EXPLAIN ETHEREUM SMART CONTRACT HACKING LIKE I AM FIVE

Zoltan Balazs
2018 October

MRG Effitas
Efficacy Assessment & Assurance

EXPLAIN TO ME HOW BITCOIN WORKS.
AND DON'T GIVE ME THE CONDESCENDING SIMPLE VERSION FOR MANAGERS. I WANT A FULL TECHNICAL EXPLANATION.
EARLY CIVILIZATIONS HAD NO CONCEPT OF ZERO.
GO ON.

Derivative work. Original by Scott Adams. www.dilbert.com

Original © 2003 United Feature Syndicate, Inc.

Is there anything malware related in this talk?

NO

This is like the “skateboarding dog” at the end of the news.
Totally not relevant, but probably funny …

Questions

Hands up if you know something about blockchain

Hands up if you have ever tried to explain Bitcoin to your parents/colleagues/kids

Hands up if it ended: “it is complicated”

Hands up if you have ever interacted with a Smart Contract

Why am I talking about this?

ITSEC folks laugh about Blockchain a lot

They believe it is

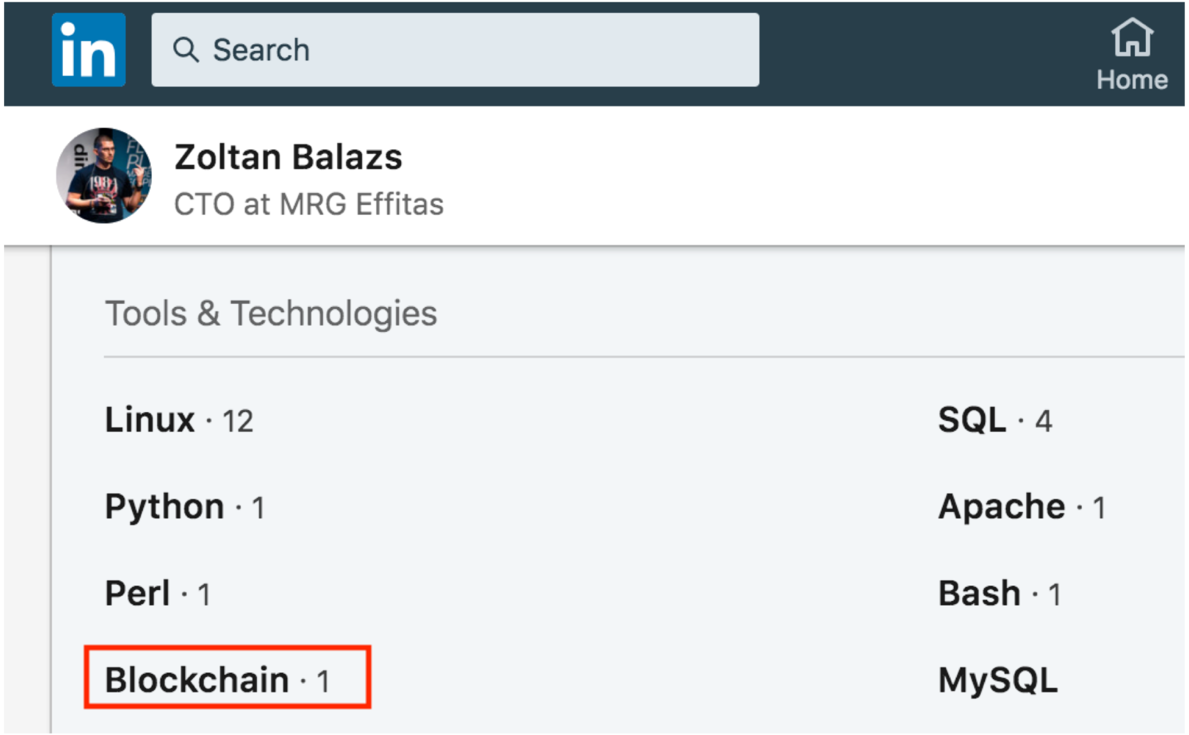
- not happening
- not important
- not working

Bad news: it is happening!

