Blockchain Autopsies: Analyzing Ethereum Smart Contract Deaths

Jay Little

Blackhat USA 2018

August 6, 2018

Jay Little

Principal Security Engineer

@computerality

Favorite IDA Pro keyboard shortcuts : Y and D

Working with Smart Contracts >1 year

Trail of Bits

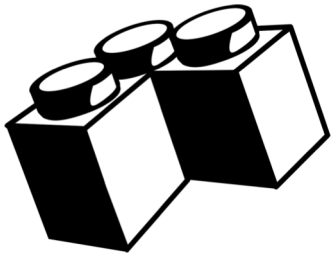
Cyber security research company - High-end security research with a real-world attacker mentality to reduce risk and fortify code.

Security Research

- As a leading cybersecurity research provider to DARPA, the Army and the Navy – we create and release open source research tools

Security Engineering

- We offer custom engineering for every stage of software creation, from initial planning to enhancing the security of completed works



Security Assessments

- We offer security auditing for code and systems requiring extreme robustness and niche system expertise

Agenda

- Introduction to Ethereum, EVM, and Solidity
- Vulnerabilities and Reversing Tools
- Ethereum Node Software
- Blockchain Contract Trace Analysis
- Analyze Contract Deaths

Prompt

```
eth.getCode(0xcD6D2cD79fD754C6B909585E46541D32ec491962)
> 0x
```

- Where did the code go?
- Who created this contract?
- What was the last transaction to this contract?
- Why are articles only about open-source contracts?
- When did all of this happen?

Ethereum, EVM, and Solidity

Ethereum

- A blockchain based distributed ledger
- A “world computer” with “smart” contracts
- The 2nd largest cryptocurrency by valuation
- Mainnet started July 30 2015

$463.02 USD **(-1.11%)**

0.05705910 BTC (0.21%)

Market Cap	Volume (24h)	Circulating Supply
$46,765,782,741 USD 5,763,088 BTC	$1,633,000,000 USD 201,239 BTC	101,002,084 ETH

https://coinmarketcap.com/currencies/ethereum/

Ethereum Implementation

- **Ethereum is formally described by the Yellow Paper**
  - https://github.com/ethereum/yellowpaper
  - https://github.com/chronaeon/beigepaper

ETHEREUM: A SECURE DECENTRALISED GENERALISED TRANSACTION LEDGER

BYZANTIUM VERSION c0444c1 - 2018-03-06

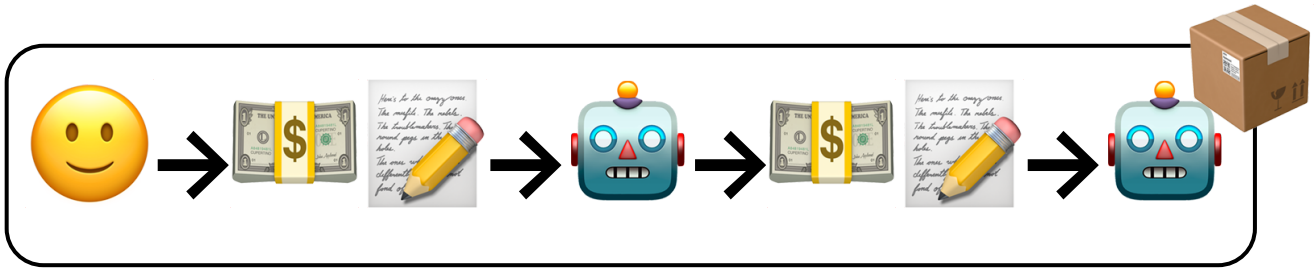
DR. GAVIN WOOD
FOUNDER, ETHEREUM & PARITY
GAVIN@PARITY.IO

ABSTRACT. The blockchain paradigm when coupled with cryptographically-secured transactions has demonstrated its utility through a number of projects, with Bitcoin being one of the most notable ones. Each such project can be seen as a simple application on a decentralised, but singleton, compute resource. We can call this paradigm a transactional singleton machine with shared-state.

Ethereum implements this paradigm in a generalised manner. Furthermore it provides a plurality of such resources, each with a distinct state and operating code but able to interact through a message-passing framework with others. We discuss its design, implementation issues, the opportunities it provides and the future hurdles we envisage.

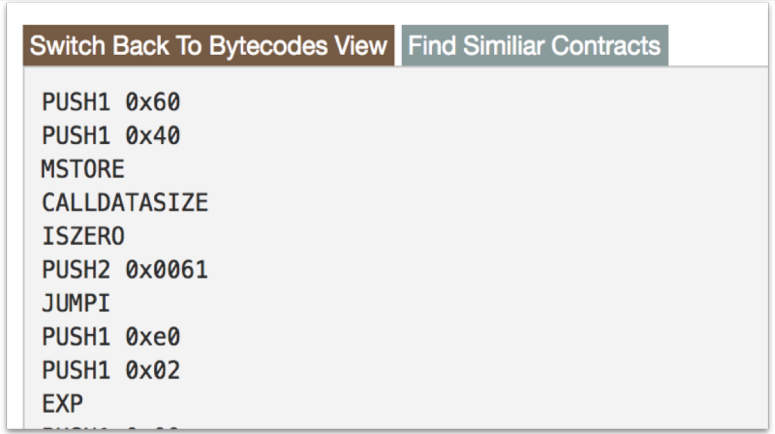
Accounts and Transactions and Blocks

- Account: 🙂
- Contract: 🤖
- 1 Ether (ETH) = 10^{18} Wei
- 21000 Wei per TX
- Contracts can call other contracts



EVM: Ethereum Virtual Machine

- Big Endian stack machine
- ~185 opcodes
- Native data width is 256 bits
- Many instructions are similar
  - PUSH1 – PUSH32
  - DUP1 – DUP16
  - SWAP1 – SWAP16
- Instructions have various gas cost
- ethervm.io or https://github.com/trailofbits/evm-opcodes



ABI and Address Spaces

- EVM is a Harvard architecture
- There are ~5 address spaces
- Storage and memory are 256-bit address space
- All execution enters at PC=0x0
- Jump destinations labeled with JUMPDEST
- Functions dispatched based on first 4 bytes in TX input

Code	EVM, implements contract logic
Stack	Limited to 32 elements
Call Data	Invocation arguments
Memory	Non-persistent storage, per tx
Storage	Persistent storage

Solidity

- JavaScript-inspired high-level language for smart contracts
- Compiles to EVM

```
contract NiceGuyTax {

    // Make a database of investors.
    struct Investor {
      address addr;
    }
    Investor[] public investors;

    // Make a database of Nice Guys.
    struct NiceGuy {
      address addr;
    }
    NiceGuy[] public niceGuys;

    //Counters. this counts things. A new round begins when investorIndex reaches 10.
    uint public payoutIndex = 0;
    uint public currentNiceGuyIndex = 0;
```

Sample Contract

```
contract CookieShop {
    address owner;
    mapping (address=>uint) public jar;

    constructor() public  {
        owner = msg.sender;
    }

    function bake() public payable {
        if(msg.value > 0.1 ether) {
            jar[msg.sender] += 13;
        }
    }

    function eat(uint count) public {
        jar[msg.sender] -= count;
    }

    function close() public {
        require(msg.sender == owner);
        selfdestruct(msg.sender);
    }
}
```

