# Audit and Compliance

Audit trail, evidence and periodic reviews · Checked against tacctl 0.2.25

## Contents

1. Where the audit trail lives ..... 2
2. Reading an audit line ..... 3
3. Search recipes ..... 4
4. Evidence by control ..... 5
5. Monthly review checklist ..... 5
6. What is never logged ..... 6

How to answer "who did what, when, and why" from tacctl's audit trail, and how to run the periodic reviews that auditors ask for. Written for superusers, security teams and compliance staff.

> **Example names used on this page** Users: `alice` (superuser), `bob` (engineer), `carol` (read-only), `dave` · Scope: `site-a` · Device: `sw1`

## 1. Where the audit trail lives

tacctl writes every audit line to **syslog**, facility `auth`:

| Tag | Source |
|---|---|
| `tacctl` | Every tacctl command that changes state, and every refusal |
| `tacctl-console` | Login console sessions (start, end, refused commands) |

tacctl keeps no log of its own. On a journald system:

```
journalctl -t tacctl -t tacctl-console                # everything
journalctl -t tacctl --since "2026-10-01" --until "2026-11-01"
```

> **Ship the log off the server.** For example, `systemd-journal-upload` to a central collector, somewhere an operator who runs tacctl cannot rewrite it. That is what turns it into an audit trail.

Separately, `tacctl log …` shows the **authentication service's** logs (who logged in to which device, and accounting of commands run on devices):

```
tacctl log search dave           # logins by a user (TACACS+: last 7 days)
tacctl log failures              # failures in the last 24 hours
tacctl log accounting 100        # device command accounting
```

From 0.2.24 these take filters and `--json`, which makes the evidence repeatable:

```
tacctl log accounting --since 2026-11-01 --until 2026-11-02 --user alice --device sw1
tacctl log accounting --type change --since -24h --json | jq .
tacctl log failures --since yesterday --json
```

- Accounting records name the device that sent them (`"Device":"sw1"`, the registered name when the caller may see it, else the address).
- `--type change` is the records that carry a Junos change-log entry; `command` is any other command; `login` the start and stop of a session.
- `--json` is one object per line; the time is UTC. The filters are described in *Day-to-Day Device Operations* §1.4.

## 2. Reading an audit line

Each command that changes something writes a **start** line, a **detail** line (what changed) and an **end** line, all sharing one `sid`:

```
cmd start v=1 user=alice uid=1001 origin=sudo tier=superuser tty=pts/3 console=-
    cmd="user add" args="dave ops --hash <redacted>" sid=k3x9a0bq
user add name=dave group=ops scopes=core,lab by=alice sid=k3x9a0bq
cmd end v=1 user=alice cmd="user add" result=ok status=0 duration=0.312s
    snapshot=20261021_093015_412 sid=k3x9a0bq
```

(Each is one line in the log. The examples are wrapped here.)

| Field | Meaning |
|---|---|
| `user`, `uid`, `origin` | Who ran it. `origin=root` (no login identity) only for a root shell or script. |
| `tier` | The tier the person was held to |
| `tty`, `console` | Terminal and console session (`-` = none) |
| `cmd`, `args` | What was run. Secrets and hashes show as `<redacted>`. |
| `result`, `status` | `ok`, `failed`, `denied`, `interrupted` or `aborted`, and the exit code |
| `snapshot` | The snapshot taken **before** the change. `tacctl backup diff <id>` shows it. |
| `sid` | Ties all the lines of one command together |

Commands that only read (lists, shows, checks) write no lines. Every refusal is logged at `auth.warning`:

```
tier DENY user=carol tier=readonly cmd=user remove sid=...
cmd DENY v=1 user=bob ... tier=engineer cmd="scope secret"
    args="site-a set <redacted>" reason=superuser sid=...
```

`reason=superuser` means the verb belongs to superusers. `reason=scope` means the target is outside the caller's scopes.