Trust me, this is an important topic

I will calculate losses in Lamborghinis – 200K USD

So who am I to talk about this topic?



Zoltan Balazs

CTO at MRG Effitas

Tools & Technologies

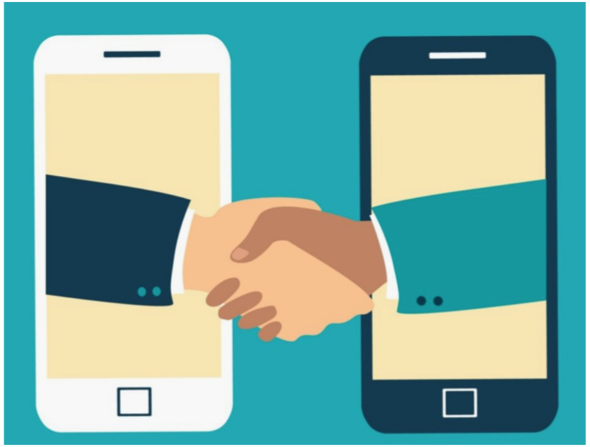
Linux · 12	SQL · 4
Python · 1	Apache · 1
Perl · 1	Bash · 1
Blockchain · 1	MySQL

I don’t give advice on investing/ selling/ HODLing cryptocurrencies

Everything you don’t understand about money combined

with everything you don’t understand about computers

Smart Contracts



Assume you have a basic understanding of cryptocurrencies in general

Let’s take a deep dive into **Smart Contracts**

Bitcoin is also capable of doing Smart Contracts

Ethereum YAC (Yet Another Cryptocurrency) was designed for Smart Contracts

Smart Contracts

You want to sign and get a countersign of the contract

carve the contract into stone
contracts carved into the stone
cannot be modified

In the smart contract world, the **stone** is the **blockchain**

it is powered by the time and
energy spent on solved math
challenges

Code is universal

Ethereum Virtual Machine

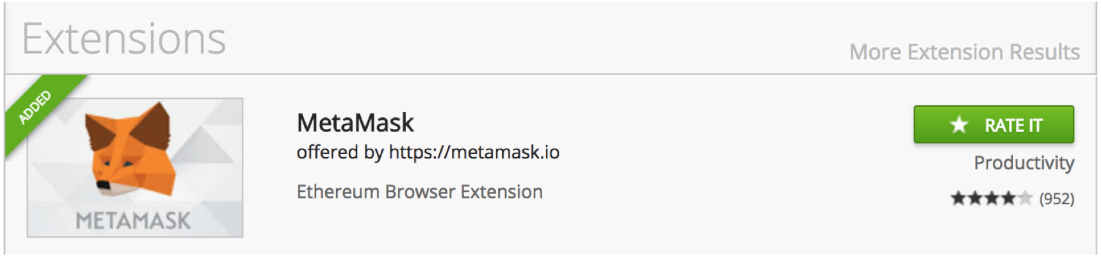
Bytecode: it is not a machine code, thus you need a VM to execute it

- **Solidity**: compile JavaScript-like code into **EVM** bytecode

Source code can be published - creates trust

Solidity source code compiles into the same bytecode (reproducible)