Sample Contract Creation

```
contract CookieShop {
    address owner;
    mapping (address=>uint) public jar;

    constructor() public  {
        owner = msg.sender;
    }

    function bake() public payable {
        if(msg.value > 0.1 ether) {
            jar[msg.sender] += 13;
        }
    }

    function eat(uint count) public {
        jar[msg.sender] -= count;
    }

    function close() public {
        require(msg.sender == owner);
        selfdestruct(msg.sender);
    }
}
```

 →  → 0x0

 owner: 

jar[]

Sample Contract Death

```
contract CookieShop {
    address owner;
    mapping (address=>uint) public jar;

    constructor() public  {
        owner = msg.sender;
    }

    function bake() public payable {
        if(msg.value > 0.1 ether) {
            jar[msg.sender] += 13;
        }
    }

    function eat(uint count) public {
        jar[msg.sender] -= count;
    }

    function close() public {
        require(msg.sender == owner);
        selfdestruct(msg.sender);
    }
}
```

close() = 0x43d726d6

👨‍🍳 → close() → 🤖

🤖 → 💵 → 👨‍🍳

🤖 = 0x

owner:[]

jar[]

Sample Contract Usage

```
contract CookieShop {
    address owner;
    mapping (address=>uint) public jar;

    constructor() public  {
        owner = msg.sender;
    }

    function bake() public payable {
        if(msg.value > 0.1 ether) {
            jar[msg.sender] += 13;
        }
    }

    function eat(uint count) public {
        jar[msg.sender] -= count;
    }

    function close() public {
        require(msg.sender == owner);
        selfdestruct(msg.sender);
    }
}
```

bake() = 0xb0de262e

👷 → bake() → 🤖

👷 → bake() → 🤖

🏋️ → bake() → 🤖

🤖 owner:👨‍🍳

jar[🏋️]=🍪🍪

jar[👷]=🍪🍪🍪🍪

Sample Contract Usage (2)

```
contract CookieShop {
    address owner;
    mapping (address=>uint) public jar;

    constructor() public  {
        owner = msg.sender;
    }

    function bake() public payable {
        if(msg.value > 0.1 ether) {
            jar[msg.sender] += 13;
        }
    }

    function eat(uint count) public {
        jar[msg.sender] -= count;
    }

    function close() public {
        require(msg.sender == owner);
        selfdestruct(msg.sender);
    }
}
```

CookieShop.eat(5) =
0x85e0ebaf00000000000000000000000000000000000000000000000000000000000000000000000000000005

🏋️ → eat(5) → 🤖

🤖 owner: 👨🏻‍🍳

jar[🏋️]=🍪

jar[🥷]=🍪🍪🍪🍪

Sample Contract Usage (3)

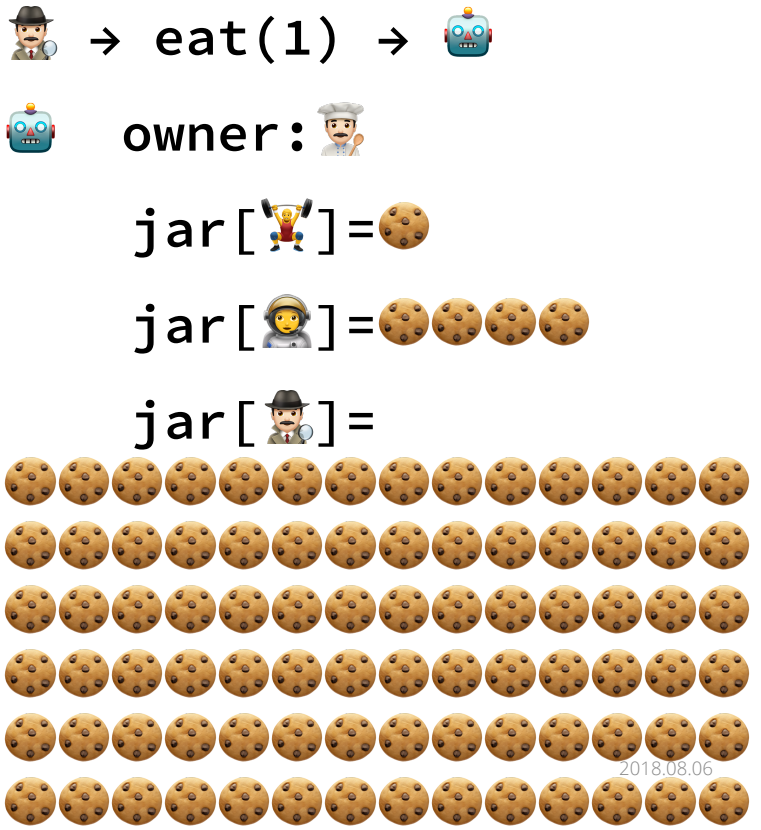
```
contract CookieShop {
    address owner;
    mapping (address=>uint) public jar;

    constructor() public  {
        owner = msg.sender;
    }

    function bake() public payable {
        if(msg.value > 0.1 ether) {
            jar[msg.sender] += 13;
        }
    }

    function eat(uint count) public {
        jar[msg.sender] -= count;
    }

    function close() public {
        require(msg.sender == owner);
        selfdestruct(msg.sender);
    }
}
```



Features / Vulnerabilities

Solidity Behaviors and Issues

- Int overflow/underflow
- Incomplete initialization
- Uninitialized variables
- Callbacks / re-entrancy
- Variable name shadowing
- Type inference (var)
- Unintentional visibility
- Array.length
- Delegatecall
- Division by zero
- Race conditions
- Replay attacks
- Bad RNG
- Time sensitivity
- Blockchain as random

Uninitialized Variables

```
contract OpenAddressLottery{      0x741F1923974464eFd0Aa70e77800BA5d9ed18902
    struct SeedComponents {  https://www.reddit.com/r/ethdev/comments/7wp363
        uint component1;
        uint component2;
    }

    address owner; //address of the owner
    uint private secretSeed; //seed used to calculate number of an address

    function forceReseed() {
        require(msg.sender==owner);
        SeedComponents s;
        s.component1 = uint(msg.sender);
        s.component2 = uint256(block.blockhash(block.number - 1));
    }

}
```

Not So Smart Contracts

https://github.com/trailofbits/not-so-smart-contracts

(Not So) Smart Contracts

This repository contains examples of common Ethereum smart contract vulnerabilities, including code from real smart contracts.

