

◆ MEPHIS

• PRIVATE OFFENSIVE SMART-CONTRACT SECURITY

The adversary you retain.

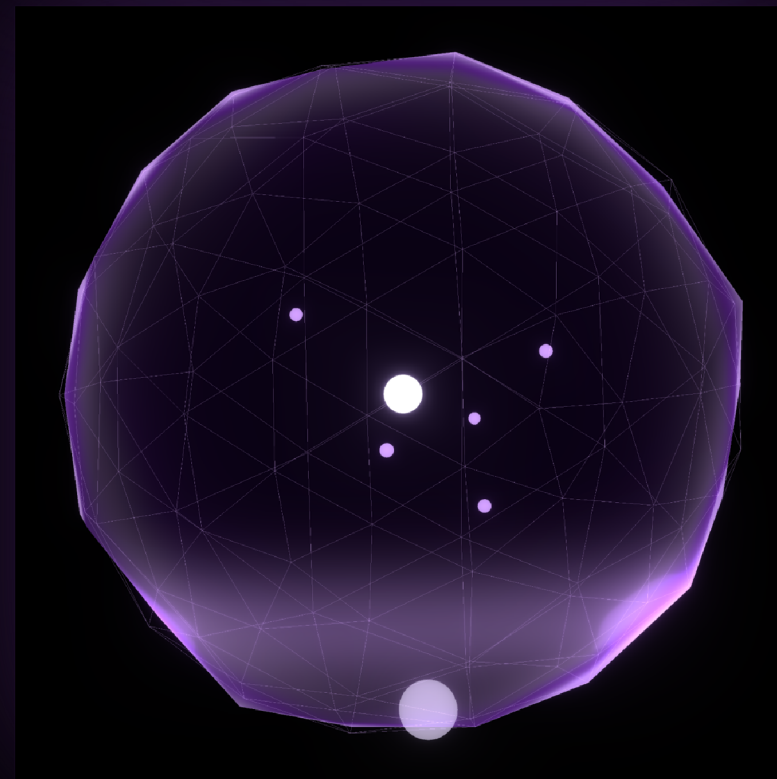
We prove fund-moving paths before attackers do.

A 23-chain autonomous vulnerability-research engine for EVM protocols. It mines live targets, reconstructs attack paths, and reproduces the drain on a pinned mainnet fork, then hands you the exact transaction sequence.

• FORK-ONLY, NO FUNDS BROADCAST • AUTHORIZED TARGETS ONLY

• COMPANY OVERVIEW

© 2026 NVIDIA, the NVIDIA logo, and NVIDIA Inception are trademarks and/or registered trademarks of NVIDIA Corporation in the U.S. and other countries.



Protocols keep getting drained. The audit said it was fine.

Billions

lost to smart-contract exploits every year across DeFi

FIGURE · CITE CHAINALYSIS / DEFI LAMA

Snapshot

a manual audit is point-in-time; code, dependencies and market state move on

STRUCTURAL GAP

Opinion

most reports flag what *might* be exploitable, not what provably is

NO REPRODUCTION

A traditional audit produces a checklist of possibilities. An attacker produces a transaction: one that moves your users' funds. The two artifacts are not the same, and the gap between them is where the money leaves.

Proof, not opinion.

● THE AUDIT YOU BUY TODAY

- × A human reads the code and writes an opinion.
- × Findings are severities on a checklist, not executable.
- × Valid for a single point in time; drifts the moment you ship.
- × You still don't know what an outsider can actually take.

● MEPHIS

- ✓ An autonomous engine writes and runs the exploit against your live code.
- ✓ A finding ships only when value actually moves on a fork of the chain.
- ✓ You get the exact transaction sequence that reproduces it.
- ✓ Nothing is broadcast. It runs on a disposable fork, then reverts.

Low count, high confidence. If it does not reproduce, it does not ship.

An adversarial council writes a bespoke exploit for every target.

● Analyst

TOPOLOGY

Maps the contract, its proxies and money-flow surface into an attack graph.

● Strategist

RL ROUTER

Routes budget to the techniques most likely to pay, from a profile by technique ledger.

● Hunter

JIT EXECUTOR

Spins up the ephemeral fork fleet and runs candidate attacks just in time.

● Composer

EXPLOIT AUTHOR

Writes a bespoke Solidity attacker, compiles it in process, deploys and fires it.