- At least with the same parameters and same compiler version



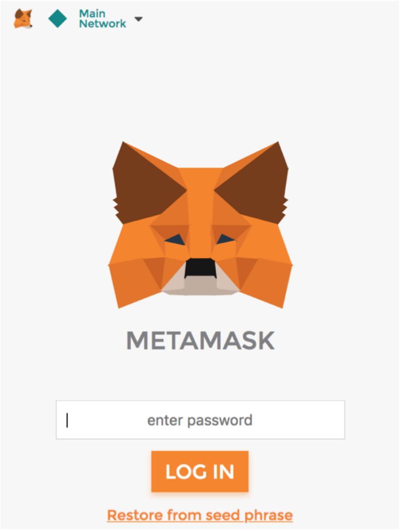
Extensions

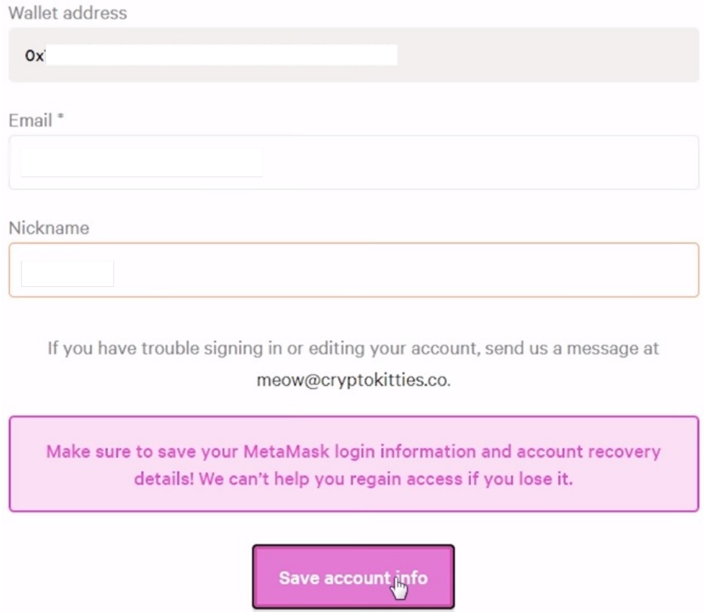
More Extension Results

MetaMask

offered by https://metamask.io

Ethereum Browser Extension





Wallet address

0x[redacted]

Email *

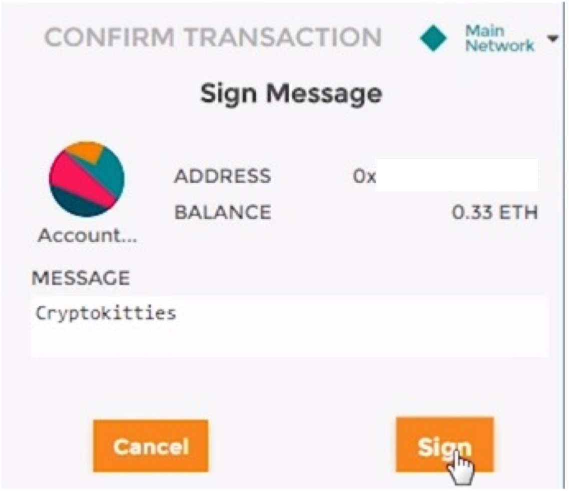
[redacted]

Nickname

[redacted]

If you have trouble signing in or editing your account, send us a message at meow@cryptokitties.co.

Make sure to save your MetaMask login information and account recovery details! We can't help you regain access if you lose it.



