DIGITAL FOUNDRY SOLUTIONS, LLC

actualis

See what your coding agents actually did.

actualis.app · github.com/digital-foundry/actualis

PROBLEM

Agents run shell commands. Nobody reads the logs.

Did an agent see a production key?

Agents read .env files and config. The secret lands in a transcript no one opens.

What ran without approval?

Auto-approve modes are becoming the default. Installs, curl | sh and git push run unreviewed.

What came in from the network?

Supply-chain attacks on npm and PyPI target exactly the installs agents run unattended.

Every session leaves a transcript on disk, in each vendor’s own format. None of it gets reviewed.

PRODUCT

One command, five agents, nothing leaves the machine

```
$ actualis --days 7 --fail-on high
SECRETS 2 credentials reached agent context 1 critical: rotate
SHELL 1,284 commands · 61% ran without approval · 3 high
NETWORK 212 downloads · 38 unasked · 1 matched an IOC list
COST $412.80 across 3 models · cache saved $96.10
exit 3: findings at or above "high"
```

Illustrative output, invented figures. Reads Claude Code, Codex, Copilot CLI, Gemini CLI and Qwen Code.

WHERE IT RUNS

Wherever the agents run

CI gate

--fail-on high fails the build on a leaked key or a known-bad install.

GitHub Action

A drop-in workflow step for any repo.

MCP server

An agent can query its own record before it acts.

Live monitor

--watch flags risky commands as sessions happen.

Menu-bar app

macOS and Windows tray with a status at a glance.

HTML report

One file to share with a security lead.

pip install actualis · brew install digital-foundry/tap/actualis

TRUST

Don’t trust us. Run the check.

actualis --self-check proves on your own machine that the core imports no networking code and changes no file it reads.

No account · no config · no telemetry · one readable file

WHY NOW

Agent adoption is ahead of agent oversight

Agent mode by default	Coding agents now run shell commands, often in auto-approve modes.
Supply chain under attack	Malicious npm and PyPI packages target the installs agents run unattended.
Logs split by vendor	Each vendor’s logs cover only that vendor. Teams use several agents.

actualis reads all five formats in one place: the neutral audit layer.

TRACTION · EARLY

Shipping, and early on distribution

444
PyPI downloads, last 30 days

59
unique cloners, last 14 days

1,880+
automated tests, 3 OSes

5
agent formats supported

Released on PyPI, Homebrew and actualis.app. 0.5.0 adds offline host reputation. No paying users yet. Recruiting the first design-partner teams now.

Figures as of 2026-10-31: pypistats.org and GitHub traffic.

BUSINESS MODEL

Open core: free CLI, paid team layer

FREE · AGPL

actualis

Does the adoption work. No account, no config, no network. Every developer runs it on their own machine.

PAID · PER SEAT · PRICING IN DESIGN

actualis-pro

- Encrypted, tamper-evident team history
- PII gate for CI
- Central policy and dashboards
- Commercial threat-intel feeds

Buyer: security leads at companies where agents already have shell access. actualis-pro is in design.

COMPETITION

Others see part of the picture

Approach	What it sees	What it misses
Secret scanners	Secrets committed to repos	What an agent read or ran
Vendor admin logs	One vendor’s sessions, in its cloud	The other four agents
AI security gateways	Prompts in the network path	Shell commands and installs on the machine
actualis	**All five agents, on the machine, offline**	**Nothing leaves the machine**

Moat: neutrality across vendors, and a privacy claim anyone can verify.

TEAM AND ASK

Aaron Wise

Founder, Digital Foundry Solutions, LLC

Builds and ships a portfolio of products with fleets of coding agents. Solo, took actualis from first commit to 0.5.0 in ten weeks: about 20 releases, 1,880 tests, three security review passes.

The ask

Anthropic credits to build actualis-pro, and introductions to three to five teams running coding agents at scale who will become design partners.

aaron.wise@digitalfoundry.tech · actualis.app