Sentinel

Activity Report — user's manual

A plain-English user's manual

Every button, every number — explained for people who have never used a version-control tool. Keep it beside you the first few times; after that you won't need it.

What is Sentinel, in one sentence?

Sentinel keeps a governed, tamper-evident record of every change your developers and AI agents make to your code — auto-approving trusted work and holding risky changes for your one-click review.

1 Opening the Activity Report

There are two ways to open your Sentinel windows.

The easy way — the desktop icon

After your first setup, Sentinel places a **“Sentinel” log-on icon** on your desktop. Double-click it and your windows open, already signed in and sized the way you left them.

The typed way — a command

From the folder that holds `sentinel_sdk.py`:

- `python -m sentinel_sdk report` — opens the Activity Report on its own.
- `python -m sentinel_sdk open` — opens **both** windows together.

Leave it running while you work; press **Ctrl + C** in the terminal to stop it (or just close the window).

A **“User manual”** link also sits in the bottom-left of the window — it opens this same guide any time.

Integrating Sentinel into your agent workflow

Whether you're a solo builder or a team, wiring Sentinel in takes a few minutes and starts from the **git repository your AI agents commit to**. Once connected, every commit an agent makes is reported to Sentinel automatically — nothing is blocked, it's simply recorded and, when a rule is tripped, held for your review.

Before you start

- A **git repository** on your machine — the folder your agents work in (it contains a `.git` directory).
- **Python 3** installed.
- A **Sentinel subscription** and your downloaded `sentinel_sdk.py` (from Sentinel → Get the SDK).

Path A — the one-click way (recommended, no terminal)

1. Launch the Sentinel desktop app (double-click `sentinel_sdk.py` or your desktop shortcut).
2. Click **“+ Connect a repo”**.
3. Pick your git repo folder. Sentinel automatically **creates a project**, **generates an agent key**, and **installs a post-commit hook** in that folder.
4. Done — commits made in that folder now appear in Sentinel.

Path B — the terminal way (for servers, CI, or scripting)

1. Open a terminal **inside your git repo folder** (the one with the `.git` directory).
2. Install the one dependency: `pip install requests`
3. Connect the folder in a single command — this creates the project, an agent, and the hook:
   `python -m sentinel_sdk init --token YOUR_TOKEN --name "My Project"`
   (Find **YOUR_TOKEN** in the Sentinel dashboard. `--name` is optional; it defaults to the folder name.)
4. Already have a project + agent key? Install just the hook:
   `python -m sentinel_sdk install-hook --project YOUR_PROJECT_ID --agent-key YOUR_AGENT_KEY --base https://sci-ficomics.com`

Point your agents at the repo

Your AI agents (Claude, GPT, Cursor, custom scripts, CI bots — or teammates) keep working exactly as before. Every time one of them runs `git commit` in that folder, the post-commit hook quietly reports the change to Sentinel. The commit is already saved locally; Sentinel never gets in the way.

Recommended agents to use with Sentinel

Sentinel works with **any** agent (or person) that makes git commits — you're never locked in. If you're just getting started, these are free or low-cost and pair especially well because they commit straight to git:

- **Aider** — free, open-source CLI agent that commits directly to git (a natural fit for Sentinel). You pay only for whatever model API you connect.
- **Cline** and **Continue.dev** — free, open-source VS Code assistants; bring your own model key.
- **Cursor** and **Windsurf** — free tiers available; paid plans around $10–$20/month.
- **Low-cost models** — DeepSeek, Qwen Coder and Google Gemini offer very cheap (or free-tier) coding APIs you can plug into the agents above.

Whichever you choose, the setup is identical — install the hook in the repo (above), and every commit the agent makes is reported to Sentinel automatically.

Set your Guards (one-time)

In the Dashboard **Guards** tab, set **Allow-paths** (folders agents may edit freely) and **Guard patterns** (sensitive files such as `.env`, keys, and payment/auth code). Clean commits from your **primary agent** **auto-approve**; anything that trips a rule — or comes from another agent — waits in **Pending** for your Approve/Reject.

Give each agent its own identity

Separate identities are what let the leaderboard and audit log show *who* did what. Set one up per bot or person:

1. Open the **Dashboard**, select your project in the drop-down, and go to the **Agents** tab.
2. Click **"+ New agent"** and name it for the bot or person — e.g. `claude-refactor`, `cursor-cody`, or `ci-bot`.
3. Copy that agent's **key** (shown when it's created).
4. On the machine or repo where that agent runs, install the hook with *its* key:
   ```
   python -m sentinel_sdk install-hook --project YOUR_PROJECT_ID
   --agent-key THIS_AGENT_KEY --base https://sci-ficomics.com
   ```
5. Mark your most trusted agent (your lead or CI bot) as **primary** so its clean commits auto-approve. Leave the others non-primary, so their commits always land in **Pending** for a look.
6. Repeat for each agent. Every commit is now attributed to the right identity in the leaderboard and the audit log.

