{ agent security audit }

Bring us an agent to break.

A fixed-scope security audit of one agent, MCP server or agentic payment flow. One to two weeks. You get the threat model, the findings with file and line, the patch with tests, and a report your security team can act on.

WHAT WE AUDIT

URL and network gates. Everything the agent can fetch or browse: link-local and cloud-credential ranges, parser bypasses, redirects, DNS rebinding. Forty test vectors with expected verdicts.

Payment and spend paths. Who dictates amount, asset and recipient, where the pre-signature decision point is, and which primitive to ship first: audit log, asset pinning, allowlist, caps.

MCP and remote read surfaces. Every operation exposed over MCP, checked against the privacy, tenancy and deletion filters the primary read path applies. We find the one that forgot.

Audit trail. Whether an autonomous decision can be reviewed after the fact, and whether the log would survive tampering. We ship a signed, hash-chained format if there is none.

WHAT YOU GET

Threat model. The four axes for this system: the model as a non-boundary, the process surface, third-party trust, missing web-layer assumptions. Grounded in your code, with file and line.

Findings, ranked. Each with the vector that proves it, the impact stated precisely (who can trigger it, from where, what they reach), and the smallest fix that matches your own conventions.

Patch with tests. A pull request against your repo, with the vectors as regression tests. If a finding exposes live credentials in production, it goes through your security process instead of a PR.

Report. One document your security team reads without us in the room. What was audited, what was found, what was fixed, what stays open and why.

HOW IT RUNS

Day 1: access to the repo and a 45-minute call on what the agent can do and who it acts for.

Days 2 to 7: the audit, in your repo, with findings shared as they are confirmed. No surprises at the end.

Days 8 to 10: patches, tests, report. A closing call with your engineers, not a slide deck.

SCOPE AND PRICE

One system per engagement. Fixed price agreed before day 1, based on the size of the surface. A second system is a second engagement, not a change order.

METHOD, IN PUBLIC

The audits are the same ones packaged as skills in agent-security, and the same ones that found a range gap in gstack's browser daemon, a missing decision point in an x402 payment server, and two gaps in our own payment guard, all fixed with tests. github.com/tuturama/agent-security

---

Tuturama. An AI-native software factory for systems that cannot break.
Tuturama OÜ · Reg. 16022682 · Tornimäe 7, 10145 Tallinn, Estonia

tuturama.com/audit
contact@tuturama.com

{ tuturama }

An AI-native software factory for systems that cannot break.

products
upstream
agents
audit
how we work
about
archive

An Estonian company. European innovation.

Tuturama OÜ · Reg. 16022682
Tornimäe 7, 10145 Tallinn, Estonia
contact@tuturama.com