● Divergent

MUTATOR

Escalates and mutates a sound idea instead of abandoning it on the first revert.

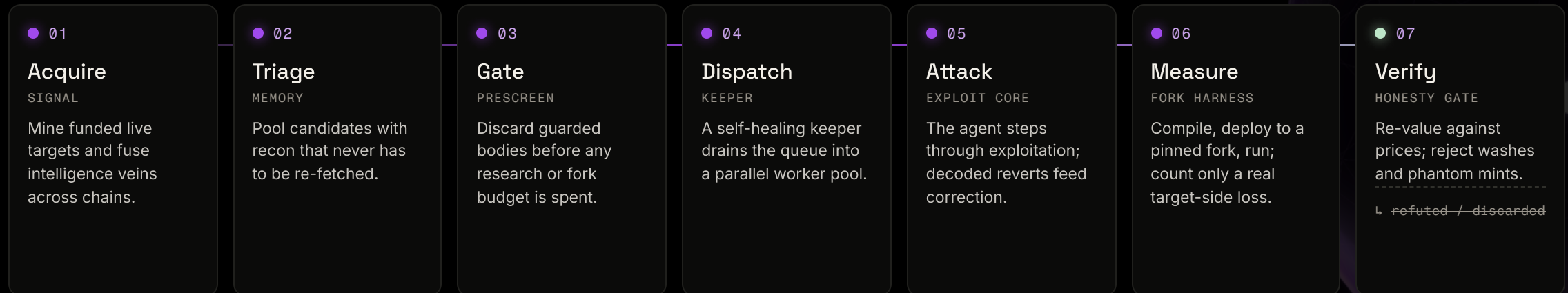
● Critic

ADJUDICATOR

The gatekeeper: only a fork-measured, target-side loss is allowed to count.

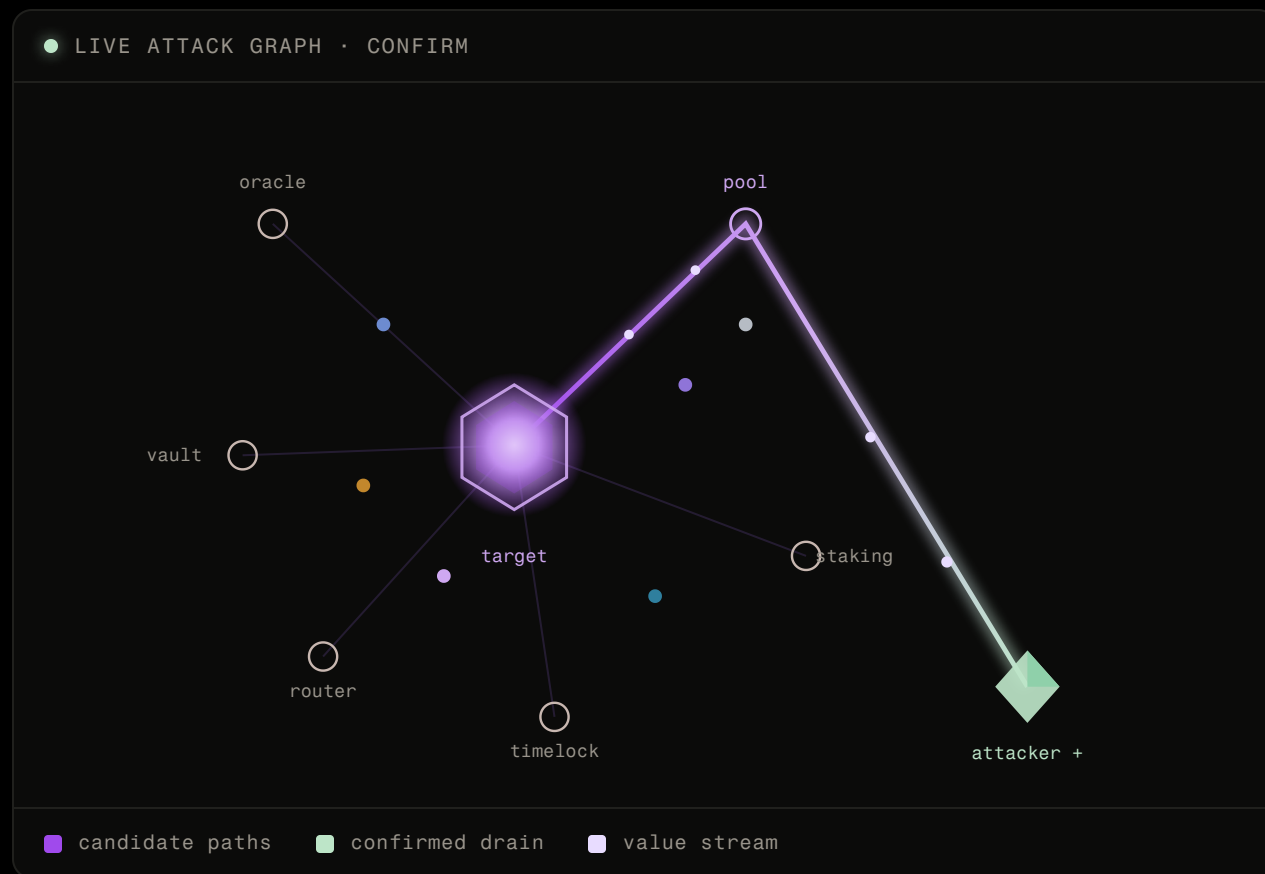
The self-writing exploit: the model proposes, the fork decides. The LLM never self-declares success, so it cannot hallucinate a drain.