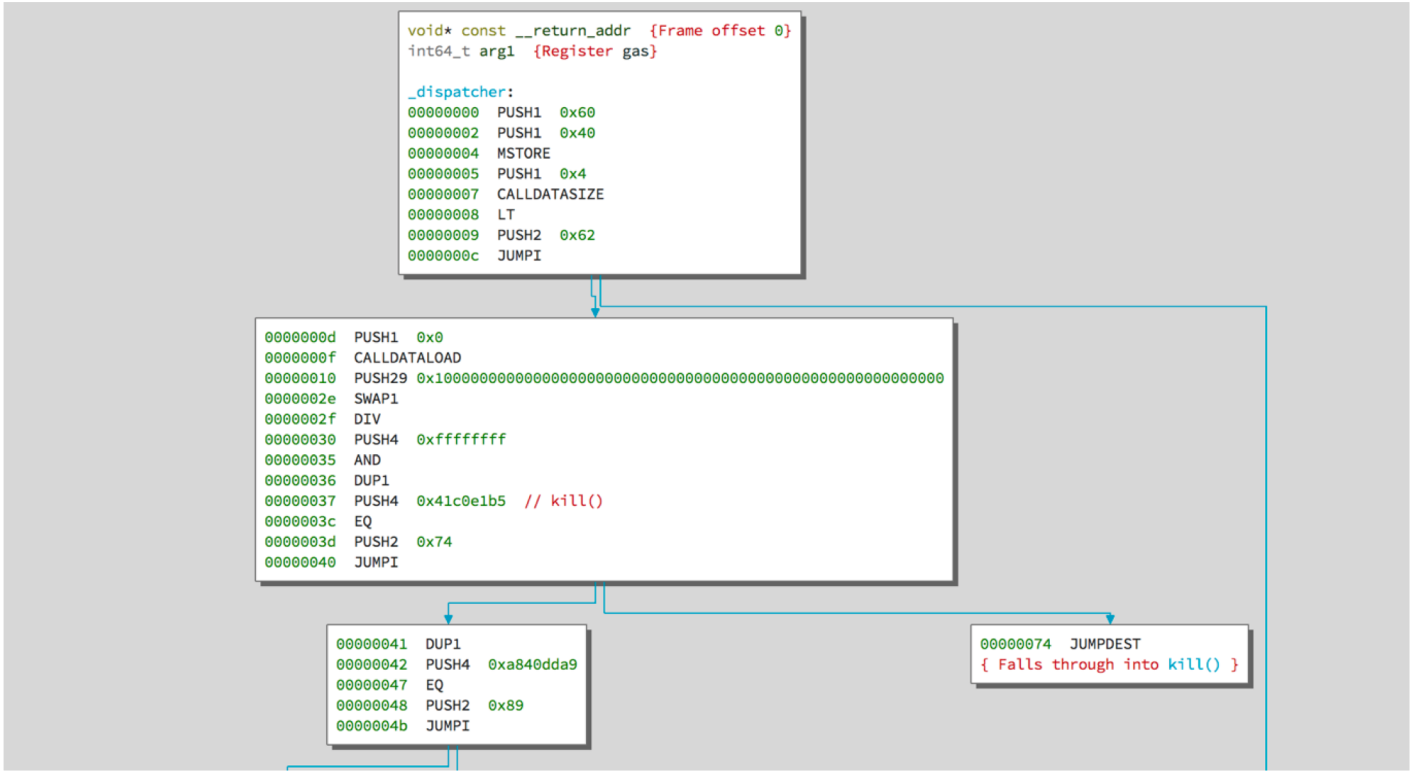
Vulnerabilities

- Integer Overflow
- Missing Constructor
- Reentrancy
- Unchecked External Call
- Unprotected Function
- Incorrect Interface