CONFIRM TRANSACTION

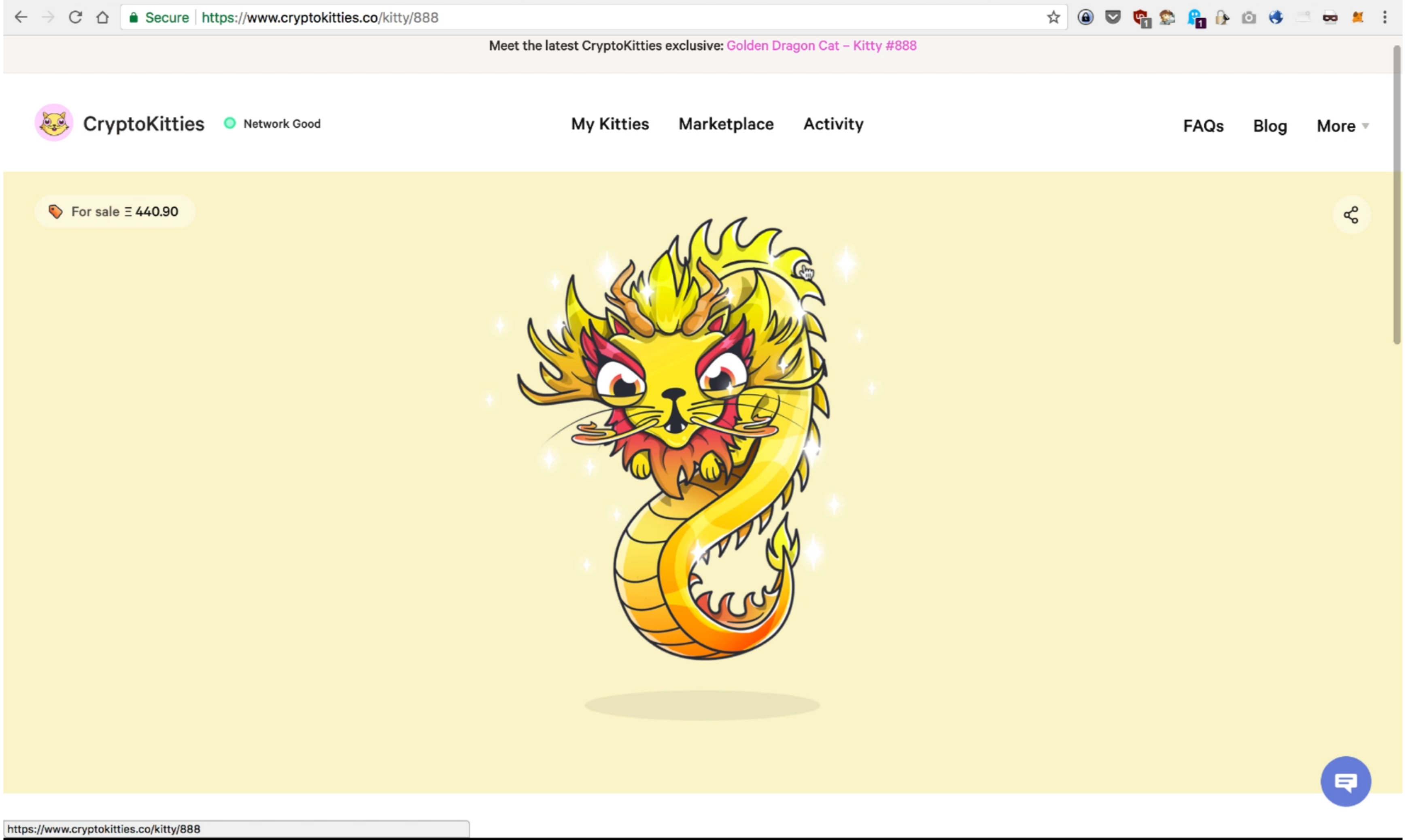
Sign Message

ADDRESS 0x[redacted]