One loop, from signal to proof.



■ active flow ■ verified proof ■ refuted / discarded

The council converges on one path, then value moves.



```
./run.sh
```

```
$ ./run.sh
[PASS] victim lost <amount> <asset>
block:  <chain> / pinned
profit: target -, attacker +
wash:   rejected
phantom mint: rejected
```

Self-contained package
report.md · Exploit.sol · PoC.t.sol · foundry.toml ·
run.sh

Reproduces from a clean checkout. No live capital.

The fork is the only source of truth.

● DISPOSABLE PROOF

Every exploit runs on a fresh mainnet-fork snapshot, then the snapshot is reverted. Proving is side-effect-free. It never mutates shared state, and no transaction touches a live chain.

● REAL-VALUE ADJUDICATION

A drain counts only when a target-side contract actually loses value and the attacker gains it, net of dealt capital. Washes, valueless-token mints and pricing artifacts are rejected outright.

● FAIL-CLOSED AUTHORIZATION

A five-gate chain guards any sensitive action and fails closed if any gate does not pass: bounded sessions, an armed kill-switch, explicit approval. Sensitive actions require manual confirmation.

● KEPT HONEST BY CONTROLS

A known-drainable canary must be re-drained on a rotation, and reentrancy and oracle-manipulation positive controls stay in the suite, so a regression in the measurement trips an alarm, not a false headline.

LLMs propose; isolated testnets adjudicate. Reverts are decoded to the exact named guard, so the next attempt knows precisely which check blocked it.

Fork-only by construction. Disclosure by design.

SIMULATE ONLY

No broadcast token is ever granted

The customer flow assembles a reproduction, never live calldata. Bundles carry `broadcastReady: false`, and the disclosure RPC refuses `eth_sendRawTransaction` outright.

TRIAGED

Ranked for the maintainer, P1 to P4

Each finding becomes a fork-only advisory, prioritized by proven-on-fork status, value at risk, confidence and severity, so the owner knows what to fix first.

HUMAN-GATED

Nothing auto-publishes

A separate monitoring swarm produces disclosure-safe advisories behind a human approve or reject gate. Authorized targets only.

The next step after a proof is responsible disclosure to the contract owner, not broadcast.

Twenty-three chains. One unpatched bug travels.

23

EVM chains in discovery, pricing and per-chain fork endpoints

13 FORK-EXECUTABLE IN REPLAY

7

bridge protocol families detected by signature: LayerZero, Wormhole, CCIP, Axelar, Hyperlane and more

100+ DRAIN SELECTORS

10_{min}

fresh-hack replay cadence: decode the attacker tx, find funded twins, fork-prove in the window

RACING DISCLOSURE TO DRAIN

● WORKS ON UNVERIFIED BYTECODE

Attack-surface detection is selector-driven, so it reads contracts with no published source. Proxy look-through resolves the real logic behind minimal, EIP-1967 and UUPS proxies, so proxies are not blind spots.

● SAME CODE, STILL FUNDED, ELSEWHERE

Twin discovery finds the same runtime code (by address, codehash and deployer) across chains, counting a contract only when it is actually funded. Cross-chain infrastructure singletons are denylisted, never blasted.

Point it at a live contract. Get a proof back.

