WHAT SENTINEL IS

A control layer between identity and execution.

Teams hand Claude Code to their developers. The agent reads files, calls tools, runs commands, and touches systems across the org. It executes, and it does not push back. Sentinel ships as an SDK that installs as a native Claude Code hook and checks every tool call **in process, before it runs, deterministically, with no model in the enforcement path.** The question is no longer who has access. It is whether this actor should be allowed to do this, here, now.

TWO HOOK CHECKPOINTS · FOUR DETERMINISTIC DEFENSE LAYERS

PRE_EXECUTION › ROLE BOUNDARY | INTENT ALIGNMENT | BEHAVIORAL BASELINE | HOOK PREVENTION › POST_EXECUTION

On-prem. Bring your own keys. Zero external model calls in the enforcement path. The thing protecting the agent does not inherit the prompt-injection surface of the agent it watches.

WHAT YOU GET · FOUR CAPABILITIES

PREVENT

Policy enforced before execution.

Every tool call is checked against your **.sentinel.yaml** and allowed or denied before it runs, not flagged afterward. Out of the box the policy protects credential stores, private keys, cloud configs, and system directories. A network allowlist gates tool-level requests.

DETECT

A behavioral baseline per agent.

Sentinel learns each agent's normal pattern of work and surfaces deviations from it in reports. An anomaly layer on top of the static rules, for the novel behavior a fixed list will not catch. It informs. It does not block.

PROVE

A signed record of every tool call.

Every decision, allowed or denied, is appended to a per-project, hash-chained, Ed25519-signed trail anchored by a signed manifest. **You can compromise the agent. You cannot compromise the guard.**

ASSESS

See the blast radius first.

sentinel scan walks the repo and lists the files a coding agent could reach, scored by how sensitive each looks by path pattern. Read only. A blast-radius map, so you can tighten policy before it matters.

Endpoint watches processes. Network watches traffic. Both are blind to which file an agent read and whether it matched the task it was given. *Sentinel runs where the action happens, in process, at the tool-call boundary.*

THE POLICY · ONE FILE, .SENTINEL.YAML

You do not write rules from scratch. **init** writes a starter policy with a default floor already in place; you tune it, and Sentinel recommends tightening from the actions your developers approve and deny.

```yaml
policy:
  allow:
    actions: [file_read, file_write, tool_invocation, network_request, command_exec]
    targets:        # advisory for file and tool actions, logged if outside
      - "src/**"
      - "test/**"
    networkHosts:   # a tool-level network_request is denied unless listed
      - "api.anthropic.com"
      - "github.com"
  forbid:
    targets:        # hard deny. Ships a floor for credentials, keys, cloud, system dirs
      - "secrets/**"
      - "/etc/**"
enforcement:
  restrictAfter: 3    # restrict the agent after this many violations
  quarantineAfter: 5  # quarantine after this many. One command restores it
```

ENFORCEMENT OPTIONS · ALL OPTIONAL, UNDER ENFORCEMENT

FIELD	DEFAULT	WHAT IT DOES
`scopeBoundary`	`advisory`	**enforce** denies out-of-scope file reads and writes. Advisory logs a scope violation but still runs.
`egressScreen`	`off`	Screens shell network commands against the allowlist. Detection first: it logs interpreter and obfuscated egress rather than blocking it.
`askOnDestructive`	`on`	Routes a curated set of irreversible destructive shell commands to Claude Code's confirmation prompt.
`unknownTools`	`warn`	Disposition for unrecognized tool names. **warn** allows and logs; **deny** blocks, with an allowlist escape hatch.

THE SIGNED AUDIT TRAIL · PROVE WHAT RAN

VERIFY **sentinel --verify-audit** re-walks the whole hash chain and checks every Ed25519 signature, then returns a single VALID or INVALID verdict you can hand to a reviewer.

READ **sentinel --show-audit** prints the actual signed entries behind that verdict, newest first. Paths and commands are redacted to shape not contents by default; a deny names what was reached, not the secret it would expose.

RECOVER On a soft flag the developer releases a blocked action in one click. A restricted or quarantined agent is restored with **sentinel release** from a separate terminal.

SCOPE AND SECURITY MODEL · WHAT IT IS, PRECISELY

FRAME **A guardrail with a verifiable record, not a sandbox.** Enforcement happens where Claude Code asks to run a tool, through its hook, on a cooperative agent doing normal work. It is not adversarial containment.

TRAIL **Tamper-evident, not tamper-proof.** The trail is hash-chained and signed, so tampering or a deleted entry is detectable and provable. The value is proof.

HOST **Same-host ceiling.** Sentinel runs on the same machine as the agent, so a process determined to route around it on that host can. That is where the signed trail is the backstop: you can still detect and prove what happened. For prevention against a hostile process, pair Sentinel with OS and network isolation.

DETECT **Behavioral detection is observational.** The baseline layer surfaces anomalies in reports. It is a signal to investigate, not an enforcement control.

COVERAGE AND INSTALL

AGENTS **Claude Code today, through a native hook.** That is the only live integration. Support for additional runtimes is on the roadmap; the policy model is built to extend without forking per tool.

WORKLOADS **google-workspace and microsoft-365** starter policies differ only in the network allowlist. The file and action rules are identical, so a workload is a convenient starting policy, not a separate security mode.

INSTALL **npm install @tuent/sentinel** then **npx sentinel init claude-code**. Requires Node 20 or newer. The gateway auto-starts on the next Claude Code run. Licensed Apache 2.0, open source, inspect every line.

PRESENT STATE

VERSION **@tuent/sentinel v0.1.8** is published to npm, dist-tag latest. The signed audit trail and the false-positive recovery loop are live end to end.

TESTS **3,378 tests across 188 test files, all passing** at HEAD. The test suite is written and maintained alongside the enforcement engine.

ACCURACY NOTES · CLAUDE CODE IS THE ONLY LIVE ADAPTER. CROSS-VENDOR CORRELATION IS THE NEXT BUILD, NOT A PRESENT CLAIM. FIGURES REFLECT THE PRESENT SHIPPED STATE AND MOVE AS THE PRODUCT SHIPS.

"The thing protecting the agent cannot inherit the failure modes of the agent itself."

TUENT.AI
JAMES CUNNINGHAM
CO-FOUNDER