

Forum:[Hack Forums](#)

Thread: [Work-In-Progress Decoder Engine](#)

MutatedArm0964 - December 21, 2025, 10:08 pm

Good day/night to you all! I have been working on this for the past 3 weeks, on and off, but nonetheless I have something! Feedback is MUCH appreciated! Please send feedback through GitHub or here.

*IMPORTANT - GitHub link is provided at the bottom. Please read the ReadMe, Code of Conduct, and Contributions tabs!

TLDRSpun a mental globe in my head and my finger landed on "Let's make a decoder!"

Started with ChatGPT >> Helped code a pretty nice script >> Came on here to share and tell people not to use ChatGPT >> After reading comments, I felt stupid and went back to ChatGPT Implemented Gemini >> Gemini annoyed me later on >> Went back to ChatGPT

Asked if there was a site I could use native Linux environment [Too broke to afford a computer/laptop & using a phone] >> Found Github Codespaces (then remembered @mrbumbastik recommended github copilot, thanks btw)

Started using Codespaces >> Fed my code >> Work for next few weeks >> Here I am with a...beta?

Disclaimer

ChatGPT came up with the name. I like it though.

Anyways, below is the description of what I'd like to share.

Titan Decoder Engine (Beta) — Offline-first Python DFIR + malware triage engine

TL;DR

Titan Decoder Engine is an offline-first Python DFIR/malware triage engine for cybersecurity practitioners who need fast, reproducible answers from suspicious payloads and surrounding incident context. It's designed as an engine (not a full EDR/sandbox platform): run it locally or in pipelines to recursively decode/inspect content, extract normalized indicators, correlate against optional IR evidence exports, and generate investigator-friendly outputs.

Core Approach (What It Does)

Recursively "unwraps" data by decoding nested encodings/containers and analyzing extracted artifacts (bounded by depth/count/size/time limits).

Extracts and normalizes IOCs from payloads and derived artifacts (IPs, domains, URLs, emails, hashes, etc.).

Produces a structured, versioned JSON report plus optional exports (IOCs, timelines, graphs, case reports).

Supports optional enrichment (Geo/WHOIS/YARA + optional VT key) with strict offline controls.

Key Capabilities

1) Decoding + Analysis 18 built-in decoders (+ plugin support)

Smart detection: auto-enables format-specific decoders when the input matches

Recursive analysis for multi-stage/nested payloads (configurable depth)

Archive support: ZIP/TAR with anti-zip-bomb protections and extraction limits

Binary analysis: PE/ELF metadata extraction

Built-in decoders: Base64 (and recursive), Base32, Gzip, Bz2, LZMA, Zlib, Hex, XOR, ROT13, URL decode, HTML entities, Unicode escape, UUEncode, ASN.1, QuotedPrintable, PDF, OLE

2) Detection + Risk Rules-as-data detections (upgradeable rule packs) + 7 built-in starter detections:

Deep Base64 nesting

Office macro + network IOCs

LOLBin patterns

Packed/encrypted payload heuristics

Multi-stage infrastructure

XOR + C2 indicators

Malicious PDF patterns

Risk scoring: 0–100 heuristic threat score with levels (CLEAN/LOW/MEDIUM/HIGH/CRITICAL)

Automation-friendly behavior (risk-based exit codes for CI/pipelines)

3) DFIR Evidence Ingestion + Correlation (Optional, Major Focus) Ingests common IR exports (DNS/Proxy/Firewall/VPN/Auth/DHCP) into a canonical Event/Indicator model

Endpoint artifact parsing: PowerShell history (PSReadLine)

Browser history (SQLite: Chrome/Edge History, Firefox places.sqlite)

Evidence-backed pivots: “Top pivots” and “last seen” summaries with provenance (multi-source indicators bubble up)

Evidence links with reason codes + confidence (no black-box correlation)

Evidence timeline export (CSV/JSON) from normalized evidence inputs

4) Reporting + Export Structured report: versioned JSON report for repeatable automation

IOC exports: JSON, CSV, STIX 2.1, MISP

Case reports: Markdown/HTML for investigator handoff

Timeline export (CSV/JSON) for workflows like Timesketch/Excel

Graph export: JSON, Graphviz DOT, Mermaid

5) Workflow / Trust Posture Offline-first mode (--offline) to prevent accidental network enrichment (plus a best-effort local outbound socket kill switch)

Deterministic enrichment caching (SQLite) with refresh control for repeatable runs

Doctor/self-check mode (--doctor) for diagnostics and bug reports

Quiet mode for clean pipelines (--quiet)

JSONL event export for ingestion (--jsonl-out)

Local vault for run history/search (--vault-store, --vault-search)

Secure logging with optional PII redaction

What It's For (Typical Use) Malware triage: unwrap suspicious blobs, inspect decoded stages, extract IOCs, run detections, get a risk score.

Incident scoping: feed DNS/proxy/firewall/VPN/auth/DHCP exports + endpoint artifacts to find pivots/last-seen and produce evidence-backed summaries.

Handoff reporting: generate a structured JSON report and an investigator-friendly case report for teammates or external stakeholders.

What It's Not

Not a full dynamic sandbox/EDR replacement (no endpoint sensor, no full system emulation).

Not a SIEM—though outputs are designed to be ingestible and automation-friendly.

StatusWork in progress / beta: features are functional and tested, but interfaces and report fields may evolve based on feedback.

Links

<https://github.com/pragmaconflux/titan1/releases/tag/v2.0.0-beta.1>

Edit 12/22/25 : I forgot a bunch of stuff to setup on my GitHub like adding a feedback page. It's updated and will be continued to be updated. Sorry for being a newb. Feel free to drag me across the concrete, that way I'll learn better.

Anonymuk - September 24, 2026, 9:51 am

Any screenshots for this? For some reason, the links aren't working