● OPERATOR CONSOLE · LIVE EXPLOIT BOARD

Understandingmapped

Dependencies7 resolved

Hypotheses1 proven · 2 ruled out

Built live from the attacker agent's own variables. Understanding, Dependencies and ranked Hypotheses stream in as the attack runs.

● CUSTOMER REPORT

Target

contract

Verdict

DRAIN REPRODUCED

Value at risk

USD

Method

fork · reverted

Broadcast

none

One renderer, two hosts: the same scan record produces the customer report and the operator report. Portal, console, disclosure funnel and per-contract settings all read one schema.

Prepaid credits. One credit, one proven audit.

STARTER

\$2,000

\$200 / audit

10 credits. One credit runs one full adversarial audit, mapped, forged and proven on a mainnet fork.

GROWTH

\$8,000

\$160 / audit

50 credits. Best fit for teams shipping continuously across multiple contracts and chains.

MOST POPULAR

SCALE

\$25,000

\$125 / audit

200 credits. Best value for protocols and security teams running audits at volume.

Pay on-chain · ETH · USDC · Monero

Self-serve portal + private assessment

Enterprise, continuous-monitoring & disclosure retainer · pricing TBD

Prices shown are current defaults. Land on self-serve credits; expand into continuous monitoring and disclosure retainers.

The tools to prove exploits arrived before the defense caught up.

WHY NOW

LLMs can now author working exploits

Frontier models write and repair bespoke Solidity, the capability step-change that makes autonomous proving real.

TVL at risk keeps compounding

More value, more chains, more integrations, all guarded by the same point-in-time audit model.

Disclosure & regulatory pressure

Protocols increasingly need defensible, reproducible evidence of what an outsider can take, not opinions.

THE MOAT

PROOF HARNESS

A hardened fork adjudicator that rejects the false positives everyone else ships. Trust compounds with every proof.

REUSABLE EXPLOIT MEMORY

Every proven drain becomes a bytecode-fingerprinted recipe: a matching clone is drained deterministically, with no model call.

CROSS-CHAIN SURFACE + PATTERN LIBRARY

Selector-driven detection across 23 chains and a curated attack-pattern library spanning 2016 to 2026.

A working engine, not a deck-ware diagram.

23

chains wired for discovery, pricing and per-chain forking

VERIFIED IN CODE

20

curated real-world attack patterns, 2016 TheDAO to 2026

VERIFIED IN CODE

6

specialist agents, a live exploit console and one shared report renderer

VERIFIED IN CODE

● THE PROOFS LEDGER

Every reproduced drain is recorded and tiered (real attacker-feasible, historical, or legacy) and browsable in an operator proofs panel. The live count is the metric that matters.

proven drains · live figure

value-at-risk proven · live figure

● GO-TO-MARKET

design partners / pilots · count

audits run to date · count

revenue / pipeline · figure

Operational figures are tracked live on the platform. Fill from the current dashboard before sending.

• THE TEAM

The team behind the engine.

Security, AI, and capital — from architecture to market.



Eric Schulman

COFOUNDER & CTO

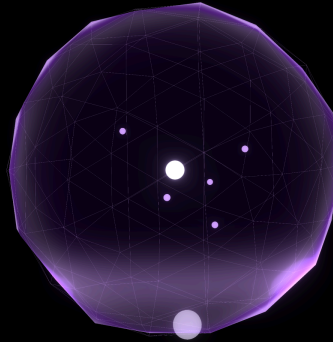
MIT (AGI/NLP research). Prior Head of Blockchain Security at **Quantinium** (L1 hardening) and **Catalyx** (digital-asset exchange). 20+ years in reverse engineering, exploit research, and autonomous AI agents. Built the Mephis engine, product, and infra end to end.



Addiel Lopez

COFOUNDER & CEO

Michigan State University. Founding Partner at Standard Fund (private equity). Scaled Quantinium's DePIN infrastructure and go-to-market. Deep experience in digital infrastructure, data centers, and PropTech investment.



• THE ASK

The adversary you retain.

We prove fund-moving paths before attackers do, on a fork, with the money actually moving, and never on a live chain. Help us put a proving engine in front of every protocol that has value to lose.

RAISING

Pre-Seed · \$250K–\$1M

TEAM

Eric Schulman, Cofounder & CTO.
Addiel Lopez, Cofounder & CEO. (See Team.)

CONTACT

Eric · eas@mephistopheles.zip