NERO PROJECT

Adversarial Review - AI Reviewers Attack Nero's Rules

Three independent reviews in nine days. One ran overnight with no one watching.

Paul Schindler | Built with Claude | November 2026

Executive summary

Nero is the AI operating layer I run at home, and its agents work under written rules that I approve. In July 2026 I had those rules and the running system examined by three independent AI reviewers. Each worked in a fresh session with read-only access and instructions to find where the rules fail. The second review ran overnight in an isolated folder with no one watching and returned 14 ranked findings. Its first finding was that my always-on messaging and voice agent held far more authority than its written list allowed, and I confirmed that by experiment before I accepted it. I then re-recorded the authority model, retired a trust counter that measured nothing and logged a written decision on every finding.

3
independent reviews in July 2026

14
ranked findings from the overnight review

224
files the first reviewer read end to end

0
changes a reviewer was able to make

What is in this report

Page	Contents
2	The problem, why it mattered and the approaches I ruled out
3	How the overnight review was run, what went wrong during it and the three reviews side by side
4	What the reviews found, how I confirmed the most serious finding and what I changed
5	Results and status, the checks on the reviewers themselves and what this shows about how I work

My role

I commissioned each review, wrote its instructions and its limits and decided what to do with every finding. AI models did the reading and the judging. I am not a programmer, so each finding had to reach me as a claim in plain English with its evidence attached.

The finding I value most

The reviews showed that I had been counting activity as evidence of safety. A procedure that had run many times looked trustworthy to me, and the count behind that impression had never been kept. I now grant autonomy from results I have observed.

1. The problem and why it mattered

My wife and I wrote Nero's rules over many working sessions with the same AI agents the rules govern. By July 2026 those agents could change home-automation settings, operate locks and the alarm and run unattended overnight. I had two reasons to doubt my own reading of the rules.

I had approved every rule

A person who approves a rule tends to read it the way he meant it. I was the reader least likely to notice a rule that promised a protection it did not deliver.

The system had kept growing

Version 2.0 of the rules took effect in June 2026. During the month that followed I added agents and permissions one approval at a time, and nobody had compared the total with the rulebook.

The overnight review put the risk in one sentence, which I have kept.

> *"This is the load-bearing gap between what you built and what you think you built."*
>
> Overnight review report, July 2026, on the authority of the always-on agent.

2. What I ruled out

Approach	Why I ruled it out
Reviewing the rules myself	I wrote them and I do not read code. I can judge a plan written in plain English and a result I can see. Neither skill finds a permission that should not exist.
Asking the agents that helped draft them	An agent that followed the drafting holds the same assumptions I do. A reviewer that has read the argument for a rule tends to agree with it.
A checklist audit	A checklist confirms what its author already thought to ask. I wanted findings I had not thought of.
A reviewer that could fix what it found	A review that changes the system alters its own evidence and adds changes nobody approved. Every review was read-only, and every fix went through my normal approval afterward.
Accepting findings as delivered	An AI reviewer can be wrong with full confidence. Each finding was checked against the files or by experiment before any rule changed.

What every reviewer was told

The task	Read the rules and the system as it is built. Report where the rules fail, promise more than they deliver or describe a protection that does not exist.
The standard	Accuracy over approval. Name the file behind every claim and mark each statement as a fact, an inference or an unknown.
The limit	Change nothing, however small or obvious the fix. Findings go in the report.
The honest ending	Close with a list of what you could not verify.

Why the reviewer had to be a stranger

Each review began in a new session that held no record of how the rules were written. The first reviewer met the system for the first time and was told to skip the folder that held an earlier self-review, so that its findings would be its own.