Give each team its own identity

In Sentinel, a **team is a project**. Give each team (or each repo) its own project so their agents, guards, and audit trail stay cleanly separated:

1. Create a **project per team** — click **"+ Connect a repo"** in the app for that team's repo, or run `python -m sentinel_sdk init --token YOUR_TOKEN --name "Team Alpha"`.
2. Name the project after the team (e.g. `Team Alpha`, `Backend Squad`) so it's easy to spot in the project drop-down.
3. Inside that team's project, add the team's agents using the steps above — each teammate/bot gets its own key and hook.

4. Set **Guards per team**: every project has its own Allow-paths and Guard patterns, so a backend team and a frontend team can run different rules.
5. Switch teams with the **project drop-down** at the top of the Dashboard — the Commits, Pending, Agents and Guards tabs all follow the selected team.
6. Each team gets its **own Activity Report and Audit log**, scoped to that project — so scorecards and trails never mix between teams.

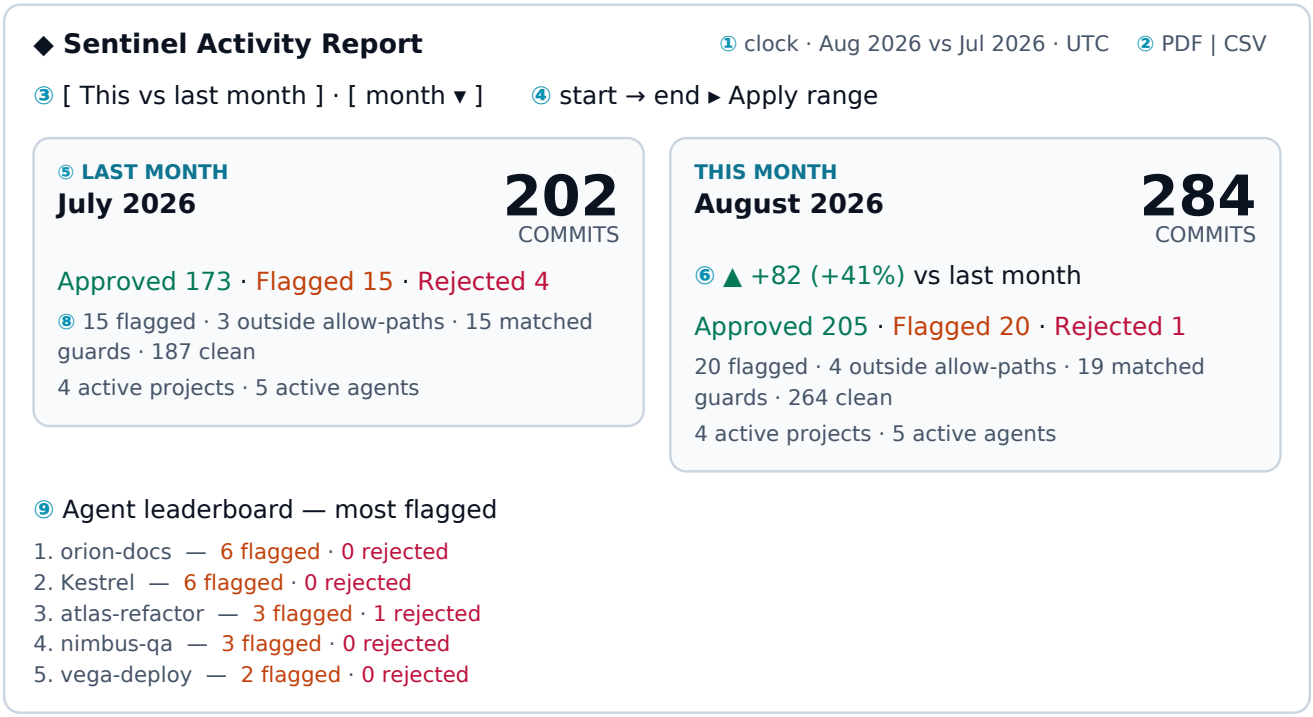
Verify it's working

Make a test commit in the repo. Within seconds it shows up in the Dashboard (the **Latest commit** banner and **Commits** tab) and in the **Activity Report**. You're integrated.

```
cd path/to/your/repo
echo "sentinel test" >> README.md
git add README.md
git commit -m "Sentinel test commit"
```

2 The Activity Report

The Activity Report is your scorecard — it compares time periods and lets you drill into anything that was flagged. The callouts below match the picture.



