HACK KRMU 5.0

- PROBLEM STATEMENT ID : PRE-EXECUTION SIMULATION ENGINE WITH SHADOW ENVIRONMENT
- TEAM NAME : 0DAY
- TEAM ID : HK- 005
- TEAM MEMBERS : HARSH DEV SINGH
  ASHISH ATTRI
  MANSIMAR SINGH

PROBLEM & SOLUTION

Blockchain transactions are final and irreversible. A malicious smart contract can drain funds, seize control, manipulate protocols, or permanently disable contracts.
The main issue is the lack of a mandatory pre-execution security layer, allowing risky transactions to execute without prior validation.
Critical Threats Mitigated:

- Rug pulls and sudden liquidity withdrawal
- Reentrancy-based fund draining
- Flash loan price manipulation
- Malicious governance takeovers
- Hidden ownership escalation
- SELFDESTRUCT contract elimination
- Proxy upgrade abuse and contract hijacking

ShadowGuard is a Pre-Execution Security Proxy that checks blockchain transactions before they are finalized. It intercepts a transaction, simulates it in a safe shadow environment, analyzes possible changes and risks, assigns a risk score, and then decides whether to allow or block it — preventing damage before it happens.
This helps reduce fraud, smart contract exploits, and accidental losses.
It acts like a firewall layer, adding extra security before transactions reach the blockchain.

FLOW OF SOLUTION

TECH STACK & APPROACH

Tech Stack

Blockchain Network
- Ethereum Sepolia Testnet (Chain ID: 11155111)
- EVM Compatible (Sepolia RPC)

Simulation Engine
- Python 3 & web3.py
- eth_call Simulation
- eth_estimateGas Analysis

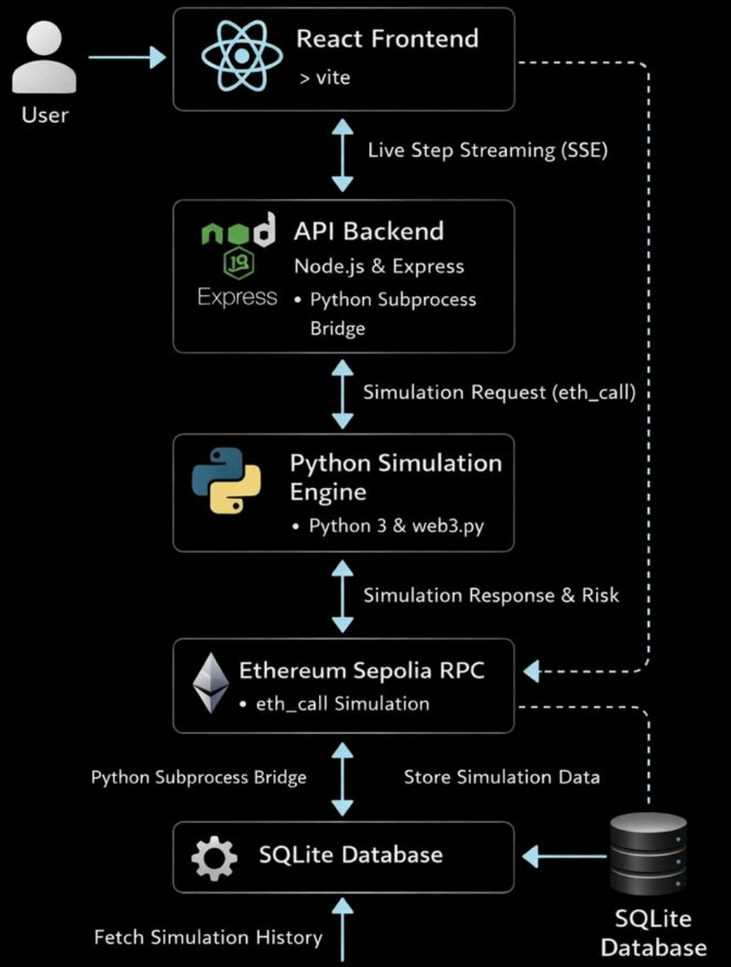
API Backend
- Node.js & Express.js
- Python Subprocess Bridge
- Server-Sent Events (SSE)

Frontend
- React 18 & Vite
- Real-Time Simulation UI
- Brutalist Design

Database
- SQLite Database
- Store Simulations & Policies

Dev Tools
- Concurrently Runner
- Docker-Ready Setup

Architecture



UNIQUENESS & INNOVATION FACTOR

Existing Tools

- **Slither / MythX → Static analysis**
- **Tenderly → Simulation only**
- **Forta → Post-execution monitoring**
- **OpenZeppelin Defender → Alerts**

None provide:

- **Pre-execution enforcement**
- **State mutation diffing**
- **Policy-based blocking**

What Makes ShadowGuard Different

- ✓ Real-time transaction interception
- ✓ Deterministic shadow execution
- ✓ State mutation diff engine
- ✓ Opcode & behavioral analysis
- ✓ Explainable weighted risk scoring
- ✓ Configurable security policies
- ✓ Middleware-level integration

FEASIBILITY & CHALLENGES

Feasibility

- **Built on Ethereum Sepolia (EVM Compatible)**
- **Uses deterministic eth_call simulation**
- **No blockchain modification required**
- **Uses standard RPC infrastructure**
- **Modular Python-based simulation engine**
- **Real-time SSE streaming architecture**
- **SQLite logging for traceability**
- **Docker-ready deployment**
- **Technically viable using existing EVM capabilities.**

Challenges & Mitigation

1. Simulation Latency (1–3 sec)
   → Prioritize high-value transactions
   → Async queue scaling
2. False Positives
   → Configurable policy engine
   → Adjustable risk thresholds
3. Compute Overhead
   → Bytecode caching
   → Horizontal scaling
4. Multi-Chain Expansion
   → EVM adapter-based architecture

RESEARCH & REFERENCE

- Chainalysis Crypto Crime Reports – https://www.chainalysis.com/blog/crypto-hacking-stolen-funds-2024/
- DeFiLlama Analytics – https://defillama.com/
- Immunefi Research – https://immunefi.com/research/
- Halborn Security Reports – https://halborn.com/
- Tenderly Documentation – https://tenderly.co/
- Forta Monitoring – https://forta.org/
- Ethereum Yellow Paper – https://ethereum.github.io/yellowpaper/paper.pdf
- web3.py Documentation – https://web3py.readthedocs.io/
- Polygon Documentation – https://polygon.technology/
- Research Methodology: Analysis of DeFi exploit case studies, industry reports, technical documentation review, and feasibility validation through Polygon testnet simulation.