BALANCE 0.33 ETH

MESSAGE

Cryptokitties



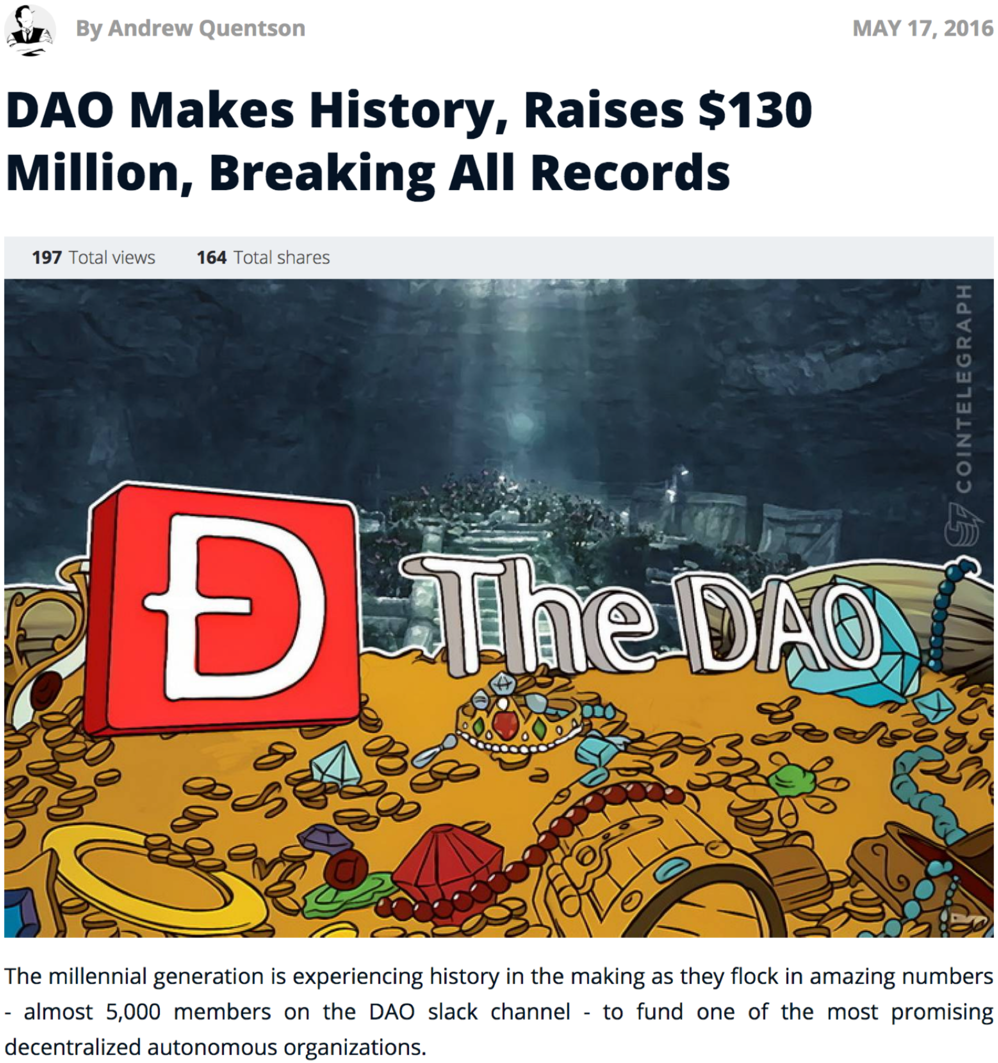
Meet the latest CryptoKitties exclusive: Golden Dragon Cat – Kitty #888

