Emergency Lockdown

Refusing the logins of chosen groups on a scope, and lifting it again · Checked against tacctl 0.2.25

Contents

1. When to use it 2
2. Preview first 2
3. Lock 2
    3.1 The enrolled Linux hosts 3
4. Close the gaps a lockdown cannot close 4
5. Monitor 4
6. Lift 5
7. Quick reference 5

An emergency lockdown makes the tacctl server **refuse** the logins of selected groups on every device and Linux host in a scope. It is the fastest way to shut a team or a tier out during an incident, without touching a single device. Locking and lifting are **superuser** tasks.

> **Example names used on this page** Scopes: `site-a`, `site-b` · Groups: `operator`, `readonly` · Ticket `INC-4821`

1. When to use it

Typical triggers:

- a leaked shared credential, or several users' credentials exposed;
- a suspected compromised account whose group should be frozen;
- an incident that requires an immediate change freeze on a scope;
- a suspicious script running under an operator account.

For a single account, disabling the user is usually enough (*Incident Response Runbooks* R1).

How it works: the locked groups' users are taken out of the scope when the server's configuration is rendered. The server keeps answering, and answers each login and every further command authorization with an explicit refusal (TACACS+ FAIL, RADIUS Reject). Because it refuses rather than staying silent, **devices do not fall back to their local accounts**. Superusers are never locked out.

On an enrolled Linux host the server cannot refuse a key login, which never asks it. From 0.2.23 the lockdown also reaches the hosts through `host sync` (§3.1).

---

2. Preview first

`--dry-run` lists who loses access and who keeps it, and changes nothing:

```
tacctl emergency lockdown site-b enable --dry-run
```

---

3. Lock

Pick one form. Always give a `--reason` with the ticket number (1 to 256 characters); it is stored with the lockdown and written to the audit log.

```
# Every group that is not a superuser's, on one scope
tacctl emergency lockdown site-b enable --reason "INC-4821 credential leak"

# Only the groups named
tacctl emergency lockdown site-b enable operator,readonly --reason "INC-4821 change
freeze"

# Every group of one tier (readonly, operator or engineer)
tacctl emergency lockdown site-b enable --tier engineer --reason "INC-4821"

# Every scope, including this server's own logins
tacctl emergency lockdown --all-scopes enable --reason "INC-4821"
```

- At the terminal you confirm by typing the **scope name** back, or the word `all` for `--all-scopes`. `--yes` does not answer that question; it answers only the question about syncing the hosts (§3.1).
- A further `enable` adds to what is already locked.
- This server's own login (the reserved scope `tacctl`) is locked by `--all-scopes` only, and `--all-scopes disable` lifts it; its lines say `this server's own login`. `emergency lockdown tacctl ...` is refused.

> **Never put a secret in a reason.** Reasons are logged as typed.

3.1 The enrolled Linux hosts

A device is closed by the render. A Linux host is closed by one sshd drop-in that `host sync` writes: no key login and no forwarding for any account that is not a superuser's. A host knows tiers, not the store's groups, so on the hosts a lockdown locks **every** account that is not a superuser's, even when it names only some groups. `enable` says so, in red and naming the users, when the groups it locks leave out users of the scope. Root, the superusers and the provisioning login are never matched.

After the change on the devices, `enable` and `disable` name the enrolled hosts of the scope with the command to run and, on a terminal, ask `Sync these hosts now? [y/N]`. Only `--yes` answers for you; with no terminal nothing runs and the notice is all.

```
tacctl emergency lockdown site-b enable --reason "INC-4821 credential leak" \
    --yes
tacctl host sync web1 db1          # what the notice names, if you did not answer yes
```

- `enable` also ends the live sessions of the accounts the drop-in matches (`disable` ends nothing).
- A host that is unreachable, held for break-glass or refused stays on a **pending** list. `tacctl emergency lockdown status` shows it with the command, `config validate` warns of it, and the next `host sync` of that host completes the change.
- A host enrolled or synced while the lockdown stands gets the drop-in; one synced after the lockdown is lifted loses it.

- A host provisioned through a tacctl account that is not a superuser's is refused by the sync (the lockdown would close its own key). `enable` names such a host first; enroll it through a local account.
- The exit status of `enable` and `disable` is 0 when hosts are left on the pending list: the lockdown on the devices is in force, and the warning names the hosts and the command.

---

4. Close the gaps a lockdown cannot close

A lockdown blocks only what asks the tacctl server. Check these separately during the incident:

Gap	What to do
Sessions already open (Junos and WTI keep a session until it logs in again or runs `sudo`)	Clear them on the devices. On a Linux host `enable` ends them at the sync (§3.1).
A host that was not synced (unreachable, held, refused)	`status` lists it as pending; run `tacctl host sync <host>`
Device local accounts: the Cisco console line, Junos `root` on the console	Check and, if needed, rotate them on the device
Scopes set to `local-first` (`tacctl scope aaa-order <scope>`)	`enable` warns about these in red
Local ssh keys on devices	Remove them on the device
Devices that cannot reach the server	Handle on the device
Devices under a break-glass hold	`tacctl breakglass list` (*Break Glass*)

`tacctl emergency lockdown status` lists these ways in for the scope.

---

5. Monitor

```
tacctl emergency lockdown status    # all lockdowns: groups, since, by whom, why, who is locked out,
                                    # and the hosts still pending
tacctl emergency lockdown status site-b
tacctl emergency lockdown status --json
```

While a lockdown stands:

- every command that names the scope prints a red `EMERGENCY LOCKDOWN on scope …` line;
- `tacctl status`, `tacctl config validate` and `tacctl backup restore` show it (engineers see it there for their own scopes);

- locked-out users see the red line too, so expect calls: tell the service desk which superuser to contact.

---

6. Lift

```
tacctl emergency lockdown site-b disable                # lift everything on the scope
tacctl emergency lockdown site-b disable readonly       # lift for one group only
tacctl emergency lockdown --all-scopes disable
```

The rendered configuration comes back byte for byte. When the last group is lifted, `disable` offers the sync that removes the sshd drop-in from every enrolled host of the scope (§3.1), and the lifted lockdown stays on the record, with no group, until each host has been synced.

> **There is no timer.** A lockdown stays until a superuser disables it. Agree at the start of the incident who lifts it, and when.

Verify: `tacctl emergency lockdown status` lists nothing for the scope, and the red line no longer appears on other commands.

---

7. Quick reference

Task	Command
Preview	`tacctl emergency lockdown <scope> enable --dry-run`
Lock everyone but superusers	`tacctl emergency lockdown <scope> enable --reason "<ticket>"`
Lock chosen groups	`tacctl emergency lockdown <scope> enable <g1>,<g2> --reason "<ticket>"`
Lock one tier	`tacctl emergency lockdown <scope> enable --tier <tier> --reason "<ticket>"`
Lock every scope	`tacctl emergency lockdown --all-scopes enable --reason "<ticket>"`
Sync the scope's hosts without being asked	add `--yes` to `enable` or `disable`
Status	`tacctl emergency lockdown status [<scope>]`
Lift	`tacctl emergency lockdown <scope> disable [<group>] [--yes]`

See also: Incident Response Runbooks R2 · *Break Glass* · Emergency commands (repo) · `man tacctl`