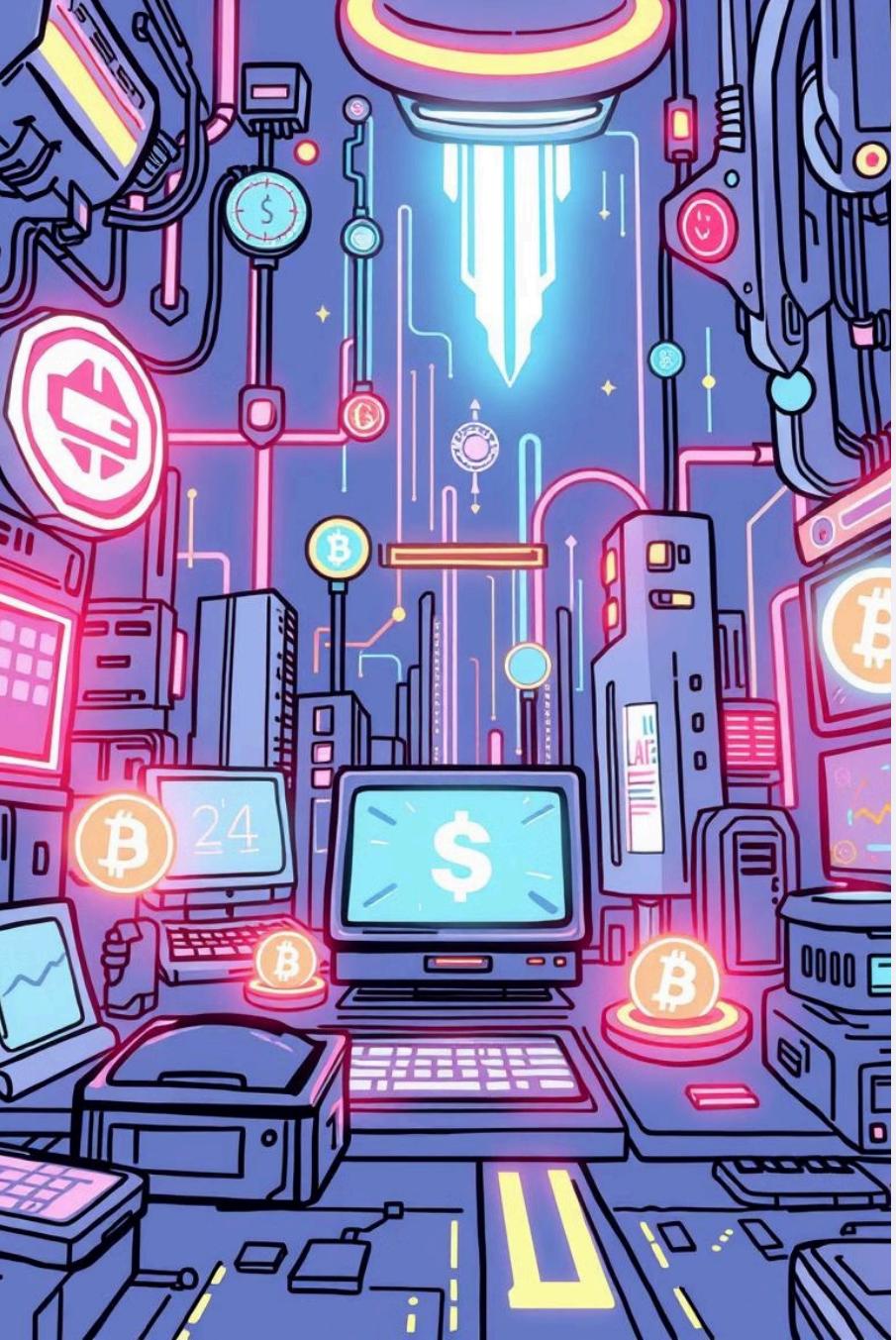


Smart Contract Vulnerabilities: Hacking, Detecting, and Preventing

A deep dive into the world of smart contract vulnerabilities. Learn how to hack, detect, and prevent exploits//

Jayavamsi Villuri - CEH Master, Software Engineer, Wannabe Security Engineer / Blockchain Enthusiast

https://velofjay.netlify.app | LinkedIn | X | Email

Contents

Terminology

Smart Contract

Self-executing agreement on the blockchain.

