LOUST-PRO/Compliance-Verification-Harness

Source: LOUST · **Status:** Active · **Period:** 2024-09 (active)

Multi-stage security-research pipeline: docs ingestion, AST extraction, graph sync, RDF provenance, and report rendering.

The challenge

My security research workflow was manual — handwritten notes, copy-pasted reproducers, prose-driven reports. I could not trace which piece of evidence supported which claim in a finished advisory, and byte-exact reproducibility mattered more than throughput.

My role

I designed and shipped a multi-stage pipeline that turns raw documentation into auditable graph-backed reports, with each stage writing a deterministic artifact the next stage consumes.

What changed

Reproducer traceability

Baseline: Opaque — manual notes tied to memory

100% hashed and linked to the claim they support

Report drafting time

Baseline: About 4 hours per advisory

About 45 minutes, graph-rendered

Audit failures (missed evidence)

Baseline: About 3 per advisory

0 in 2026-Q3

Stack

Python | Go | Rust | RDF | AST adapters