Incident Response Runbooks

Step-by-step security and availability runbooks · Checked against tacctl 0.2.25

Contents

R1. Compromised or suspicious user account 2
R2. Leaked credentials: emergency lockdown 3
R3. Shared secret exposed 4
R4. Device locked out after a bad change 4
R5. Linux host: tacctl logins or sudo broken 5
R6. Host key changed unexpectedly 5
R7. Mass login failures, or the server not answering 5
R8. A tacctl change made things worse 6
R9. After any incident (close-out) 6

Step-by-step runbooks for security and availability incidents, written for the **superuser** on call. Each runbook gives the trigger, the steps in order, how to verify, and how to stand down. The command detail behind each runbook lives on its topic page, linked from the runbook.

> **Example names used on this page** Scopes: `site-a`, `site-b` · Devices: `sw1` (Cisco), `rtr1` (Junos) · Host: `web1` · Users: `carol`, `alice` · Ticket `INC-4821`

Runbook index

#	Incident	Main tool
R1	Compromised or suspicious user account	`user disable`, `host sync`
R2	Leaked credentials affecting a group or scope	`emergency lockdown`
R3	Shared secret (TACACS+/RADIUS key) exposed	`scope secret generate`
R4	Device locked out after a bad change	`breakglass hold`
R5	Linux host: nobody can log in or use sudo	`breakglass hold` (host)
R6	Host key changed unexpectedly	`device hostkey`
R7	Mass login failures or the server not answering	`status`, `log failures`
R8	A tacctl change made things worse	`backup restore`

Rules for every incident

- Give a `--reason` that includes the ticket number. The reason is stored with lockdowns and holds, and written to the audit log.
- Preview with `--dry-run` wherever a command offers it.
- Lockdowns and holds have **no timer**. Plan who lifts them, and when.

R1. Compromised or suspicious user account

Trigger: unusual logins in `log search`, a lost laptop, a user who reports a phished password.

```
tacctl user disable carol          # 1. logins refused on every device right away
tacctl host sync --all             # 2. account expired on Linux hosts (files kept)
tacctl log search carol            # 3. what happened: logins, sources
tacctl log accounting 200          # 4. commands run (accounting records)
```

1. Ask the device teams to clear any sessions that are already open. A disable does not end them.

2. When cleared: `tacctl user passwd carol`, `tacctl user enable carol`, `tacctl user require-password-change carol` (carol must change the new password at her next login to the tacctl server; a reset of a disabled user cannot require it), then `tacctl host sync --all`.

Verify: `tacctl user show carol` shows `disabled`. `tacctl log failures` shows the refused attempts.

R2. Leaked credentials: emergency lockdown

Trigger: credentials of several users, or a whole team, are exposed. Or you need an immediate change freeze on a scope.

A lockdown makes the server **refuse** the logins and every further command authorization of the chosen groups in a scope. Devices do not fall back to their local accounts, and superusers keep access. Full detail: *Emergency Lockdown*.

```
# 1. Preview: who loses access, who keeps it
tacctl emergency lockdown site-b enable --dry-run

# 2. Lock (pick one)
# every non-superuser group
tacctl emergency lockdown site-b enable --reason "INC-4821 credential leak"
# chosen groups
tacctl emergency lockdown site-b enable operator,readonly --reason "INC-4821"
# one tier
tacctl emergency lockdown site-b enable --tier engineer --reason "INC-4821"
# everywhere, incl. this server
tacctl emergency lockdown --all-scopes enable --reason "INC-4821"
```

To confirm, type the scope name (or `all` for `--all-scopes`). Then answer the question about syncing the scope's enrolled Linux hosts, which closes key logins on them and ends the live sessions of the locked accounts (*Emergency Lockdown* §3.1).

1. **Close the gaps a lockdown cannot close:** open sessions, device local accounts, `local-first` scopes, local ssh keys, unreachable devices, held devices and hosts still on the pending list. The checklist is *Emergency Lockdown* §4.
2. Monitor: `tacctl emergency lockdown status site-b`.
3. Fix the root cause. Typically that means R1 for the affected accounts and R3 if a secret was exposed.
4. **Stand down:**

```
tacctl emergency lockdown site-b disable             # all groups
tacctl emergency lockdown site-b disable readonly    # or one group at a time
```

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

R3. Shared secret exposed

Trigger: a scope's TACACS+/RADIUS key appeared in a ticket, a paste, a config backup outside your control, or a decommissioned device.