3. How the overnight review was run

1. Fresh folder
A new session in its own folder, with no history of the rules.

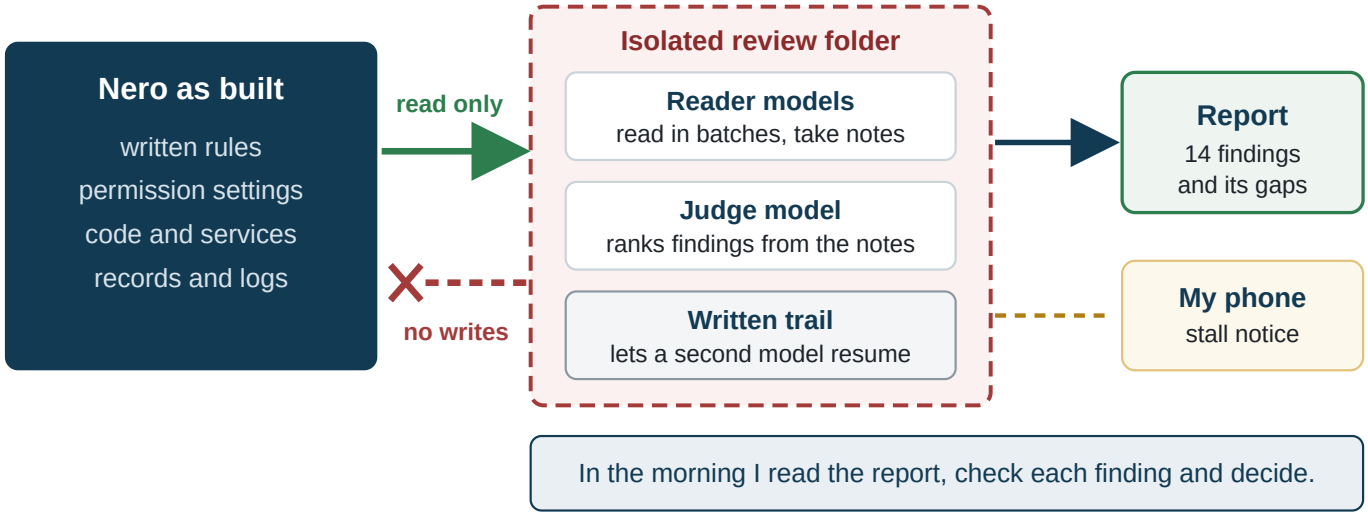
2. Fence
The strictest permission mode, a short allowed list and the command guard.

3. Read
Lower-cost reader models work through the files in batches and log each one.

4. Judge
A frontier model ranks the findings by severity and confidence.

5. Report
One document, with evidence for each finding and a list of gaps.

6. Heartbeat
A notice to my phone if the run stalls.



The reviewer could read the rules, the settings, the code and the records. It could write only inside its own folder. Nothing it produced reached the live system without my decision.

What went wrong during the run

Two things went wrong, and both are in the record. Two hours in, the judge model's safety system began refusing to continue, because the notes it had gathered were dense with home-security detail. The run switched to the fallback model I had named in advance, which finished the report from the written trail in one 16-minute pass. The fence also proved stricter than I intended and blocked the reviewer's live checks, so the review worked from files alone. Its report named four claims it could not verify for that reason.

The three reviews of July 2026

Date	Reviewer and method	Output
July 1	An independent AI reviewer on first contact with the system. It read all 224 tracked files end to end and re-opened every cited line after drafting.	12 ranked findings
July 4	The overnight review described on this page, with reader models, a judge and a reading log of 290 files.	14 ranked findings and a verdict of "conditional go"
July 9	An operations and control review. Six read-only reader agents and live read-only checks covered which agents can act, how much reaches me for approval and where the records had drifted.	5 headline findings and 8 ranked trust risks

I saved the overnight setup as a written procedure so that a later review can reuse it. The brief for the July 9 review is written to be run again when the system has grown.

4. What the reviews found

The most serious finding concerned the agent that answers our messages and voice requests around the clock. The overnight review found that its real authority was far wider than the short list the rules gave it.



The always-on agent ran in the same folder as my keyboard sessions, so it loaded their permission file as well as its own list. Each approval I gave during ordinary work also widened the unattended agent.

I tested the claim before I accepted it

The reviewer said plainly that it had not been able to run the proof, so I ran it. In a test session launched with a short allowed list, a command permitted only in the shared file ran without a prompt. The finding was correct.

