Access Lifecycle

Joiners, movers and leavers · Checked against tacctl 0.2.25

Contents

1. The access model in brief 2
2. Joiner: a new person 2
3. Mover: a change of role or territory 4
4. Leaver: offboarding 4
    4.1 Planned departure 4
    4.2 Urgent removal (termination, suspected misuse) 5
5. Passwords 5
    5.1 A required password change 5
6. Periodic access review (monthly or quarterly) 7
7. Quick reference 7

How a superuser manages a person's access from their first day to their last: accounts, roles, scopes, passwords, Linux hosts, and periodic reviews. Each section is a short procedure you can follow as written.

> **Example names used on this page** Users: `alice` (new NOC operator), `bob` (engineer), `carol` · Scopes: `lab`, `site-a`, `site-b`

1. The access model in brief

A person's **group** decides *what* they can do (device privilege and tacctl tier), their **scopes** decide *where* they can log in, and their **status** decides *whether* they can log in at all. *Concepts* §1–2 has the full model and the group table.

> An **empty scope list means no access anywhere**. It never means everywhere. The `engineer` group exists only after `tacctl group preset roles`; check with `tacctl group list`.

The tacctl server and its console are the exception: every user can log in there, whatever their scopes, unless disabled (`user disable`) or in a group of tier none. The server's own login is a scope that tacctl manages and no listing shows (*Superuser Daily Operations* §11.1).

---

2. Joiner: a new person

```
tacctl user add alice operator --scopes site-a,site-b    # 1. create; prompts for the
first password
tacctl host sync --all                                   # 2. accounts on the enrolled
Linux hosts,
                                                         #    the tacctl server included
tacctl user show alice                                   # 3. confirm (Password change
line)
```

Give alice the first password your usual way. She must change it at her first login to the tacctl server (*Getting Started* §2.1): until then no device accepts her, and the other hosts keep her account expired. From 0.2.23 the command syncs **this server's own accounts** at once (`This server's accounts were synced: 1 created`), so alice can log in here to change the password. The other enrolled hosts get one warning with the `host sync` command that brings them along; on a terminal the command also asks `Sync these hosts now? [y/N]` (the default is no, and `--yes` does not answer it). Step 2 is that sync for all hosts. The requirement and what to check are in §5.1.

An email address (optional). For the people who look after the accounts, a superuser can record one address per user. No mail is ever sent to it.

```
tacctl user add alice operator --scopes site-a --email alice@example.net
tacctl user email alice                      # show it
tacctl user email alice alice@example.net    # set or change it
tacctl user email alice clear                # remove it
```

- An address belongs to one user: another user's address is refused, naming that user.
- `user show` prints an `Email` line to every tier that can run `user show`; `user list` does not show it. `user rename` carries it along.
- The address is never in an audit line: the `--email` value is written as `<redacted>`, and the `user email` line says only `op=set` or `op=clear`.

Names are lower case: letters, digits, `_` and `-` (`alice-smith`). A period is refused, because every user has an account on the tacctl server and a period cannot be part of a Linux account name. `root` and `tacquito` are reserved.

A user stored with a period. 0.2.20 and 0.2.21 accepted a period, so a store can still hold a name with one. The user stays readable (show, list, disable, move, scopes, remove) and has no Linux account on any host, so no console login. Rename it:

```
tacctl user rename <name> <new-name>
```

Until then `config validate` reports it as an error with that command (also in `--json`; `--fix-names` does not offer it), `passwd`, `user passwd` and `user require-password-change` refuse it with `rename it first`, `host sync` skips it with a warning, and `tacctl upgrade` refuses and lists each such user. Rename them before upgrading from 0.2.20 or 0.2.21.

Password without you seeing it. The person creates a bcrypt hash on their own machine (`tacctl hash commands` prints the recipes for Linux, macOS and Windows) and sends you only the hash:

```
tacctl user add alice operator --scopes site-a --hash -    # paste the hash on standard input
```

Reading from `-` keeps the hash out of sudo's log and out of `ps`. A new user must change the password whichever way it was set; `--no-change-required` skips that, for example for a password only the person knows.

Service accounts (backup and NCM tools): use `readonly`, give them only the scopes they poll, and record the owner in your inventory. Create them with `--no-change-required`: a tool cannot change its password at the tacctl server, and no device would accept it until it did.

---

3. Mover: a change of role or territory

Change	Command	Notes
Promotion or demotion	`tacctl user move alice engineer`	Password kept. Run `host sync --all` so the host tier follows.
New site	`tacctl user scope alice add site-b`	Comma lists accepted
Leaving a site	`tacctl user scope alice remove site-b`	
Reorganization	`tacctl user scope alice replace site-a,lab`	Replaces the whole list
Name change	`tacctl user rename alice alice2`	Devices need the **new** name at login. The UID is kept, and hosts rename the account at the next sync.

After any of these, push the change to Linux hosts:

```
tacctl host sync --all
```

> **After a rename,** sync **all** hosts before you give the old name to a new person. Otherwise a host's UID check will refuse the sync.

A name may also clash with a scope's break-glass local account. tacctl refuses that and names the command that resolves it.

---

4. Leaver: offboarding

4.1 Planned departure

```
tacctl user disable carol      # day 1: logins refused everywhere; password hash kept
tacctl host sync --all         # Linux accounts expired; files kept
```

Then, when HR/retention allows:

```
tacctl user remove carol       # confirms
tacctl host sync --all         # Linux accounts deleted; asks per home directory
```

When a user is removed, `host sync` asks about each home directory at a terminal. `--remove-home` deletes the homes without asking. Without a terminal they are kept and moved to `.tacctl-removed` under `/home`, a directory only root can open. The UID stays reserved on the server.

