	<code>cis_level1_server</code>	Level 1 - Server
Ubuntu 24.04	<code>cis_level2_server</code>	Level 2 - Server
Ubuntu 24.04	<code>cis_level1_workstation</code> / <code>cis_level2_workstation</code>	Workstation levels
RHEL 9 / 10	<code>cis_server_l1</code>	Level 1 - Server
RHEL 9 / 10	<code>cis</code>	Level 2 - Server (its ID is the bare <code>cis</code>)
RHEL 9 / 10	<code>cis_workstation_l1</code> / <code>cis_workstation_l2</code>	Workstation levels
Fedora	<code>cis_server_l1</code> , <code>cis</code> , <code>cis_workstation_l1</code> , <code>cis_workstation_l2</code>	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 returns `error` and `unknown` 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
```

Expect a non-zero exit code. `oscap` returns **2** when at least one rule came back fail, unknown, or error, **1** when the command itself failed, and **0** when none of the three appeared. 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. Besides pass and fail, four results come up often:

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.

unknown

The check ran but could not reach a verdict.

error

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

Rules outside the profile you selected show as `notselected`, so ignore those. 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. 35 of the 408 blocks in the v0.1.82 Ubuntu Level 1 Server script print that message, three of them firewall rules (`nftables_ensure_default_deny_policy` , `set_ufw_default_rule` , `firewall_single_service_active`), so the default policy is yours to write whichever utility you keep.

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 undo: 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 v0.1.82 Ubuntu CIS Level 1 Server profile:

• The fix removes ufw and puts nothing in its place

CIS requires a single firewall utility, and the CIS profile picks `nftables` (`var_network_filtering_service`). The fix for `package_ufw_removed` runs an `apt-get remove` on ufw. The fix that would install nftables, `package_nftables_installed` , is skipped while ufw is still running, and `nftables_ensure_default_deny_policy` has no automated fix at all. On a stock ufw-protected server, one pass takes your firewall away and leaves the host unfiltered.

Before you apply anything, either:

- write your ruleset in nftables and load it; or
- delete the `package_ufw_removed` block from the script and set the firewall variable to ufw in a tailoring file.

Run `nft list ruleset` after the run and again after the reboot.

The single-utility rule and the 4.3.2 ufw removal both belong to the v1.0.0 benchmark this content targets. CIS Ubuntu 24.04 v2.0.0 dropped the nftables and iptables sections and requires ufw at 4.1.1, which is the number the single-utility rule carried in v1.0.0, so a host that keeps ufw now matches the current benchmark rather than deviating from it.

• IP forwarding is turned off

The profile sets `net.ipv4.ip_forward` to 0 (`sysctl_net_ipv4_ip_forward`) with no variable to change it, and sets `net.ipv6.conf.all.forwarding` to 0 (`sysctl_net_ipv6_conf_all_forwarding`) through a variable you can move to

1 (`sysctl_net_ipv6_conf_all_forwarding_value=enabled`). A machine that routes packets stops routing: a WireGuard or IPsec gateway, a NAT router, a Kubernetes node, or a Docker host running the nftables backend, which will not switch forwarding back on and reports an error instead.

Check `sysctl net.ipv4.ip_forward` before you apply. Tailor the IPv4 rule out if the machine forwards, and set the IPv6 variable rather than dropping that rule, so the check stays in force. Many cloud images need forwarding for their own networking, which CIS notes as well.

CIS moved this recommendation to Level 2 - Server in the Ubuntu 24.04 Benchmark v2.0.0 and marked it not applicable to a router, so a routing host that skips it is following the current benchmark.

- **SSH root login goes away and weak crypto is disabled**

`sshd_disable_root_login` plus the strong cipher, KEX, and MAC lists go into `/etc/ssh/sshd_config.d/00-complianceascode-hardening.conf`. Nothing in the script reloads `sshd`, so your connection keeps working until you reload it yourself or reboot. Key-based auth for normal users is untouched (no CIS rule disables `PubkeyAuthentication`). Log in with your non-root account, keep that session open, and use Step 7 to reload and test before Step 8 reboots the machine.

- **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.

- **su stops working for everyone**

The profile points `pam_wheel` at a group called `sugroup` (`use_pam_wheel_group_for_su`) and creates that group empty (`ensure_pam_wheel_group_empty`), so `su` to root fails for every account until you add someone to it. If you administer the box with `su -` rather than `sudo`, add your account to `sugroup` before you disconnect.

- **The AIDE database build takes a long time**

`aide_build_database` installs AIDE and runs `aideinit -y -f`, which reads every file it is configured to watch. On a server with real data the run can take tens of minutes and looks stalled while it works. `-f` overwrites an existing database, so a host that already runs AIDE loses its current baseline. Save that database first, or tailor the rule out and initialize AIDE on your own schedule.

- **Three smaller changes that break other things**

- `sudo_add_use_pty` makes sudo demand a terminal, which breaks `ssh host sudo command` and any automation that runs sudo without one.
- The default umask moves to 027 (`var_accounts_user_umask`), so a new file is created `rw-r-----` where it used to be `rw-r--r--` and a new directory `drwxr-x---` where it used to be `drwxr-xr-x`. Anything reading those files from outside the owning group stops working.
- `autofs`, `avahi-daemon`, and `cups` are removed, which takes out automounting, mDNS service discovery including `.local` name lookups, and printing on a host that does any of them.

- **Skipping a rule is legitimate**

Delete the rule's block from the script, or build a tailored profile and generate the script again from it:

`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, the ufw removal, because...". The SCAP Workbench GUI does the same job, though it is not currently packaged for Fedora 44.