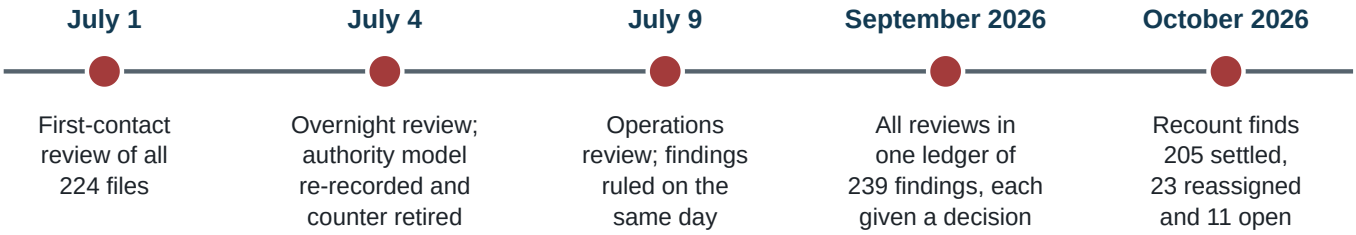
Findings and what I changed

Finding	Review	What I changed
The always-on agent inherited every permission approved in my keyboard sessions.	July 4	Its launch settings now name specific commands and refuse the most dangerous inherited ones outright.
"Voice cannot disarm the alarm" was an instruction to the model. No code enforced it.	July 4	Code now decides. Since September 2026 voice disarm works only when the code confirms a household phone is home.
The rules still said agents were read-only everywhere. In practice agents applied approved changes and one acted unattended.	July 1, July 4	I re-recorded the authority model on July 4 as two explicit postures, one for keyboard sessions and one for the always-on agent.
Each procedure carried a count of clean runs that was meant to earn it autonomy. Nothing ever updated the count.	July 1	I removed the counter on July 4. A person now assigns each procedure a trust level from observed results.
The whole operating layer existed as one copy on one disk.	July 1, July 4	The files and the database are now backed up every night to separate storage.
The command guard honored its override word anywhere in a command, and a failure inside the guard left no trace.	July 1	The override counts only at the start of a command. Every override and every guard failure is logged.

What I accepted instead of fixing

My own keyboard sessions keep a broad list of allowed commands, because a narrow list would block ordinary work. Their limit sits in the command guard and in my approval of each plan, and I recorded that decision beside the finding.

5. Results and status



The July reviews fed a single ledger that also holds the findings of a June self-review and a September audit by a reviewer from a second AI provider.

Measure	Status in November 2026
Reviews	Three in July 2026, each read-only. No reviewer changed a file outside its own folder.
Rule changes	The authority model was re-recorded and the trust counter retired on July 4, 2026, the day the overnight report arrived.
Findings ledger	239 findings from six review passes between June and September 2026. I gave each one a decision in September 2026. At the October 2026 recount 205 were settled, 23 had moved to other workstreams and 11 were still open.
My own workload	The July 9 review counted 133 approvals and check-ins reaching me each week. That count led me to build one daily report that triages open items, and in November 2026 to commission a draft charter for delegating routine decisions.

6. Checks on the reviewers

1. **Evidence with every claim.** Reviewers named the file behind each finding and marked statements as fact, inference or unknown. A finding without a source did not reach the ledger as a fact.
2. **A check before any change.** I confirmed the most serious finding by experiment. The first reviewer re-opened each cited line after drafting and corrected its own references. No rule changed on a reviewer's word alone.
3. **A tested reviewer.** In July 2026, before I accepted one review agent's verdicts, I gave it five test notes without saying it was a test. Four carried planted errors and one was clean. It caught all four and passed the clean one.
4. **Claims without the argument.** A high-stakes proposal goes to a reviewer that receives the claims and the supporting data and is not shown the reasoning behind the proposal. A reviewer that has read the case for a proposal tends to agree with it.
5. **A list of gaps.** Each report ended with what its author could not verify. The overnight report listed four such claims, and I treated them as open until someone checked them.

7. What it shows about how I work

I assume that a policy I wrote has faults I cannot see, and I commission an independent review before I rely on it. I give the reviewer read access, clear limits and a duty to show its evidence, and I test the reviewer as well. Every finding receives a written decision, including the findings I decline to fix. The same practice applies to a grant program or a partnership agreement, where an outside review before launch costs less than a failure after it.

I commissioned the reviews, set their limits and decided every finding. AI models did the reading and the judging. Details of the home's systems are left out on purpose.