For sale Ξ 440.90

https://www.cryptokitties.co/kitty/888

Smart contracts are code

Code can be hacked



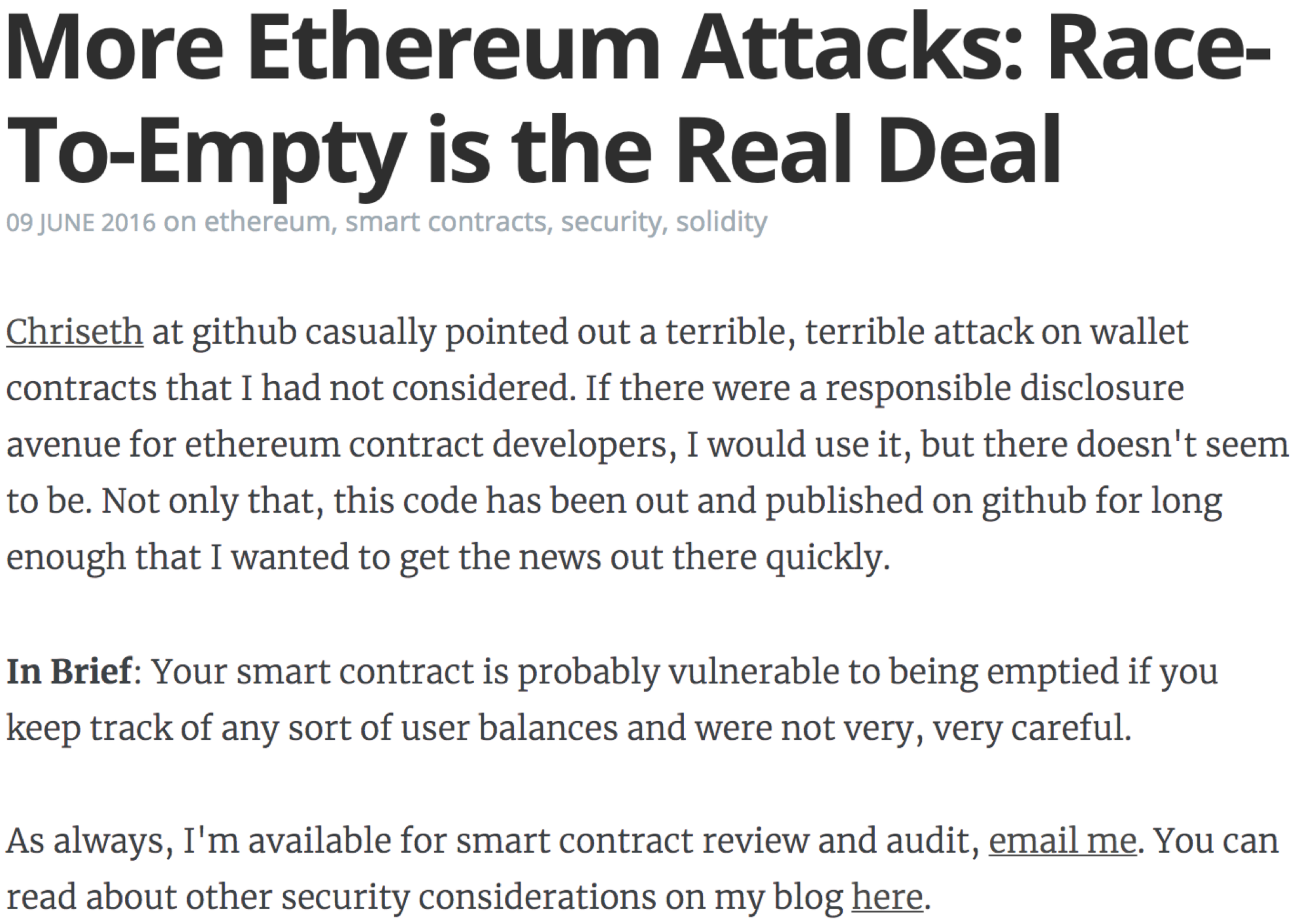
By Andrew Quentson

MAY 17, 2016

DAO Makes History, Raises $130 Million, Breaking All Records

197 Total views **164** Total shares

The millennial generation is experiencing history in the making as they flock in amazing numbers - almost 5,000 members on the DAO slack channel - to fund one of the most promising decentralized autonomous organizations.