Analysis Tools

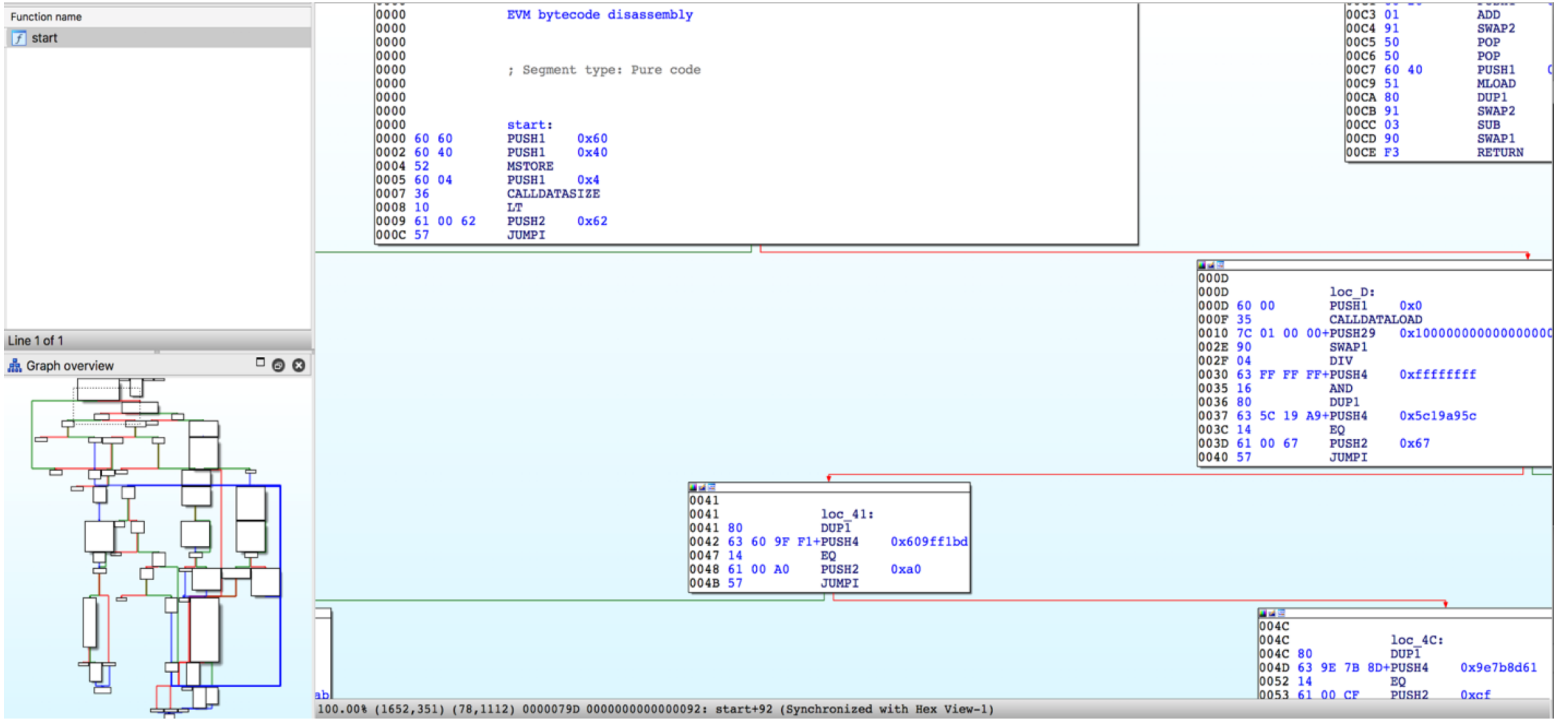
Ethersplay

Binary Ninja Plugin https://github.com/trailofbits/ethersplay



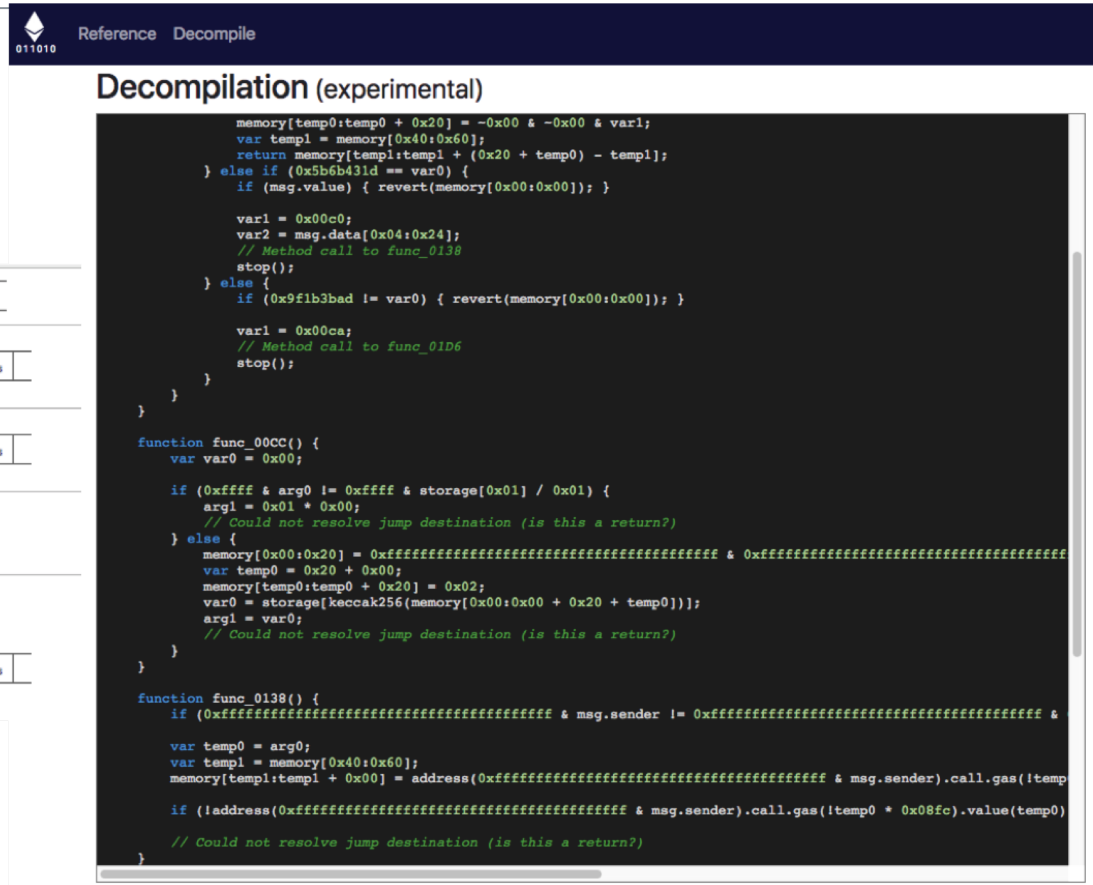
IDA-EVM

IDA Pro Module https://github.com/trailofbits/ida-evm



ethervm.io

F0	CREATE	value, offset, length	addr
F1	CALL	addr, value, argsOffset, argsLength, retOffset, retLength	success
F2	CALLCODE	addr, value, argsOffset, argsLength, retOffset, retLength	success
F3	RETURN	offset, length	-
F4	DELEGATECALL	addr, argsOffset, argsLength, retOffset, retLength	success



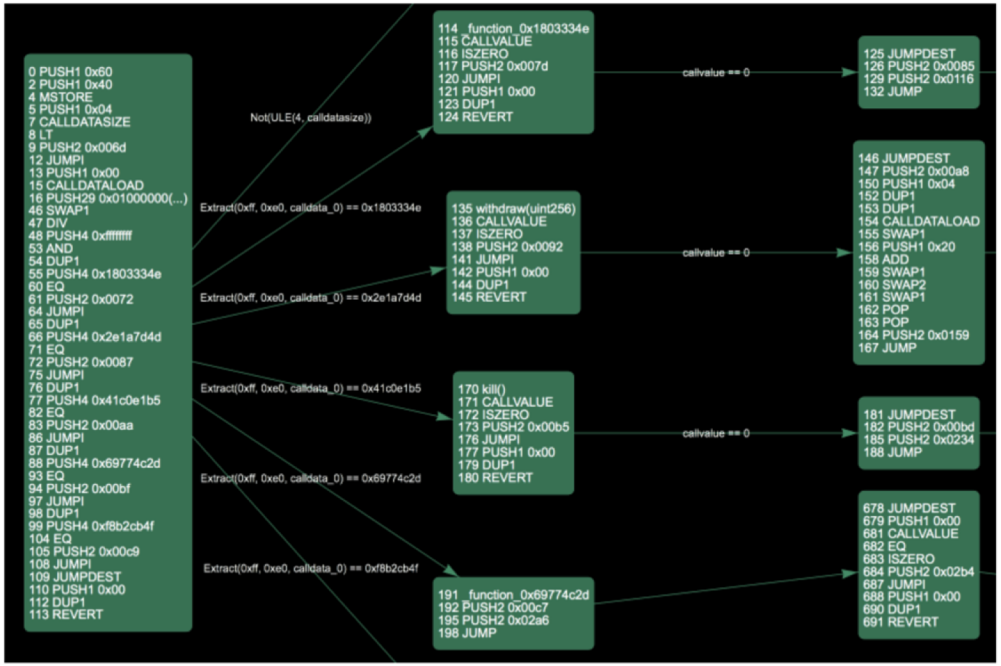
Mythril

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)
```



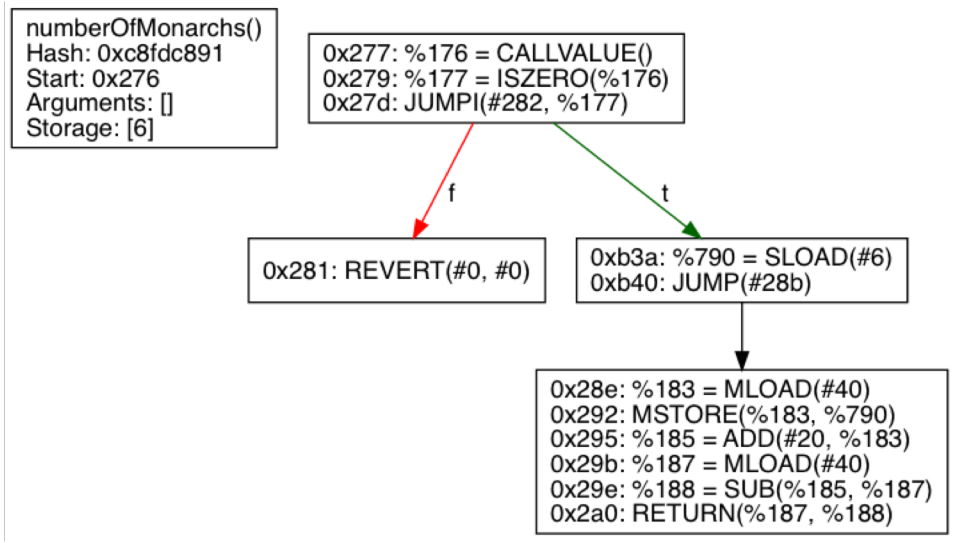

```
$ myth --search "func#changeMultisig(address)#"
$ myth --search "code#PUSH1 0x50,POP#"
$ myth --search "func#changeMultisig(address)# and code#PUSH1 0x50#"
```

Rattle

https://github.com/trailofbits/rattle

Released this morning!

- Recovers EVM Control Flow
- Lifts EVM to IR to SSA IR
- Optimizes and simplifies
- Recovers variables
- Generates function CFGs



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
```

Ethereum Software

Storage Requirements

Check stackexchange first.