> Rotating the secret takes effect on the server **immediately**. Every device in the scope fails with `bad secret` until it has the new key. Prepare the device side first.

```
# 1. devices and users affected
tacctl scope show site-a
# 2. fresh pulls of the scope
tacctl device config pull --scope site-a
# 3. rotate
tacctl scope secret site-a generate
# 4. paste files incl. the new key
tacctl device config plan --scope site-a --secrets --out-dir ~/inc-4821
# 5. Junos: let tacctl apply (review each device)
tacctl device config apply --scope site-a --section aaa
```

1. Paste the Cisco and WTI plans from the `--out-dir` files. **These files contain the secret.** Store them safely and delete them afterwards.
2. Watch for devices still failing: `tacctl log failures` (look for `bad secret detected`).

Verify: `tacctl device config list --scope site-a` shows `ok` for `aaa`, and `tacctl log failures` is clean.

> **Junos tip:** if `bad secret` continues on a Junos device, delete the secret and set it again, to avoid hidden characters. **WTI:** keep the key hex-only and at most 32 characters, for example `tacctl scope secret site-a set $(openssl rand -hex 16)`.

R4. Device locked out after a bad change

Trigger: a paste cut AAA, and no tacctl login works on the device.

```
tacctl breakglass hold sw1 --reason "INC-4821 failed AAA paste"
```

1. Type the device name to confirm. The output lists the vendor's way in and the local break-glass accounts for the scope (`tacctl scope breakglass site-a`).
2. Log in with the local account. How, per vendor (Cisco, Junos, WTI): *Break Glass* §3.
3. Fix or back out. The engineer can print the back-out with `tacctl device config plan sw1 --backout`.
4. Release, then verify:

```
tacctl breakglass release sw1
tacctl device config pull sw1 --diff
tacctl ssh sw1
```

> A hold carves out **only the device's registered address**. A device that sends AAA from another source (a source interface, NAT, IPv6) is still answered. The hold report warns you when this looks likely.

Several devices at once, and everything else about holds: *Break Glass* §2.

R5. Linux host: tacctl logins or sudo broken

```
tacctl breakglass hold web1 --reason "INC-4821 sudo broken"
```

While the hold stands, the host's local administrator and console work, and tacctl users cannot log in or `sudo` there. Fix the host, then run `tacctl breakglass release web1` followed by `tacctl host check web1`. `host sync` refuses a held host until you release it. Details: *Linux Host Operations* §6.

R6. Host key changed unexpectedly

Trigger: `tacctl ssh` or a pull refuses, printing the pinned and offered fingerprints.

1. **Assume it is hostile until you have checked it.** Confirm on the console (Cisco `show ip ssh`, Junos `file show /etc/ssh/ssh_host_ed25519_key.pub`) and with the change record. Was the device replaced, reset or upgraded?
2. Only when it matches the console: `tacctl device hostkey sw1 set SHA256:<fp>`.
3. If it does not match, treat it as a possible interception. Don't connect, and escalate to security.

Full procedure: *Connecting to Devices* §5.

R7. Mass login failures, or the server not answering

```
tacctl status                 # are the backends running? errors?
tacctl backend status         # services and listeners
tacctl server show            # address, backends and every listener with its unit state
tacctl log failures           # failures in the last 24 hours; look at the pattern
tacctl config validate        # drift or broken configuration
```

Pattern	Likely cause
Many `bad secret` from one scope	The secret was changed (R3), or the devices were pasted wrong
`failed to validate` for one user	That user's password (normal), or an attack on one account (R1)
`failed to validate` for many users	Password spraying. Consider R2.
No lines at all	Service down, a listener or firewall problem, or devices that cannot reach the server
Exit 3 / `DRIFT`	A rendered file was edited by hand. `config validate` names both ways out.

R8. A tacctl change made things worse

```
tacctl backup list                 # snapshots, newest first
tacctl backup diff                 # what changed since the latest snapshot
tacctl backup restore <timestamp>  # roll back (confirms; re-renders and restarts)
```

The audit log's `cmd end ... snapshot=<id>` line names the snapshot taken just before each change. Before restoring, read *Backups and Undo* §3–4: a restore also undoes later changes, and the Linux hosts need a `host sync` afterwards.

R9. After any incident (close-out)

1. Lift every lockdown and hold (`emergency lockdown status`, `breakglass list`).
2. Run `tacctl config validate`. It should show no leftover lockdown or hold warnings.
3. Collect the audit trail: `journalctl -t tacctl -t tacctl-console --since "<start>"`, then search for `emergency lockdown`, `breakglass` and `cmd end`.
4. Write down the reasons used, and who locked, held and released what.

See also: Emergency Lockdown · Break Glass · Troubleshooting · Audit and Compliance · `man tacctl`