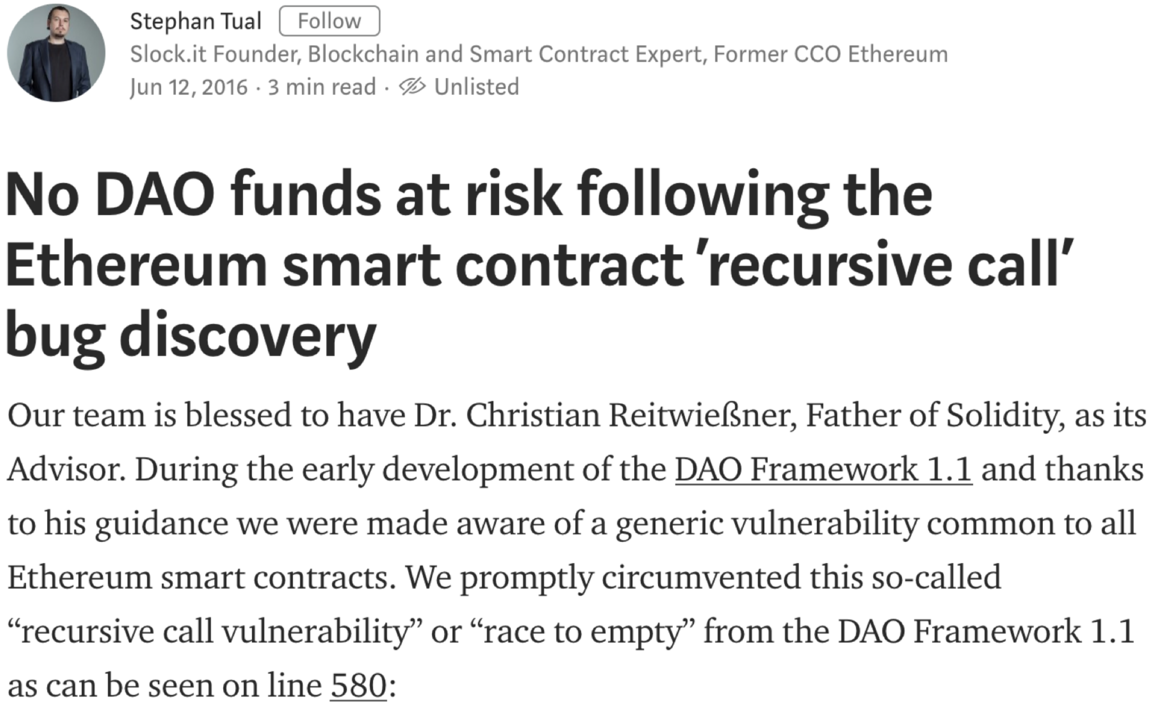
More Ethereum Attacks: Race-To-Empty is the Real Deal

09 JUNE 2016 on ethereum, smart contracts, security, solidity

Chriseth at github casually pointed out a terrible, terrible attack on wallet contracts that I had not considered. If there were a responsible disclosure avenue for ethereum contract developers, I would use it, but there doesn't seem to be. Not only that, this code has been out and published on github for long enough that I wanted to get the news out there quickly.

In Brief: Your smart contract is probably vulnerable to being emptied if you keep track of any sort of user balances and were not very, very careful.

As always, I'm available for smart contract review and audit, email me. You can read about other security considerations on my blog here.



Stephan Tual Follow

Slock.it Founder, Blockchain and Smart Contract Expert, Former CCO Ethereum

Jun 12, 2016 · 3 min read · Unlisted

No DAO funds at risk following the Ethereum smart contract 'recursive call' bug discovery

Our team is blessed to have Dr. Christian Reitwießner, Father of Solidity, as its Advisor. During the early development of the DAO Framework 1.1 and thanks to his guidance we were made aware of a generic vulnerability common to all Ethereum smart contracts. We promptly circumvented this so-called "recursive call vulnerability" or "race to empty" from the DAO Framework 1.1 as can be seen on line 580:

The important takeaway from this is: as there is no ether whatsoever in the DAO's rewards account—this is NOT an issue that is putting any DAO funds at risk today.