```
# Record a deviation instead of deleting the block
autotailor --var-value var_network_filtering_service=ufw \
  --output cis-tailoring.xml \
  ssg-ubuntu2404-ds.xml cis_level1_server

# Then generate the script again from the tailored profile
oscap xccdf generate fix --fix-type bash \
  --tailoring-file cis-tailoring.xml \
  --profile xccdf_org.ssgproject.content_profile_cis_level1_server_customized \
  ssg-ubuntu2404-ds.xml > cis-remediation.sh
```

That writes a new profile whose ID is the base ID with `_customized` appended, so every later scan and every later `generate fix` needs both `--tailoring-file` and the customized ID. A scan that leaves out `--tailoring-file` measures the untailored profile, and the report will not match what you agreed to run. The tailoring file has no effect on a script you generated before it existed.

Then apply:

```
# 1. Snapshot or image the machine (cloud snapshot, VM snapshot, or backup)
# 2. Keep a second session open as your non-root administrative account
# 3. Apply
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, and nothing in the script enforces that. About three fifths of the fix blocks (251 of 408 in the v0.1.82 Ubuntu Level 1 script) are wrapped in a condition that prints `Remediation is not applicable, nothing was done` when it fails, but that condition tests whether a package is installed or a service is running, never which distribution you are on: the wrapper on the `ufw` removal only checks that the `linux-base` package is installed, which holds on any normal Debian or Ubuntu system. Run the script on the machine you scanned, and nowhere else.

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 Test SSH before you reboot

The script wrote the SSH settings to a drop-in file and left `sshd` running on the old configuration, so nothing about your connection has changed yet. Apply it deliberately, while the session you kept open is still there:

```
sudo sshd -t
sudo systemctl reload sshd

# then, from a second terminal:
ssh youruser@yourserver
```

`sshd -t` reports a configuration error without applying anything, so run it first: silence means the file parses. Reload, open the second connection, and confirm it works before you close the first one. The unit is `sshd` on Fedora and RHEL, and on Ubuntu it is `ssh`, which also answers to `sshd`.

The profile also sets `ClientAliveInterval 300` and `ClientAliveCountMax 3`, so `sshd` probes a connection that has gone quiet every five minutes and closes it after three unanswered probes, about fifteen minutes. It does not log out a user who is only idle, because the SSH client answers those probes by itself.

Check `sudo` the same way, and `su -` if you use it.