## 3. Search recipes

```
J="journalctl -t tacctl -t tacctl-console --no-pager"

$J | grep 'cmd end'                                # every change made through tacctl
$J | grep 'cmd end' | grep -v 'result=ok'          # changes that failed or were refused
$J | grep 'DENY'                                   # every refusal (tier, verb, console)
$J | grep 'sid=k3x9a0bq'                           # all lines of one command
$J | grep -E '^.*user (add|remove|disable|enable|move|rename|scope)'   # account changes
$J | grep 'user passwd'                            # password resets (never the password)
$J | grep 'user password-change'                   # password changes required or lifted
(state=)
$J | grep 'password-change DENY'                   # commands refused until the password is
changed
$J | grep 'secret-read'                            # every time a secret was shown, and by
whom
$J | grep 'scope secret set'                       # secret rotations
$J | grep -E 'emergency lockdown|breakglass'       # lockdowns and holds, with reasons
$J | grep 'breakglass hold' | grep 'firewall=failed'   # holds whose firewall did not load
$J | grep -E 'backup (keep|unkeep|recover)'        # snapshots kept, dropped or recovered
$J | grep 'user email'                             # email changes (op=set|clear, never the
address)
$J | grep -E 'device (clone|replace)'              # a device copied at a bench address,
then moved in
$J | grep 'device config apply'                    # device changes: review, approval, apply
$J | grep 'scope mgmt-acl scope=.* op=mode'        # Junos filter mode changes
(value=strict|ssh-only)
$J | grep -E 'server (addresses|listen|metrics|name)'   # changes of this server's
settings
$J | grep 'group edit'                             # privilege or tier changes to groups
$J | grep 'backup restore'                         # rollbacks
$J | grep 'console start'                          # console logins (user=, from=)
```

The changes of this server's settings each write a detail line beside the `cmd` pair: `server name` and `server metrics` carry `from=` and `to=`, `server listen` adds `backend=` and `listener=`, and `server addresses` carries `op=add` or `op=replace` and `address=`. The timing scale is recorded in the same way: `scope timeout-scale` writes `scope=`, `from=` and `to=`, `device timeout-scale` a `device set` line (`name`, `field=timeout-scale`, `from`, `to`), and `config devices timeout-scale` a `config devices` line with `key`, `from` and `to`.

> A `cmd start` with no matching `cmd end` means the command did not finish (killed, crash, power loss). The snapshot named on its lines holds the state from before it. `tacctl config validate` shows the state now.

---

## 4. Evidence by control

| Auditor asks for | Evidence |
|---|---|
| List of users and their access | `tacctl user list`, `tacctl group list`, `tacctl scope list` |
| Who has administrative rights | `tacctl user list` filtered to superuser groups; `tacctl group show superuser` |
| Joiner/mover/leaver changes in the period | `grep 'cmd end'` plus `user add\|remove\|move\|scope\|disable` detail lines |
| Password resets | `grep 'user passwd'` (`source=prompt\|hash\|generated`); `grep 'user password-change'` (`state=required\|cleared`) |
| Password age policy is enforced | `tacctl config password-age`; the warnings in `tacctl status` |
| Privileged access to secrets | `grep 'secret-read'` (kind, name, by) |
| Emergency access use | `grep -E 'breakglass\|emergency lockdown'`, with reason, duration and person |
| Device configuration changes | `grep 'device config apply'` (`mode=attended` = names typed back) |
| Commands run on devices | `tacctl log accounting` |
| Failed/denied attempts | `tacctl log failures`; `grep DENY` |
| Changes can be rolled back | `tacctl backup list`; the `snapshot=` on each `cmd end` (*Backups and Undo*) |

## 5. Monthly review checklist

1. `tacctl user list`: every account has a current owner. Remove leavers.
2. `tacctl group list`: superuser and engineer membership is justified.
3. `tacctl status`: no password-age warnings beyond policy.
4. `tacctl emergency lockdown status` and `tacctl breakglass list`: nothing left in place.
5. `tacctl config validate`: no warnings left unanswered (for example, scopes without break-glass local users).
6. `$J | grep 'secret-read'`: every secret view is explained.
7. `$J | grep DENY`: look for repeated refusals of one person (probing).
8. `tacctl device config list --differs`: devices drifting from the standard.
9. `tacctl host check --all`: Linux hosts match.
10. Save the outputs with the date as review evidence.

## 6. What is never logged

Passwords, hashes and shared secrets never appear in any line. Values passed with `--hash`, `--secret`, `scope secret … set` and similar options are replaced with `<redacted>`. So are a user's email address (`--email` of `user add`, the address of `user email`: the detail line says only `op=set` or `op=clear`) and the `--note` of `backup keep`.

Free text **is** logged as typed: `--reason` for lockdowns and holds, and arguments after `--` on `ssh`. **Never put secrets in a reason.**

Values typed on a command line also land in **sudo's own log**. Use the prompting forms (`user passwd`, `scope secret <s> set`, `--hash -`, `--stdin`) so values never reach the command line.

---

*See also: Access Lifecycle · Incident Response Runbooks* · Audit log (repo) · `man tacctl` (AUDIT LOG)