# Reversing Ethereum Smart Contracts

@f0rki

2018-03-28

# WTF? Ethereum?

- Cryptocurrency
- PoW blockchain (Nakamoto consesus)
  - Work on Proof-of-Stake
- **Smart Contracts** on the blockchain

# The Ethereum Blockchain

- Actually: one huge distributed computer
- Consesus protocol to fix one state of the computer
- Smart contracts
  - A program on the ethereum computer
  - Can modify the state of the blockchain
  - Interactions between contracts

# EVM Internals

- Specification "Yellow Paper"
- Opcode Info trailofbits/evm-opcodes
- Existing ethereum implementations
  go-ethereum/core/vm/instructions.go

# Ethereum Virtual Machine I

- A bit weird...

- Turing-complete
- Stack-machine
- 1-byte opcodes
- "Gas" limits resources
  - Number of executed instructions
  - Used memory

# Ethereum Virtual Machine II

Add 2 numbers:

```
1                                   // stack: []
2  6001    | PUSH1 0x1             // stack: [0x1]
3  614242  | PUSH2 0x4242          // stack: [0x1, 0x4242]
4  01      | ADD                   // stack: [0x4243]
```

# Ethereum Virtual Machine III

- Harvard-architecture
- Separate address spaces for
  - Code (`JUMP` / `JUMPI` / `PC`)
  - Stack (`PUSH` / `POP`)
  - Memory (`MLOAD` / `MSTORE`)
  - Storage (`SLOAD` / `SSTORE`)
- Storage is persistent state

# Ethereum Virtual Machine IV

- A `CALL` triggers a transaction
  - Usually to another contract
  - A little bit like RPC
  - $\rightarrow$ new stack, memory areas
- Transactions can be rolled back, by
  - Out-of-gas exception
  - `INVALID`, `REVERT`

# Ethereum Virtual Machine V

- Code starts at address $0$
- There is only one function
- Solidity introduced dispatcher
  - First 4 bytes of input are magic value
  - Dispatcher jump to different "functions"
  - Huge if-else on magic values
- Solidity ABI definitions are metadata for "functions"
  - Parameter types
  - Method name
  - Modifiers (such as `payable`)

# EVM Contract Creation

- Transaction to address $0$
  - Contract code
  - Constructor parameters
- Calls constructor
- Return value is "runtime" code

# Reversing Tools

- IDA with github.com/trailofbits/ida-evm
- binary.ninja with github.com/trailofbits/ethersplay
- radare2 with evm plugin
  - `r2pm install evm`
- github.com/Arachnid/evmdis (standalone)
  - `go get Arachnid/evmdis`

- Symbolic execution tools:
  - github.com/trailofbits/manticore
  - github.com/melonproject/oyente
  - github.com/ConsenSys/mythril

# Challenge Time!
# `smarties` on fuzzy.land