Gas

Unit of computation cost in Ethereum.

Solidity

Primary programming language for Ethereum contracts.

EVM

Ethereum Virtual Machine - the runtime environment.

Game On: Crack Smart Contracts with Ethernaut

1 **Hack Your Way to Mastery**

Game-based security puzzles for Ethereum smart contracts

2 **Exploit to Protect**

Practice exploiting common vulnerabilities in a controlled environment

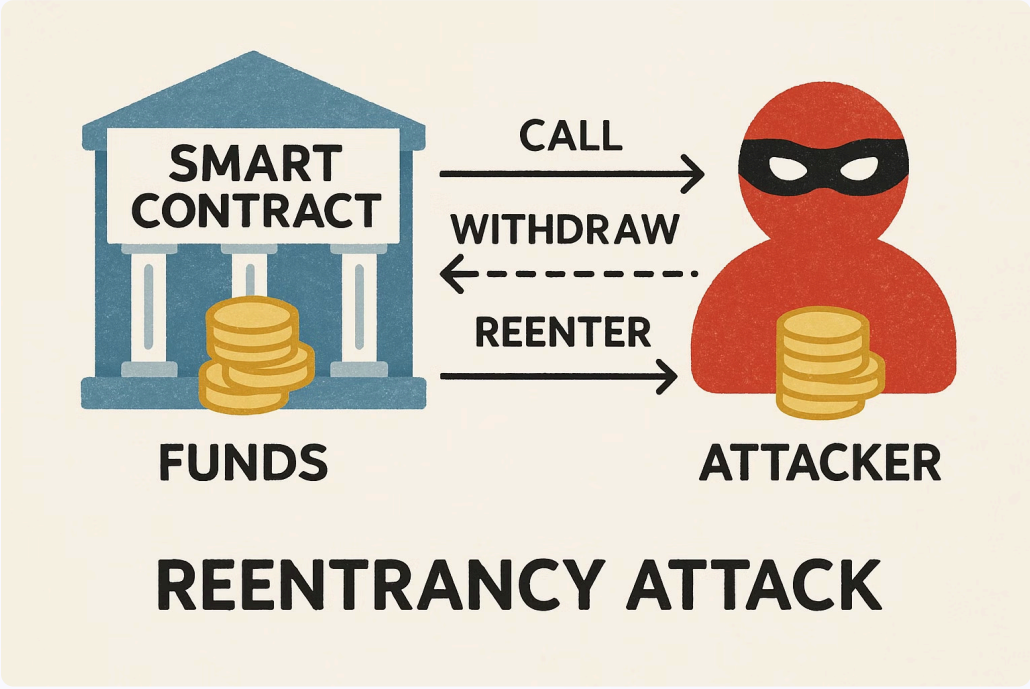
https://ethernaut.openzeppelin.com/ | Ethernaut-Setup

Vulnerability #1: Reentrancy

Malicious contract recursively calls a vulnerable function.

Occurs before state updates.

Phone call interrupting withdrawal allows repeated withdrawals.



Vulnerability #2: Integer Overflow/Underflow

Calculations exceeding or falling below the representable range. Results in unexpected behavior.

A counter resetting to its maximum value after reaching zero.

Vulnerability #3: Timestamp Dependence

Critical logic relies on block timestamps.

Miners can manipulate timestamps.

Gambling game outcome based on current block timestamp.

Vulnerability #4: Gas Limit & Loops

Unbounded loops exhaust gas limits.

Contract execution halts.

Infinite loop causes a program to freeze.

Vulnerability #5: Front Running

Observing pending transactions for profit.

Executing a more profitable one before it.

Trader seeing another's order and placing their own order ahead.

Detection & Prevention

Detection

- Static analysis (**Slither** / **Mythril**)
- Formal verification
- Fuzzing

Prevention

- Secure coding
- Audits
- Bug bounties