4.2 Urgent removal (termination, suspected misuse)

Disable at once, sync, and check what the person did. The full procedure is **R1** in the *Incident Response Runbooks*.

> Disable stops **new** logins. A session that is already open keeps running until it logs in again or runs `sudo` (Junos, WTI, Linux). Ask the device owners to clear it.

---

5. Passwords

```
tacctl user passwd alice                        # reset (prompted twice); alice must
change it
tacctl user passwd alice --no-change-required   # reset without requiring a change
tacctl passwd                                   # anyone: change their own (asks the
current one)
tacctl user verify alice                        # show details and check a typed password
tacctl config password-age                      # show the password-age policy (days)
tacctl config password-min-length
tacctl hash generate                            # prompts for a password and prints its
bcrypt hash
```

`tacctl status` warns about passwords that are getting old. Every reset logs `user passwd name=… source=prompt|hash|generated` in the audit log, and never the password itself.

5.1 A required password change

A user who must change the password can log in to the tacctl server only, and there the console asks for the change before anything else runs (*Getting Started* §2.1). `user add` and your reset with `user passwd` require it by default. To require it, or lift it, on demand:

```
tacctl user require-password-change alice       # require it (on is the default)
tacctl user require-password-change alice off   # lift it
tacctl user require-password-change --all       # every user (superusers are skipped)
tacctl host sync --all                          # the other hosts expire the accounts
```

What the requirement does, until the user changes the password:

- Every device refuses the user, as it refuses an unknown user.

- Every enrolled Linux host but the tacctl server expires the account at its next `host sync` (password and ssh key both fail; files kept). The sync after the change opens it again.
- On the server, every command but `passwd`, `version` and `help` is refused with exit status 1, in the console and on the command line.
- The user's own `tacctl passwd` lifts it, and so do `user passwd --no-change-required` and `require-password-change off`.

A superuser is never held to it: the tier of the user's group decides each time. The requirement is set only for a user who can log in to the server to make the change; otherwise the command says why and sets nothing. A user cannot when:

- the server is not enrolled (`tacctl host enroll --local`);
- the name has a period (a leftover; rename it);
- the user is disabled;
- the user's group has no privilege level;
- the server is still registered in a scope it shares with devices (an install from before 0.2.21) and the user is not in that scope.

`require-password-change --all` skips each such user and a superuser, names the reason, and ends with `Flagged N, skipped M, already required K.`

Who is waiting: `user list` marks the date with `(change required)`, `user show` has a `Password change` line, and `status` counts the users. Two marks need you:

Mark	Meaning	What to do
`flagged, cannot log in to change it`	The user has no login on the server (`user show` gives the reason)	Restore the login, or `tacctl user require-password-change <name> off`
`flagged, devices in a scope shared with the server still accept the password`	The server is still registered in a scope it shares with devices (an install from before 0.2.21), which keep accepting the old password	`tacctl host enroll --local`: it gives the server its managed scope, and the devices of the old scope stop accepting the old password

A change that would leave a flagged user with no login on the server (`host forget` or `host unenroll` of the server, or disabling the backend of its login, for example) is refused with nothing changed, and names `tacctl user require-password-change <name> off`.

> **Forwarding through the server.** From 0.2.24, while anyone is held to a password change, the console's sshd drop-in closes every kind of forwarding for them, in every tier. The verbs that change who is held (`require-password-change`, `user add`, `user passwd` and the like) rewrite it, check it with `sshd -t` and undo it when sshd refuses it. A connection the user opened before keeps its forwardings until it closes; `loginctl terminate-user <name>` ends it. `tacctl console check` reports a list that is not the users held now as drift and shows both lists.

---

6. Periodic access review (monthly or quarterly)

1. **Export the facts:**

```
tacctl user list                # name, group, status, password age, scopes
tacctl group list               # groups and user counts
tacctl scope list               # scopes
```

1. **Check each item:** - Every user maps to a current person or an owned service account. - `disabled` users older than your retention period → `user remove`. - Superusers are as few as possible. - Scopes per user are no more than the role needs. - Password ages are within policy (`status` warnings). - `user scope <name>` shows no orphan scopes (they are flagged in red).
2. **Fix**, then run `tacctl host sync --all` and `tacctl config validate`.
3. **Record evidence:** the audit lines of the changes (see *Audit and Compliance*).

---

7. Quick reference

Task	Command
New user (prompts for the first password)	`tacctl user add <n> <group> --scopes <s,…> [--email <address>]`
Reset a password	`tacctl user passwd <n>`
Require / lift a password change	`tacctl user require-password-change <n> [on|off]`
Add/remove scope	`tacctl user scope <n> add|remove <s>`
Replace scopes	`tacctl user scope <n> replace <s,…>`
Change role	`tacctl user move <n> <group>`
Rename	`tacctl user rename <old> <new>`
Suspend / restore	`tacctl user disable|enable <n>`
Delete	`tacctl user remove <n>`
Push to hosts	`tacctl host sync --all`
Review	`tacctl user list`
Email address: show / set / remove	`tacctl user email <n> [<address>|clear]`

See also: Concepts · Linux Host Operations · Incident Response Runbooks · Audit and Compliance · `man tacctl`