NEURALMIND · OCTOBER 2026

13 releases in 2 weeks

v4.16 to v4.22 in detail; v4.23 in the release notes

NEW IN V4.23 · MEASURED OFFLINE, NOT IN AGENT RUNS

Code first, unless the prompt names a test
A matching test brings the code it tests
In time for Hermes, even on a large index

A local code index for Claude Code, Cursor, Cline and Hermes Agent

EXECUTIVE SUMMARY

The short version

AI agents on CUI code	One opt-in setting for the hardened configuration, signed audit checkpoints, live audit to syslog, an OSCAL file and a 17-rule hardening check. Your assessor decides compliance.
Context you can trust	A stale index says so, a feature branch stays on its branch, doctor shows whether anything calls NeuralMind, and the demo runs offline.
Evidence	The right file on 40 of 40 benchmark queries (ripgrep: 32 of 40) and 140 of 150 on a larger eval, misses published. A pilot found no measurable cost difference: the agent called a NeuralMind tool in 0 of 60 runs.

> Getting agents to use what NeuralMind finds is the work now.

WHAT NEURALMIND IS

The relevant code for the agent’s question, not whole files

NeuralMind indexes a codebase on the developer’s machine. When an AI coding agent asks about the code, it hands over the relevant functions, files and history. Install: pip install neuralmind

01

Controls for AI agents on CUI code

The logs, configuration and paperwork an assessor asks about.

02

Context the agent can trust

Stale answers say so, branches stay separate, and doctor checks the wiring.

03

Evidence, not adjectives

Every number from committed data, including a pilot that found no difference.

PART 1 · AI AGENTS ON CUI OR FEDERAL CODE

An assessor asks three questions about any tool that reads your code

In a CMMC enclave or a system under NIST SP 800-53, an assessor treats an AI coding tool as part of the system being assessed.

What does it log?

Signed audit checkpoints

Where do the logs go?

Live forwarding to syslog

Is it set up the way your plan says?

One setting, OSCAL, and a 17-rule check

PART 1 · V4.16.0, V4.17.1, V4.17.2

One setting for the hardened configuration

BEFORE

Running NeuralMind on CUI took four separate decisions, made and checked by hand.

Nothing stopped someone from opening the graph view to the whole network for a demo.

NOW

```
security.profile: cui
```

- Opt-in: nothing changes until someone sets it
- Refuses changes that would weaken it, and logs each refusal
- Can be set in /etc/neuralmind/policy.yaml, a file a developer can’t delete
- An invalid value holds every project to the strictest settings: it fails closed

PART 1 · V4.16.0

Signed audit checkpoints

BEFORE

The audit log is hash-chained, but the chain has no secret. Anyone who can write the file can delete the newest records, or edit one and recompute every hash after it.

NOW

Checkpoints record the record count and the last record’s hash, signed with a key the developer’s account can’t read. An administrator can keep a copy where only they can write.

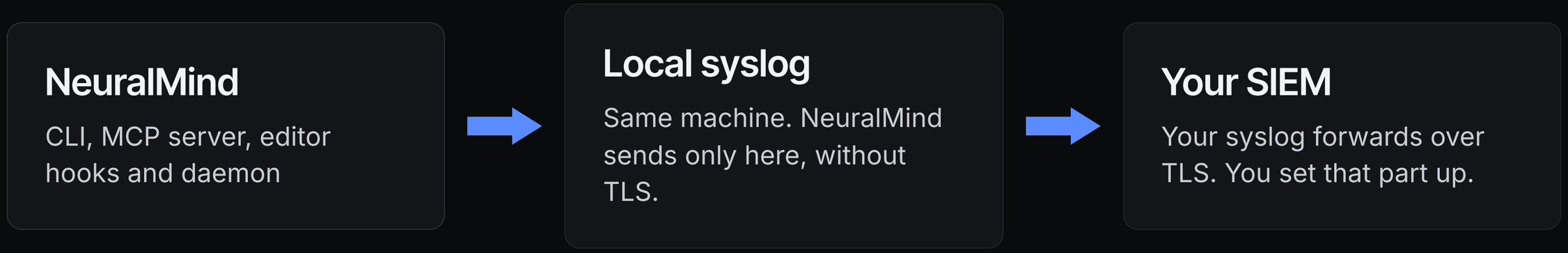
```
neuralmind audit verify
```

fails if records a checkpoint covers are removed or rewritten.

> Records written after the newest checkpoint rely on the hash chain until the next one: the checkpoint interval is the detection window.

PART 1 · V4.19.0 · SECURITY.AUDIT_SYSLOG

Audit records reach your SIEM as they’re written



- If a record can’t be written, an alert goes straight to syslog, not through the log that failed.
- Each record carries its hash-chain fields, so the SIEM can check the chain it receives.

Relates to CMMC practices AU.L2-3.3.4 (alert when audit logging fails) and AU.L2-3.3.5 (correlate audit records). It gives you evidence for them; your assessor judges the practices.

PART 1 · V4.19.0

Paperwork for your plan, and a baseline that checks itself

OSCAL component definition

OSCAL is NIST’s machine-readable format. This file states which SP 800-53 controls NeuralMind implements and which stay your responsibility.

