SELinux Mandatory access control

START HERE

`getenforce`
Enforcing or permissive?

`ls -Z <path>`
Show file labels

`ps -eZ`
Show process domains

`sudo ausearch -m AVC -ts recent`
What was just denied?

`sudo restorecon -Rv <path>`
Reset labels to policy

`sudo semanage fcontext -a -t <type> '<path>(/.*)?'`
Label a custom path for good

`sudo setsebool -P <bool> on`
Turn on a policy switch

`sudo semanage port -a -t <type> -p tcp <port>`
Allow a service on a new port

Process domain, e.g. httpd_t → **wants access to** → **Object** file or port type → **checked by** → **Policy** allow rules + booleans → **no rule** → **Denied** AVC in audit.log

CONCEPTS

Never disable SELinux to make something work. Switch to permissive, reproduce, read the denial, then fix the label, boolean or port.

Reading a context

Part	Example	Matters for
user	`system_u`	confined users
role	`object_r`	rarely
type	`etc_t`	almost every rule
level	`s0`	MLS/MCS, containers

Normal permissions are checked first; SELinux can only deny more, never grant what they forbid.

MODE AND STATUS

`getenforce`
Show the current mode

`sestatus`
Full status: mode, config, policy
Current mode and Mode from config file can differ.

`setenforce 0`
Switch to permissive until reboot
Denials are logged but not enforced. Use it to test, not as a fix.

`setenforce 1`
Switch back to enforcing

`semanage permissive -a <domain>`
Make one domain permissive
E.g. httpd_t: the rest of the system stays enforcing.

`semanage permissive -d <domain>`
Enforce that domain again

`semanage permissive -l`
List permissive domains

`SELINUX=enforcing`
Boot mode in /etc/selinux/config
Or permissive. Keep SELINUXTYPE=targeted; reboot to apply.

`grubby --update-kernel ALL --args enforcing=0`
Boot into permissive via kernel arg

Modes

Mode	Effect
`enforcing`	denies and logs
`permissive`	logs only, allows
`disabled`	no checks, no labels kept

CONTEXTS

`ls -Z <path>`
Show file contexts

`ls -dZ <dir>`
Context of the directory itself

`ps -eZ`
Show the domain of every process
Filter with grep, e.g. ps -eZ | grep httpd.

`id -Z`
Your own user context

`matchpathcon -V <path>`
Check a label against the policy

mv keeps a file's old label; cp creates a new one from the target directory. Run restorecon after moving files into place.

FILE LABELS

`semanage fcontext -a -t <type> '<path>(/.*)?'`
Add a persistent label rule
Changes the policy only; run restorecon to apply it.

`restorecon -Rv <path>`
Relabel files to match the rules

`restorecon -Rnv <path>`
Dry run: show what would change

`chcon -t <type> <file>`
Change a label temporarily
Lost on the next restorecon or relabel; use for tests only.

`semanage fcontext -a -e <src> <dst>`
Label dst like src (equivalence)

`semanage fcontext -l -C`
List your local label rules

`semanage fcontext -l | grep <path>`
Find the rule for a path

`semanage fcontext -d '<path>(/.*)?'`
Delete a local label rule

chcon or semanage?

Tool	After relabel
`chcon`	label lost: tests only
`semanage fcontext`	label kept: the real fix

BOOLEANS

`getsebool -a`
List all booleans and values
Filter with grep, e.g. getsebool -a | grep httpd.

BOOLEANS (cont.)

`getsebool <bool>`
Show one boolean

`setsebool -P <bool> on`
Set a boolean, persistently
Without -P the value resets at reboot.

`setsebool <bool> on`
Set a boolean until reboot

`semanage boolean -l`
Booleans with descriptions

`semanage boolean -l -C`
Booleans you changed locally

Common switches: httpd_can_network_connect, httpd_can_network_connect_db, httpd_use_nfs, httpd_enable_homedirs.

PORTS

`semanage port -l`
List port types
Filter with grep, e.g. semanage port -l | grep http.

`semanage port -a -t <type> -p tcp <port>`
Allow a service on a port
Example type: http_port_t, ssh_port_t.

`semanage port -m -t <type> -p tcp <port>`
Retype a port defined already

`semanage port -d -t <type> -p tcp <port>`
Remove a local port rule

DENIALS

`ausearch -m AVC -ts recent`
Show denials of the last 10 minutes
Other -ts words: today, boot, this-week.

`ausearch -m AVC -c <comm> -i`
Denials of one program, readable

`ausearch -m AVC -ts recent | audit2why`
Explain why access was denied

`sealert -l "*"`
Analysis with suggested fixes
Needs setroubleshoot-server.

`journalctl -t setroubleshoot`
setroubleshoot alerts in the journal

`semodule -DB`
Rebuild without dontaudit rules
Reveals silenced denials; restore with semodule -B.

DENIALS (cont.)

No AVC but still failing? Look for silenced denials (semodule -DB), then check plain permissions and the firewall.

POLICY MODULES

`ausearch -m AVC -ts recent | audit2allow -M <name>`
Build a module from denials
Read <name>.te first: it may allow far more than needed.

`semodule -i <name>.pp`
Install a policy module

`semodule -l`
List installed modules

`semodule -r <name>`
Remove a module

audit2allow is the last step: first rule out a wrong label, a boolean or a port type.

RELABEL AND USERS

`touch /.autorelabel`
Relabel everything at next boot
Needed after running disabled; the boot takes a while.

`fixfiles -F onboot`
Schedule a forced full relabel

RECIPES

Serve a website from /srv/www

```
sudo semanage fcontext -a -t
httpd_sys_content_t '/srv/
www(/.*)?'
sudo restorecon -Rv /srv/www
ls -Z /srv/www
```

Run a web server on port 8081

```
sudo semanage port -l | grep
http_port_t
sudo semanage port -a -t
http_port_t -p tcp 8081
sudo systemctl restart httpd
```

Track down a denial

```
# reproduce in permissive mode
sudo setenforce 0
sudo ausearch -m AVC -ts recent
| audit2why
# fix: restorecon, setsebool -P
or semanage port
sudo setenforce 1
```

Copy local tweaks to another host

```
sudo semanage export -f
selinux.local
scp selinux.local <host>:
ssh <host> sudo semanage import
-f selinux.local
```