Client / Mode	Block Number	Disk Space
geth light	5_600_000	363M
geth fast full	5_600_000	142G
geth full full	?_???_??? [1]	239G + [1]
geth full archive	4_980_000 [2]	671G

- [1] My disk was full, I didn't expect this to run out of space and wasn't able to repeat this sync mode
- [2] I didn't manage to fully sync the archive node within 6 weeks, unfortunately.

Client / Mode	Block Number	Disk Space
parity light	5_600_000	89M
parity warp fast	5_600_000	82G
parity full fast	5_600_000	78G
parity full archive	5_600_000	1.1T

- Parity 1.10.0
- Ubuntu 16.4 LTS, VPS instance with SSD backed storage

Process Name	Bytes Written	Bytes Read
parity	19.17 TB	12.79 TB

From @5chdn at https://wiki.parity.io/faq and https://ethereum.stackexchange.com/questions/143/what-are-the-ethereum-disk-space-needs

Geth and Parity

Geth - official implementation, runs 75% public nodes
    Written in Go/LevelDB
Parity - alternate implementation, runs 15% public nodes
    Written in Rust/RocksDB

Client Type	Size	Time	Details
Full	100GB-1.5TB	~ Weeks to Forever	Large SSD, can fetch any TX
Fast	50-200GB	~ Hours to Days	SSD, Recent TX only
Light	50MB+	~ Minutes to Hours	HDD, Intended for “end user”

Geth Running Options

```
./geth --datadir /mnt/fastssd/.geth
 --rpc --rpcapi=debug,eth,net,rpc,web3
 --syncmode=full
 --gcmode=archive
 --cache 4096
 --trie-cache-gens 1024
```

Parity Running Options

```
parity -d /mnt/fastssd/.parity
 --jsonrpc-apis web3,eth,net,parity,rpc,traces
 --mode=active
 --pruning=archive
 --tracing=on --fat-db=on
 --min-peers=50 --max-peers=100
 --cache-size=4096
 --db-compaction=ssd
 --tx-queue-size=8192000
 --scale-verifiers --num-verifiers=8
 --jsonrpc-server-threads 4 --jsonrpc-threads 8
```

Client Operation Suggestions

- Have patience
- Troubleshoot with `rm -rf` and resync
- Use Linux
- Use the fastest SSDs you can afford
- Ethereum clients and web browsing don’t mix

Many Days Later...

Contract Analysis

Answering Questions

```
eth.getCode(0xcD6D2cD79fD754C6B909585E46541D32ec491962)
> 0x
```

- **Who created this contract?**
- What was the last transaction to this contract?
- Where did the code go?
- When did all of this happen?

Tracing

Parity:

```
trace_replayTransaction(tx_hash, [‘trace’])
```

Geth:

```
debug_traceTransaction(tx_hash)
```

Transactions


```
{'blockNumber': '5269390',
 'from': '0xcd6d2cd79fd754c6b909585e46541d32ec491962',
 'hash':
'0x9ebcb287709403cb4c11d6c82203cfee0428b1dda72b417a65d2ba120fc70947',
 'isError': '0',
 'timeStamp': '1521260221',
 'to': '0x3f9ed84ef180fae940ebf4bce4c4d70e2f751482',
 'type': 'suicide',
 'value': '2982279810000000000'}
```

Who? What? When?

Block: 5245655
From: 0x00bb585e7be7b095be9aba3c5777121c5ba7924a
To: 0
[Contract 0xcd6d2cd79fd754c6b909585e46541d32ec491962 Created]

0x00bb585e7be7b095be9aba3c5777121c5ba7924a Adds 0.2 Ether

0x3f9ed84ef180fae940ebf4bce4c4d70e2f751482: 0xa840dda9

0x3f9ed84ef180fae940ebf4bce4c4d70e2f751482: kill()

0xcd6d2cd79fd754c6b909585e46541d32ec491962 => selfdestruct
0x3f9ed84ef180fae940ebf4bce4c4d70e2f751482

Scanning the Blockchain

Blockchain Data

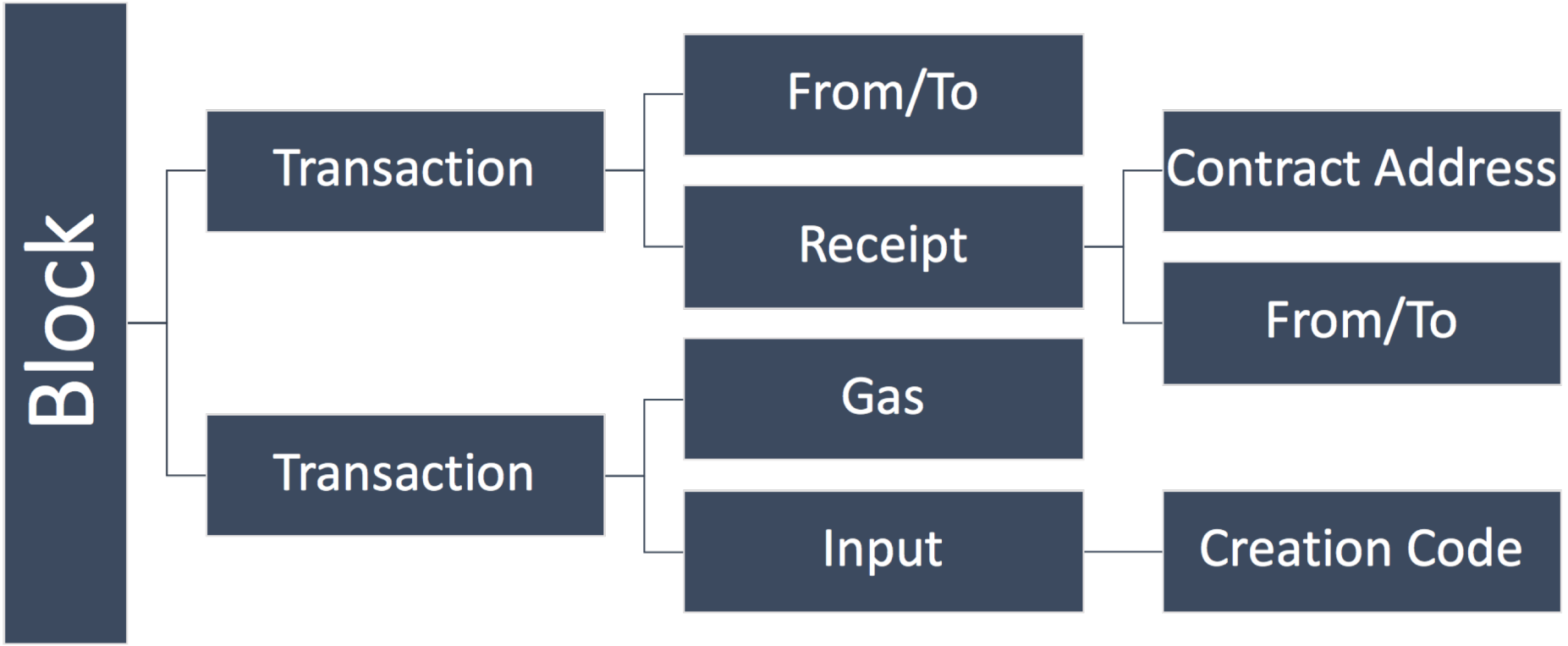
Distributed Ledger != Distributed Database

