Reverify

Deterministisches Verifikationswerkzeug für KI-gestützte Reverse Engineering, das jeden Modell-Anspruch gegen die tatsächlichen Bytes einer Binärdatei prüft.

STERNE	LIZENZ	SPRACHE	VERSION	ANGELEGT	ZULETZT GEÄNDERT
1.107	MIT	Python	v0.11.0	2026-08-31	2026-09-07

Was es ist

Reverify ist ein Python-Toolkit, das Sprachmodelle bei der Analyse von Binärdateien auf die Probe stellt. Das Modell stellt Behauptungen auf (zum Beispiel über Strukturen, Funktionsaufrufe oder Algorithmen), und deterministische Werkzeuge prüfen diese gegen die tatsächlichen Bytes — nicht gegen das Modellwissen. Das Ergebnis ist VERIFIED, REFUTED oder INCONCLUSIVE, jeweils mit Nachweis. Das Projekt ist als Kommandozeilenwerkzeug und als MCP-Server für Agenten wie Claude Code oder Cursor verfügbar.

Was es kann

- PE-, ELF- und Mach-O-Dateien parsen sowie x86-, x64-, ARM- und ARM64-Befehle disassemblieren, optional mit capstone, unicorn und lief.
- Behauptungen über Binärdateien verifizieren: Bytes an Offset, Instruktionsfolgen, Emulationsergebnisse, Funktionsaufrufe, Referenzen, Erreichbarkeit.
- Eine geschlossene Schleife betreiben: Modell schlägt vor, Werkzeuge prüfen, Modell korrigiert — bis alles grounded ist.
- Zustand über Kontext-Resets hinaus speichern: pro Binärdatei eine Ledger-Datei unter .reverify/ledger/.json.
- Quellcode-Äquivalenz prüfen: reverify equiv --lang python vergleicht Ausgaben bei gemeinsamen Eingaben.
- Als MCP-Server laufen, den Claude Code, Cursor und andere Agenten direkt aufrufen können.

So fängst du an

Installation über PyPI, dann entweder als Kommando reverify oder direkt aus dem Quellcode.

```
pip install reverify
pip install "reverify[full]"
reverify auto sample.bin --json
python reverify/cli.py auto sample.bin --json
python reverify/cli.py parse-pe sample.exe --json
python reverify/cli.py disasm 90505831C0C3 --arch x86_64
python reverify/mcp_server.py
```

Grenzen und Bedingungen

- Die semantische Schicht (Funktionsgrenzen, Aufrufgraphen, Cross-References) benötigt angr als optionalen Zusatz; ohne Engine sind diese Anfragen INCONCLUSIVE.
- Die Verifikation semantischer Behauptungen basiert auf heuristischen Analysen (CFGFast) und erreicht nur den Tier DERIVED, nicht VERIFIED.
- Die Ledger-Funktion überlebt zwar Kontext-Resets, aber bei Claude Code und Codex CLI kann reverify die Sitzung nicht selbst beenden — ein Hintergrundprozess oder eine Desktop-App umgeht den Hook.
- Die Lizenz ist MIT, aber das Werkzeug ist ausschließlich für autorisierte Reverse Engineering gedacht — Malware-Analyse, CTF, Interoperabilitätsforschung und eigene Software.
- Reverify ist kein Ersatz für vollständige Decompiler wie Ghidra oder IDA; es verifiziert Behauptungen, stellt aber keine eigene umfassende Analyse dar.

Für wen sich das lohnt

Reverify lohnt sich für Sicherheitsforscher, Malware-Analysten und Entwickler, die KI-Agenten für Binäranalyse einsetzen wollen und dabei Halluzinationen des Modells systematisch ausschließen müssen. Es ist weniger geeignet für Gelegenheitsnutzer ohne Reverse-Engineering-Hintergrund oder für reine Quellcode-Projekte ohne Binärkomponente.

Zusammengestellt von **localdev.ai**
Quelle: **github.com/2akouwu/reverify** · Stand 2026-09-07

Grundlage: README.md, Abschnitt 'Anfang': Projektbeschreibung und Installation; README.md, Abschnitt 'What Reverify does': deterministische Verifikationsschleife; README.md, Abschnitt 'Install the CLI + MCP server from PyPI': Befehle; README.md, Abschnitt 'The verification loop': Claim-Arten und Verifikationslogik; README.md, Abschnitt 'The ledger: state that survives a context reset': Persistenz und Grenzen