Teams import it into their System Security Plan tooling.

```
neuralmind hardening
```

- 17 rules, each mapped to SP 800-53 controls
- Pass, fail or not applicable, with the fix for each failure
- Exits non-zero on any failure, so it can run on a schedule
- --json keeps the result as assessment evidence

PART 1 · V4.18.0

Every place NeuralMind keeps a project’s data

BEFORE

The data deletion procedure tried to name every location by hand.

Seven rounds of review each found a place it had missed.

NOW

```
neuralmind data-locations
```

- Lists every location, inside the project, outside it, and in the integrations that write it
- Says whether each one holds anything
- Comes from the code that writes each location, so it can’t drift from the code

WHAT THIS DOES AND DOESN’T MEAN

These features support controls in your system.

Your assessor decides compliance.

They give you logs, paperwork and checks as evidence. NeuralMind issues no certificate, and it doesn’t decide an assessment.

PART 2 · CONTEXT YOU CAN TRUST · V4.17.0

An answer from a stale index says so

A hands-on audit of 12 repositories found 4 with dead semantic search. Their index was in an old layout NeuralMind no longer reads, and search answered as if nothing were wrong.

THE LINE THE AGENT NOW SEES (EXAMPLE FROM THE RELEASE NOTES)

```
Index is stale: 2 files changed in commits since the last build (src/auth.py, src/session.py);
run neuralmind build <project>.
```

It appears in neuralmind query output, in the MCP tools' answers and in the block added to each prompt, and it ends with the command that fixes it.

PART 2 · V4.22.0

A feature branch stays on its branch

BEFORE

A function call added on a feature branch was still recalled after switching back to main.

The memory file Claude Code loads at the start of a session listed what the agent had learned on every branch.

NOW

Each build replaces the call graph the last one recorded, so the branch’s calls leave recall when you switch back to main.

The memory file lists only what applies to the branch that’s checked out.

PART 2 · V4.22.0

Is anything actually calling NeuralMind?

11 days

One Hermes Agent deployment sat installed but unused, with 2 logged queries.

The daemon wasn’t running, the git hooks and the Hermes plugin weren’t installed, and doctor still said “Up and running”: every check it ran was about the index.

```
neuralmind doctor
```

now opens with a Wiring section:

- Is the daemon running?
- Are the git hooks installed?
- Is the Hermes plugin enabled?
- Which agent clients have it registered?

Each warning comes with the command that fixes it, such as neuralmind install-mcp --client hermes.

PART 2 · V4.19.0, V4.21.0

Easier to try

```
neuralmind demo
```

Runs offline straight after install, with no model download, on Linux, Apple Silicon Macs and 64-bit Windows.

It asks three questions of a small sample project bundled with the package.

Other platforms

Intel Macs: NeuralMind falls back to ChromaDB, so builds work there. Installing still needs a compiler toolchain for one dependency.

Windows on ARM: use WSL 2 or the container image.

Every number on neuralmind.uk comes from committed data

40 of 40

public-benchmark queries where NeuralMind found the file the fix changed

Keyword search with ripgrep: 32 of 40

140 of 150

questions on a larger eval with the right file in the top five (93.3%)

All 10 misses are published

> Forty pre-registered queries is a small sample, so the larger eval always sits beside it. Every figure, with its source and the command that reproduces it: neuralmind.uk/benchmark

PART 3 · TASK-OUTCOME PILOT

A pre-registered pilot found no measurable difference

+0.0%

median change in cost per task

95% confidence interval −5.2% to +8.6%

20
real bug-fix tasks

120
runs: Claude Code with and without NeuralMind, three times each

18 of 20
tasks solved either way

PART 3 · WHY IT MADE NO DIFFERENCE

The agent never called a NeuralMind tool

0 of 60

runs with NeuralMind installed in which the agent called one of its tools

It saw 28 tool names with no descriptions, and no word on when to use any of them.

The block NeuralMind added to each prompt named the file the fix changed in 17 of 20 tasks.

The agent searched the codebase with its own built-in tools anyway.

PART 3 · V4.21.0

What v4.21.0 changed, and what a rerun found

What changed

- The MCP server says what its tools are for and when to use them
- The prompt block includes the leading file's code
- Six tools by default instead of 28

-2.1%

cost change in an exploratory rerun, not pre-registered

95% CI −6.9% to +6.5%: within noise. Still no NeuralMind tool calls in 60 runs.

> Better retrieval only helps if the agent uses it. Getting agents to use what NeuralMind finds is the work now, and the next result will be published the same way.

Small studies: 20 tasks, one agent, one model.

THE LIMITS, STATED

What NeuralMind does not claim

Faster or cheaper tasks	Not shown. The pilot found no measurable difference.
Compliance	It supports controls in your system, and your assessor decides. It issues no certificate.
Flawless retrieval	Results are counts on a stated number of queries, with every miss published.
Security scanning or hardware	It’s software on a developer’s machine. It doesn’t look for threats or malware.
That agents need it	An agent works without it. In the pilot, it solved the same tasks either way.

READ MORE

Every release, and every number

neuralmind.uk/whats-new	Every release, with its notes
neuralmind.uk/benchmark	Every benchmark number and every miss
neuralmind.uk/security	Release verification and the compliance documents

```
$ pip install neuralmind
```