Only a key/value store with list of blocks/transactions

No queryable structure

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

The Block in Blockchain



web3.js and web3.py

web3.js is official client library

- Many API changes between v0.20 and v1.0

web3.py is Python implementation of web3.js

- Version 4.0 switched to Python3.5+

Both communicate to Ethereum nodes via:

- IPC - use when local
- WebSockets - use when streaming events
- RPC - use in any other situation

Finding Contracts

```
for b in range(0, 6000000):
  block = w.eth.getBlock(b, full_transactions=True)
  for tx in block.transactions:
    if tx['to'] == None:
      r = w.eth.getTransactionReceipt(tx['hash'])
      address = r['contractAddress']
      if address:
          code = w.eth.getCode(address)
          if code == '0x' and r['status'] == 1:
              saveContract(block, address, tx['input'])
```

Geth Experience

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 f
#16126 opened 17 days ago by d

```
Upgrading chain index
Imported new chain segment
Imported new chain segment
Upgrading chain index
Upgrading chain index
Imported new chain segment
Upgrading chain index
Upgrading chain index
Upgrading chain index
Upgrading chain index
```

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

```
Imported new state entries  count=2233 elapsed=9.461ms    processed=187452035 pending=8725   retry=0  duplicate=367530 unexpected=1054813
Imported new state entries  count=1519 elapsed=44.120ms   processed=187453554 pending=8918   retry=0  duplicate=367530 unexpected=1054813
Imported new state entries  count=2016 elapsed=8.563ms    processed=187455570 pending=8108   retry=0  duplicate=367530 unexpected=1054813
Imported new state entries  count=1519 elapsed=6.350ms    processed=187457089 pending=8487   retry=0  duplicate=367530 unexpected=1054813
Imported new state entries  count=751  elapsed=3.067ms    processed=187457840 pending=8207   retry=0  duplicate=367530 unexpected=1054813
Imported new state entries  count=326  elapsed=803.259µs  processed=187458166 pending=9447   retry=0  duplicate=367530 unexpected=1054813
Imported new state entries  count=1247 elapsed=7.034ms    processed=187459413 pending=10652  retry=0  duplicate=367530 unexpected=1054813
```

Parity Experience

```
Syncing #5945549 8a65…47c9      0 blk/s    11 tx/s    0 Mgas/s
bytes queue 9 MiB sync  RPC:  0 conn,  0 req/s, 340 µs
Imported #5945550 8caf…ac37 (155 txs, 8.00 Mgas, 785.31 ms, 2

Imported #5945551 585d…789b (35 txs, 2.56 Mgas, 124.59 ms, 6.
   #419735    3/25 peers    74 MiB chain 154 MiB db 0 bytes qu
Imported #5945552 9722…0dca (150 txs, 7.99 Mgas, 988.78 ms, 3
Imported #5945553 ce48…4b90 (7 txs, 5.28 Mgas, 68.28 ms, 6.68
Imported #5945554 a0c8…6a33 (199 txs, 7.89 Mgas, 489.44 ms, 3
   #427359    4/25 peers    76 MiB chain 154 MiB db 0 bytes qu
Imported #5945555 8b14…4caf (111 txs, 7.99 Mgas, 210.40 ms, 2
Imported #5945556 cb55…8e09 (168 txs, 7.98 Mgas, 688.74 ms, 2
Syncing #5945556 cb55…8e09      0 blk/s     0 tx/s    0 Mgas/s
```

```
> eth.getBlock(427360)
> null
```

```
2018-07-21 01:53:39  Syncing #1589194 0x19d5…cf87    73.32 blk/s

====================

stack backtrace:
   0:     0x7fcecbb8964c - <no info>
   1:     0x7fcecbb88c12 - <no info>
   2:     0x7fcecbb8816a - <no info>
   3:     0x7fceca710a85 - <no info>
   4:     0x7fcecbbe4d55 - <no info>
   5:     0x7fcecbbb0a16 - <no info>
   6:     0x7fcecbbb00dd - <no info>
   7:     0x7fcecb2bccc5 - <no info>
   8:     0x7fcecb1b9284 - <no info>
   9:     0x7fcecaefba1b - <no info>
  10:     0x7fcecaf182f3 - <no info>
  11:     0x7fcecaefc7fc - <no info>
  12:     0x7fcecaf021d6 - <no info>
  13:     0x7fcecbc10199 - <no info>
  14:     0x7fcecaf16228 - <no info>
  15:     0x7fcecbc015ba - <no info>
  16:     0x7fcecbbe5365 - <no info>
  17:     0x7fcec94376b9 - <no info>
  18:     0x7fcec8f4741c - <no info>
  19:                0x0 - <no info>

Thread 'IO Worker #1' panicked at 'assertion failed: rc == -1', u
```

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	Optimization Enabled:	No
Compiler Version:	v0.4.20+commit.3155dd80	Runs (Optimiser):	200

Contract Source Code </> Copy Find Similar

```
1  // KpopItem is a ERC-721 item (https://github.com/ethereum/eips/issues/721)
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 error
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 (uint256) {
18     if (a == 0) {
19       return 0;
20     }
21     uint256 c = a * b;
```

2018.08.06

Hybrid Approach

Local Ethereum software + Etherscan API

https://etherscan.io/apis

- txlist
- txlistinternal

Empty Code Results

From block 0 to 6,000,000 (July 20 2018):

- 1,799,570 total contracts
- 1,745,317 alive, 54,253 empty contracts
- 28,174 unique creation code for empty contracts
- 32,308 empty contracts with 0 balance

First Contract Creation

Block 46402 (2015-08-07)

https://etherscan.io/tx/0x6c929e1c3d860ee225d7f3a7addf9e3f740603d243260536dfa2f3cf02b51de4

From:	0xa1e4380a3b1f749673e270229993ee55f35663b4
To:	[Contract 0x9a049f5d18c239efaa258af9f3e7002949a977a0 Created] ⚠ Warning! Error encountered during contract execution [**Out of gas**] ☹
Value:	0 Ether ($0.00)
Gas Limit:	24000
Gas Used By Txn:	24000
Gas Price:	0.00001 Ether (10,000 Gwei)
Actual Tx Cost/Fee:	0.24 Ether ($108.78)
Nonce & {Position}:	3 | {0}
Input Data:	0x6060604052600080546001600a060020a031916331790556006806023600039600f3006060604052

```
00000000: PUSH1 0x60  00000015: SWAP1
00000002: PUSH1 0x40  00000016: SSTORE
00000004: MSTORE      00000017: PUSH1 0x6
00000005: PUSH1 0x0   00000019: DUP1
00000007: DUP1        0000001a: PUSH1 0x23
00000008: SLOAD       0000001c: PUSH1 0x0
00000009: PUSH1 0x1   0000001e: CODECOPY
0000000b: PUSH1 0xa0  0000001f: PUSH1 0x0
0000000d: PUSH1 0x2   00000021: RETURN
0000000f: EXP         00000022: STOP
00000010: SUB         00000023: PUSH1 0x60
00000011: NOT         00000025: PUSH1 0x40
00000012: AND         00000027: MSTORE
00000013: CALLER      00000028: STOP
00000014: OR
```

