FOR THE OPERATOR RUNNING A FLEET

# *See the blast radius. Restore a blocked agent.*

Sentinel installs project-local, so the two commands below run through **npx**. If you operate across many repos and want the CLI on your path, add a global install once and then every command is bare **sentinel**.

```
# project-local, matches the install on the site
npm install @tuent/sentinel && npx sentinel init claude-code

# optional, for operators who want the CLI everywhere
npm install -g @tuent/sentinel   # then: sentinel scan / sentinel release
```

COMMAND ONE · ASSESS BEFORE YOU ENFORCE

## *sentinel scan*

`sentinel scan`

**Maps what a coding agent could reach.** scan walks the repo and lists every file an agent could touch, scored by how sensitive each one looks by filename and path pattern. It does not read file contents, so it is a blast-radius map, not a found-secrets claim. It is read-only and writes nothing unless you pass **--save**. Run it before you scope policy, so you tighten **forbid.targets** before it matters, not after.

RUN **sentinel scan** in a project root. Add **sentinel scan --path=../other-repo --format=json** to point it elsewhere or pipe the result into tooling.

SAFE It refuses to walk a directory that is not a project root, no package.json or .git, unless you pass **--path** or **--force**. So it never crawls a home directory by accident. Symlinks are skipped.

READ Each file is flagged high, medium, or low confidence. High means two or more sensitivity rules matched, medium is one strong match, low is everything else. Start policy from the high and medium rows.

COMMAND TWO · RESTORE AN ESCALATED AGENT

## *sentinel release*

```
sentinel release --agent=<id>
```

**Repeated violations escalate an agent.** Sentinel moves an agent normal, then restricted, then quarantined at thresholds you set: **restrictAfter** defaults to 3 violations, **quarantineAfter** to 5. release is how you bring it back.

RESTORE **sentinel release --agent=<id>** returns a restricted or quarantined agent to normal. Run it from a separate terminal if the agent's own Bash is blocked, so you are never locked out of the recovery path.

SOFT On a soft flag the developer releases a single blocked action in one click and keeps working. You do not have to be in the loop for those.

HARD Hard rules stay locked. A credential read or another default-floor deny is never released by a soft click. Those are decided deterministically and stay that way.

RELATED · PROVE WHAT RAN

VERIFY **sentinel --verify-audit --agent=<id>** re-walks the signed trail and returns a single VALID or INVALID verdict for a reviewer.

READ **sentinel --show-audit --agent=<id>** prints the actual signed entries, newest first, redacted to shape not contents by default.

scan tells you what an agent could reach before you write a line of policy. release gets a developer moving again the moment a stop was right but the work should continue. *One reduces surprise. The other reduces friction.*

ACCURACY NOTES · CLAUDE CODE IS THE ONLY LIVE ADAPTER. SCAN SCORES BY FILENAME AND PATH PATTERN, NEVER BY READING CONTENTS. FIGURES REFLECT THE PRESENT SHIPPED STATE AND MOVE AS THE PRODUCT SHIPS.

*“Know the blast radius before it matters, and never leave a developer stuck.”*

TUENT.AI
OPERATOR GUIDE