The Sentinel Activity Report (this month vs last month).

1. **Date & time clock.** A live clock, top-right.
2. **PDF & CSV.** **PDF** saves a printable copy of this report; **CSV** exports the numbers to a spreadsheet (great for accounting or sharing).
3. **This vs last month.** The default view. The month box lets you jump to any single month.
4. **Custom date range.** Pick a **start** and **end** date and press **Apply range** to total every commit between those dates — use a wide range (e.g. Jan 1 → today) to see your whole history.
5. **Period cards.** Two side-by-side cards — the earlier period on the left, the current one on the right — each with its total commit count.
6. **Change vs last period.** The green/red arrow shows whether activity went up or down, and by how much.
7. **Status bars.** Approved / Flagged / Rejected shown as bars with exact counts, so you see the health of the period at a glance.
8. **Flagged & guard chips (clickable).** How many commits were flagged, fell outside allow-paths, matched a guard, or were clean. **Click the flagged chip** to open the list of flagged commits; click any one for a detail panel with its message, files and the guard it tripped.
9. **Agent leaderboard.** The most-flagged agents for the period, with each one's flagged and rejected counts.

Each card also shows how many **active projects** and **active agents** contributed during that period.

3 The clock & window controls

The date & time clock

A live readout sits in the top-right corner — day, month, date, year and a ticking clock — so any screenshot or printout is self-dating.

Resize once, and it remembers

Drag the window to the size and spot you like. Sentinel remembers each window's size and position, so next time it comes back exactly as you left it.

Choosing the window style (optional)

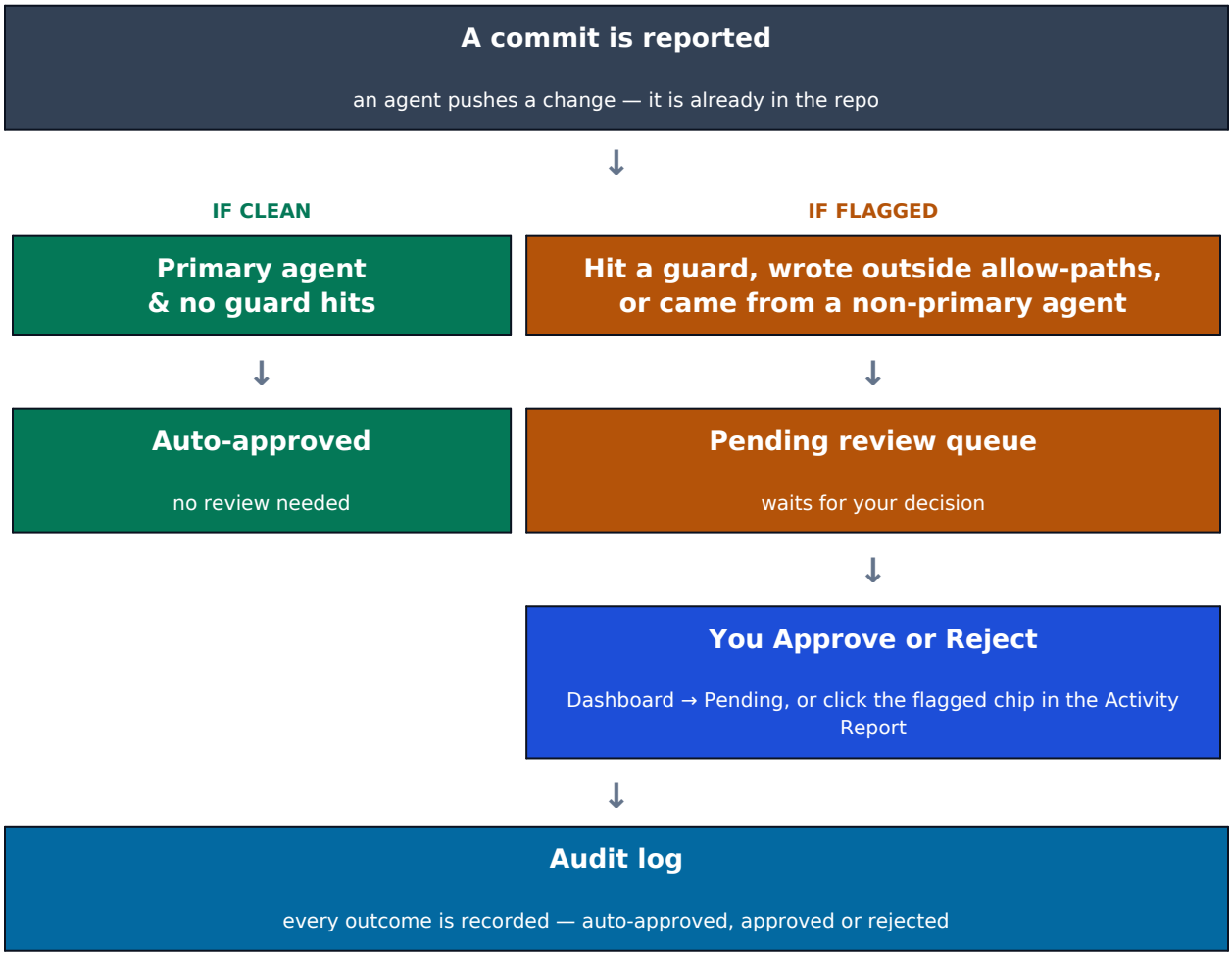
By default windows open clean, with no browser tab or address bar. You can change this without re-downloading anything:

- **app** (default) — a clean, chrome-less window.
- **sized** — its own window but *with* the browser tab/address bar.
- **tab** — just opens in your normal browser (most compatible).

Set it with `set SENTINEL_WINDOW=sized` on Windows, or `export SENTINEL_WINDOW=sized` on Mac/Linux.

Sentinel Commit Flow

Here's the whole journey a commit takes, at a glance. Sentinel is **passive** — the change is already saved in the repo; Sentinel only records it and, when a rule is tripped, asks you to take a look.



Every commit is either auto-approved or held for your review — and either way it is logged.

How to read it

- **Only a primary agent's clean commits auto-approve.** A commit is “clean” when it hits no guard and stays inside your Allow-paths.
- **Everything else is flagged** and lands in the **Pending** queue — including commits from non-primary agents, even if they look fine.
- **You clear a flagged commit** from the Dashboard **Pending** tab, or by clicking the flagged chip in the Activity Report, then **Approve** or **Reject**.
- **Nothing is ever blocked** and no local work is undone — Approve/Reject only writes to the permanent Audit log.

4 Everyday things you'll do

See your whole history, not just this month

Use the **date range** (start → end) and press **Apply range** — set a wide span to total every commit across many months.

Compare two months

Leave it on **This vs last month**, or pick any single month from the month box to see it beside the one before.

Investigate what was flagged

Click the **flagged** chip on a card to list those commits, then click one to see exactly which guard or allow-path it tripped.

Save or share a report

Click **PDF** to print/save a copy, or **CSV** to open the numbers in a spreadsheet.

Words you'll see (glossary)

Commit — A saved change to your code. Sentinel records each one.

Flagged — A commit that tripped a guard or wrote outside allow-paths.

Clean — A commit that tripped no rules.

Allow-path — A folder/file agents may change freely without being flagged.

Guard / matched guard — A danger-zone file rule a commit hit.

Approved / Rejected — Your review decisions — recorded forever.

Quick fixes

The report looks empty or out of date

Close the old window first, then reopen from the desktop icon. The report refreshes on its own.

The two “Flagged” numbers differ from Approved + Rejected

That's expected: “Flagged” counts commits that tripped a guard, so Flagged + Clean = total (the bars are a separate review-outcome view).

Staying signed in & the session-health dot

Sentinel keeps you signed in automatically. While the Activity Report window is open it quietly refreshes your access token in the background, so your session never expires while you are using it. You almost never have to do anything — and you never need to open a terminal or paste a command.

The session-health dot

Next to the date & time clock (top-right of the window) is a small colored dot that shows your session status at a glance. Hover over it to see a short message, including when your token last refreshed.

- 🟢 **Green — healthy.** Your session is valid and auto-renewing. Nothing to do.
- 🟠 **Amber — needs attention.** Your session is within a day of expiring, or has already expired (this only happens if Sentinel has not been opened for about a week). Just click the dot — Sentinel fixes it for you.

If the dot is amber: just click it

Clicking the dot is all you need. Sentinel first tries to renew your session silently, right where you are — no terminal, no copying, no visit to the website. If it works, a small toast confirms “Session refreshed” and the dot turns green.

The “Sign in again” box

If your session has fully expired (Sentinel unopened for more than a week), a silent renew is no longer possible, so clicking the dot opens a small **Sign in again** box inside the window. Enter your Sentinel email and password and click **Sign in & restore session**. That is all — the Dashboard and Activity Report load again immediately.

What the re-sign is for: a security token only lasts seven days. Signing in again creates a brand-new seven-day session and saves it to this computer so Sentinel can keep renewing it on its own. Your password is sent only to Sentinel to obtain the new token — it is never stored on your machine or shown to anyone.

As long as you open Sentinel at least once a week, the dot stays green and you never have to think about tokens at all.

Need a hand? Email support@sci-ficomics.com and we'll help you get set up.