First Contract “Creation” (With Enough Gas)

TxHash:	0x31ded263506ea36e6ea777efc2c39a999e6fba4f4d338c7313af6aac6d9bf3e3
Block Height:	47205 (6013548 block confirmations)
TimeStamp:	1088 days 18 hrs ago (Aug-07-2015 08:26:34 AM +UTC)
From:	0xfbe0afcd7658ba86be41922059dd879c192d4c73
To:	[Contract 0xc669eaad75042be84daaf9b461b0e868b9ac1871 Created]
Value:	14.985 Ether ($6,790.60)
Gas Limit:	21000
Gas Used By Txn:	21000
Gas Price:	0.0000005 Ether (500 Gwei)
Actual Tx Cost/Fee:	0.0105 Ether ($4.76)
Nonce & {Position}:	0 | {0}
Input Data:	0x

First Contract Creation (With Code)

Block 48643 (2015-08-07)

Account 0x6516298e1c94769432ef6d5f450579094e8c21fa

Block	Age	From		To	Value	[TxFee]
5868974	32 days 15 hrs ago	0x3406577823d26c…	IN	0x6516298e1c9476…	69 wei	0.000035753454
5864543	33 days 9 hrs ago	0x0668dea6b5ec94…	IN	0x6516298e1c9476…	420 wei	0.000105345
5864460	33 days 9 hrs ago	0x20c945800de433…	IN	0x6516298e1c9476…	0 Ether	0.00004508
48643	1088 days 12 hrs ago	0x3d0768da09ce77…	IN	Contract Creation	0 Ether	0.005201664565

Top Duplicates

Count: 10,072

Code:

```
0x5b620186a05a1315
60135760016020526 0
00565b600080601f60
0039601f565b6000f3
```

Top Duplicates (2)

Count: 9,512

Code:

```
0x
```

Total: 6203 ETH (~$2,600,000)

Top Duplicates (3)

Count: 1,963

Code:

```
0x0000000000000000000000000000000000000000000000000000000000000000…0000000000000000000000000000000000000000000000000000000000000000
```

6000 NULs (STOP)

*EIP-170 sets max size to 0x6000

Noise / Spam

0x7F62E6C7Ec6700187aB99f71997912A9CDF184D1

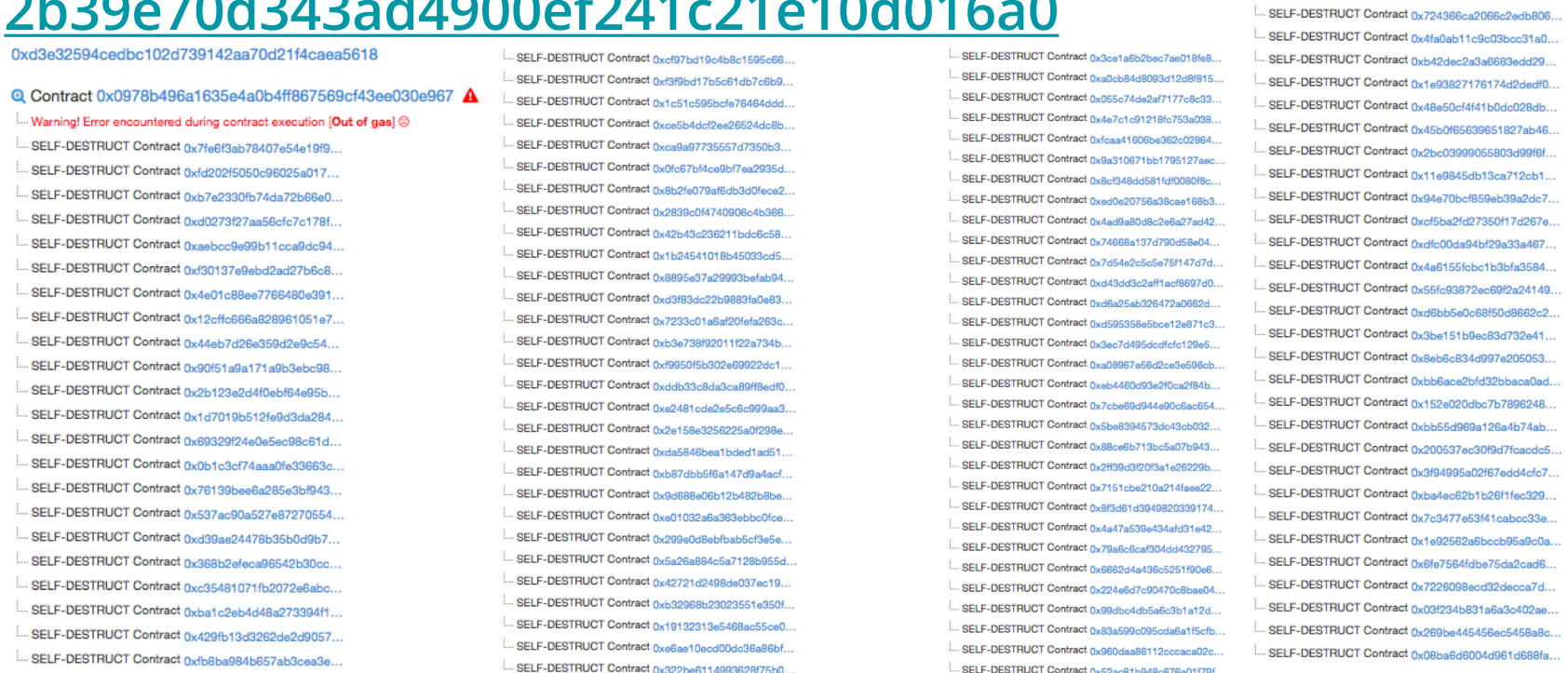
```
PUSH20 0xff5932556071d5ac315d240b92b97a3b4f7daf3d
SELFDESTRUCT
```

0, 1 or 2 Wei transferred

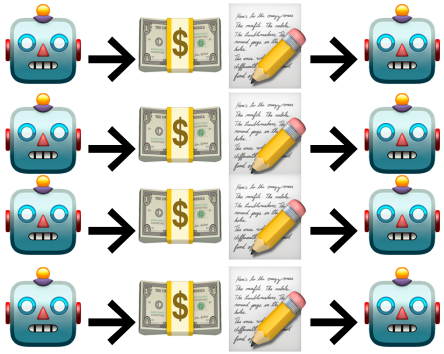
1988 contracts after filtering

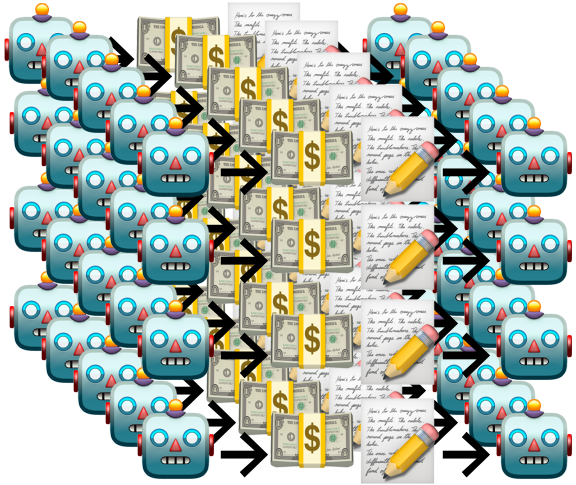
Massive selfdestruct

https://etherscan.io/tx/0x0bb3c5ec638d167a00d3e790cbf7692b39e70d343ad4900ef241c21e10d016a0