08 Reboot and re-scan

Some fixes only take effect after a reboot. The kernel module blacklists are flagged `Reboot: true` in the report, but the flag under-reports: `grub2_enable_apparmor` edits the kernel command line and carries no flag at all. Reboot after any remediation run rather than trusting the flag to tell you when you need one, and 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-0.1.82
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

# If you built a tailoring file, scan the tailored profile instead:
#   --tailoring-file cis-tailoring.xml \
#   --profile xccdf_org.ssgproject.content_profile_cis_level1_server_customized \
```

Compare `report-after.html` with the first report. Remaining failures fall into three kinds:

- rules with no automated fix (Step 5);
- rules you deliberately skipped or tailored out (document them); and
- genuine leftovers to fix by hand.

Keep `results.xml` and `results-after.xml` as well as the two HTML reports: the XML is the machine-readable record, you can re-render a report from it or diff the two, and it names the profile and the content version that produced it. Add `--results-arf results-arf.xml` if you want the full ARF bundle with the raw check data, and record the `oscap --version` output alongside them so the run can be reproduced.

A client or an auditor may ask for the before-and-after pair. These files describe the host in detail: installed versions, permissions, accounts, sshd settings, and every check that failed. Store them where the system's own configuration is stored, and send them to a client the way you would send any other finding.

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
```

Then dry-run it and apply it:

```
sudo ansible-playbook -i "localhost," -c local --check cis-playbook.yml
sudo ansible-playbook -i "localhost," -c local cis-playbook.yml
```

`--check` skips every task that shells out, which is 46 of the playbook's 220 command and shell tasks, among them the eleven `pam-auth-update` calls, the AIDE database build, the nftables table creation, and the account locking, so read the playbook for those the way Step 6 says to read the bash script.

If you built a tailoring file, add `--tailoring-file` and the customized profile ID to the generate command above, or the playbook carries the rules you tailored out.

The playbooks are idempotent, which makes them safer to re-run than the bash script. ComplianceAsCode declares Ansible 2.9 as its minimum, which is a floor rather than a version worth running, and the generated playbook does not check it, so install a currently supported release. It also needs the `community.general` and `ansible.posix` collections, so install the full `ansible` package, or add those two collections to `ansible-core` with `ansible-galaxy`.

Some rules have a bash remediation but no Ansible one (77 of the 408 in the v0.1.82 Ubuntu Level 1 Server profile, the faillock and password quality rules among them), so the two routes can leave a system with different scores.

HELP Troubleshooting

- **oscap exited with code 2**

The scan ran and at least one rule came back fail, unknown, or error. Exit code 1 means the command itself failed. A scan run without root exits 2 as well, so check the error count before you trust the score.

- **Many error results**

You ran the scan without `sudo`. Re-run as root.

- **The report shows notchecked for rules I care about**

Those rules have 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 you rebooted. The report's `Reboot: true` flag under-reports, so a rule can need one without being marked. Then check `remediation.log` for its block, which may have printed a missing-fix or not-applicable message.

- **I locked myself out of SSH**

Get back in through the console or the session you left open, then edit `/etc/ssh/sshd_config.d/00-complianceascode-hardening.conf` to re-enable what you need, or restore the snapshot. Step 7 exists so you find this before the reboot rather than after it.

- **The server has no firewall after remediation**

Expected on Ubuntu, and the reason Step 6 says to sort the firewall out first. `nft list ruleset` shows what is loaded. Either load an nftables ruleset, or reinstall ufw and set the firewall variable to ufw in a tailoring file.

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, if you decide to keep ufw as your firewall, run `sudo apt install ufw`, load your rules again, enable it, and set the profile's firewall variable to ufw in a tailoring file so the next scan agrees. Reinstalling the package does not bring the old rules back, so have them written down before you start.

DONE What you achieved

- You measured your server 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 recorded any exceptions, in a tailoring file or in your change record.

- You have before-and-after results files and reports as audit evidence, and a loop to re-run (scan, remediate, reload, reboot, re-scan) whenever you want to check for drift.

REF References

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

ComplianceAsCode release v0.1.82 · github.com/ComplianceAsCode/content/releases/tag/v0.1.82

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); differences between CIS benchmarks and SCAP Security Guide rules (KB 7139696) · access.redhat.com

CIS Benchmarks, and the CIS SecureSuite certification programs for product vendors · cisecurity.org/cis-benchmarks · cisecurity.org/cis-securesuite