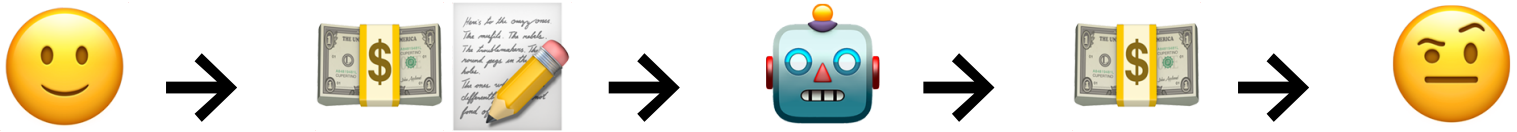
Massive selfdestruct (2)

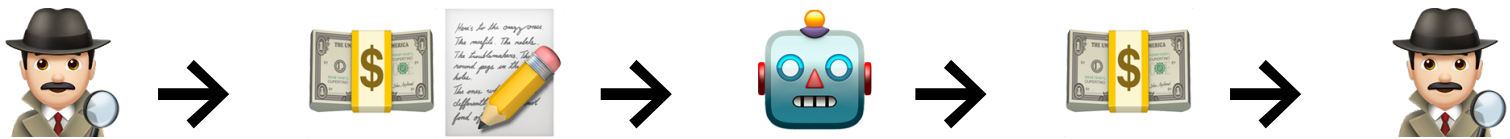




Analyzing 2,000 Deaths

Criteria





Creator != selfdestruct destination

From trace, we know the destination of selfdestruct

Filter when this is not the original contract creator

630 contracts remaining

10 of these send ETH to address 0x0

50ETH to 0x0

0xf73d247ffDBD5A9964d1a1444c86343650b67ed4

https://etherscan.io/address/0xf73d247ffdbd5a9964d1a1444c86343650b67ed4

```
Function: kill(address _to) MethodID: 0xcbf0b0c0

[0]:
0000000000000000000000000000000000000000000000000000000000000000
```

10,000 ETH!

0xf199Af8B17D81c41ABe6220a1D7C9fe04d0d9d2c

Parity multisig wallet initWallet() attack?

https://blog.zeppelin.solutions/on-the-parity-wallet-multisig-hack-405a8c12e8f7

0xbec591de75b869...	→	0xb3764761e297d6...	82,189 Ether
0x50126e8fcb9be2...	→	0xb3764761e297d6...	44,055 Ether
0x91efffb9c6cd3a6...	→	0xb3764761e297d6...	26,793 Ether

Creator != selfdestruct transaction originator

159 contracts meet these conditions

25 contracts sent > 0.1 ETH

Only 16 contracts sent >= 1 ETH

300ETH selfdestruct

```
Account:0x96f65700904cB464F3D153a2744B84FCa27ABF9C

Sent 300ETH to 0xCafe00be401442Bfb5E480C355393FD8C147abBB


Function: changeOwner(address _from, address _to) ***

MethodID: 0xf00d4b5d

[0]:  000000000000000000000000374139a05ac55917badd3f934f1b93f5c8623ded

[1]:  000000000000000000000000cafe00be401442bfb5e480c355393fd8c147abbb
```

Dice2Win

0xD1CEeee6B94DE402e14F24De0871580917ede8a7

Sent 65.7 ETH to 0xD1CEeee271fD5a8B0e2BFc12Ea5B5b2E5CeDEc95

Function: approveNextOwner(address _nextOwner)

MethodID: 0xd579fd44

[0]: 000000000000000000000000d1ceeee271fd5a8b0e2bfc12ea5b5b2e5cedec95

Etherwow

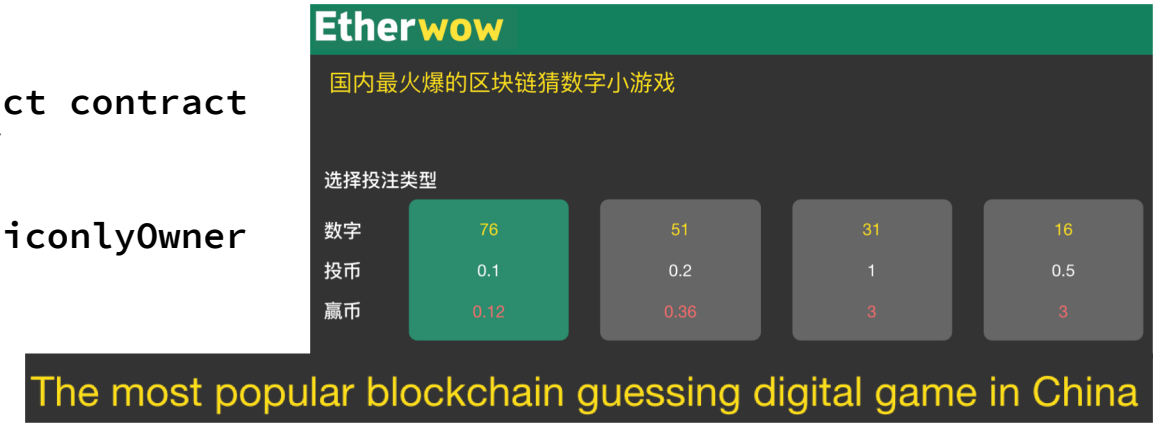
0x4DF6DE08D11f11EBAd5d9E136B768849426fB8a7

Function: ownerChangeOwner(address newOwner)

MethodID: 0x4f44728d

[0]: 000000000000000000000000007d138be0eed529ae42a468472b2beb0314af5e28

```
Function: ownerkill()
    /** @dev owner selfdestruct contract
***BE CAREFUL! EMERGENCY ONLY
 / CONTRACT UPGRADE*** */
    function ownerkill() publiconlyOwner
    { selfdestruct(owner); }
```



Becoming Mortal

```
0xf4D3CEd0929eA3F3Fd94F32ba460a66b428932F2
```

```
function mortal() { owner = msg.sender; }

function kill() {
    if(msg.sender == owner) selfdestruct(owner);
}
```

Conclusion

If you are developing contracts:

- Understand and fix all warnings
- Add an Echidna test
- Write exhaustive positive/negative tests
- **Perform an rigorous assessment**

If you are a security researcher:

- Become a blockchain explorer
- Have patience
- Symbolically execute with Manticore
- **Work with us**

Contact

Jay Little, Principal Security Engineer
jay@trailofbits.com
@computerality

@trailofbits
www.trailofbits.com
github.com/trailofbits
blog.trailofbits.com

We're Hiring!

Trail of Bits is hiring engineers and vulnerability researchers who are excited about C++ code, blockchain software, and smart contracts.