The DAO: Recursive call + race condition

June 18th, 2016

Attacker transfers Ether worth $250 million from DAO

That is 1250 Lamborghinis

Reentrancy at the splitDAO function

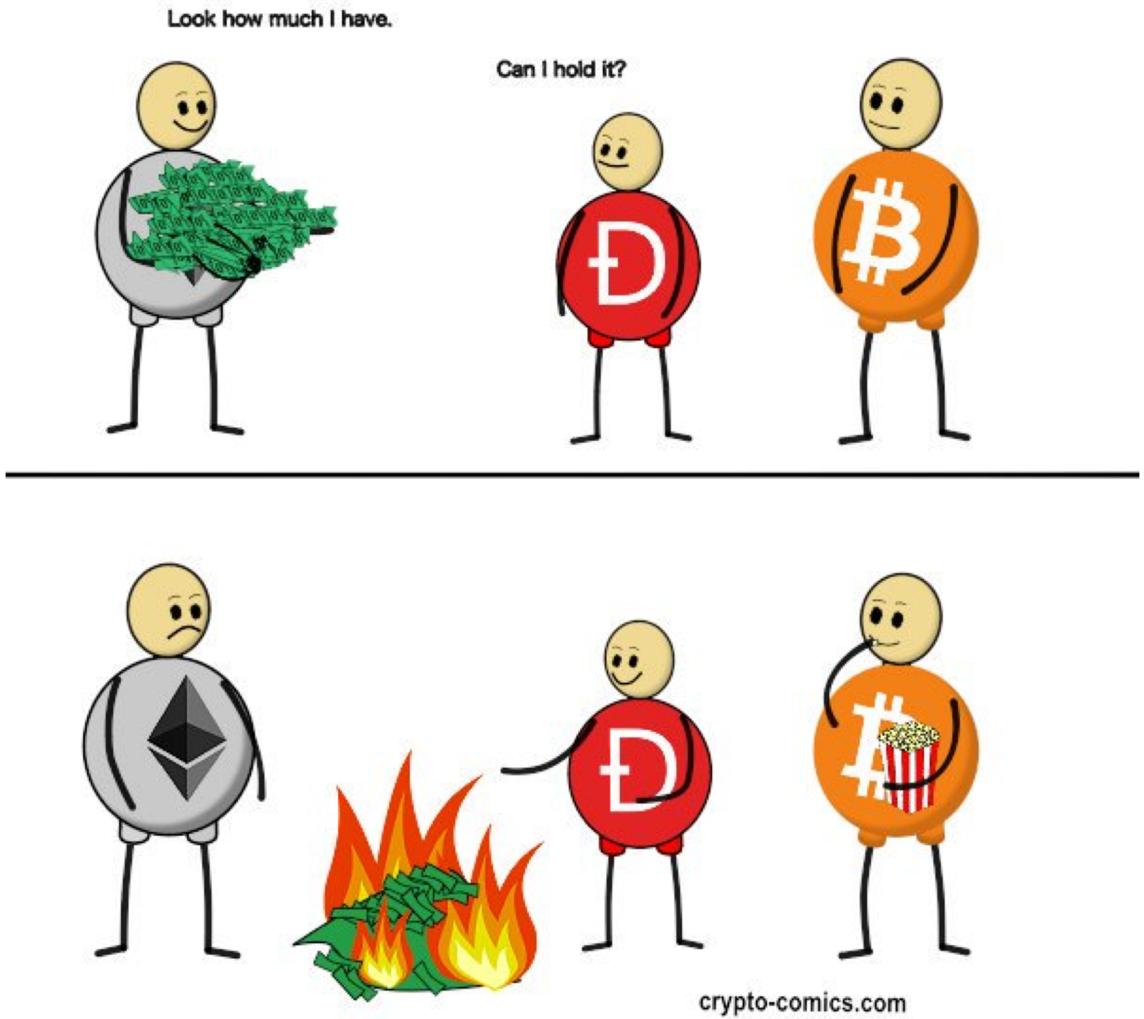
The DAO hack

You can interrupt the bank teller while he is giving you money

The bank teller only updates your balance at the end

The DAO hack …

```
// INSECURE  --- this is not DAO code, but similar so it is easy to understand

function withdrawBalance() public {                          // 1st line

    uint amountToWithdraw = userBalances[msg.sender];        // 2nd line

    require(msg.sender.call.value(amountToWithdraw)()); // 3rd line. At this point, the caller's code is
executed, and can call withdrawBalance again

    userBalances[msg.sender] = 0;                 // 4th line

}
```

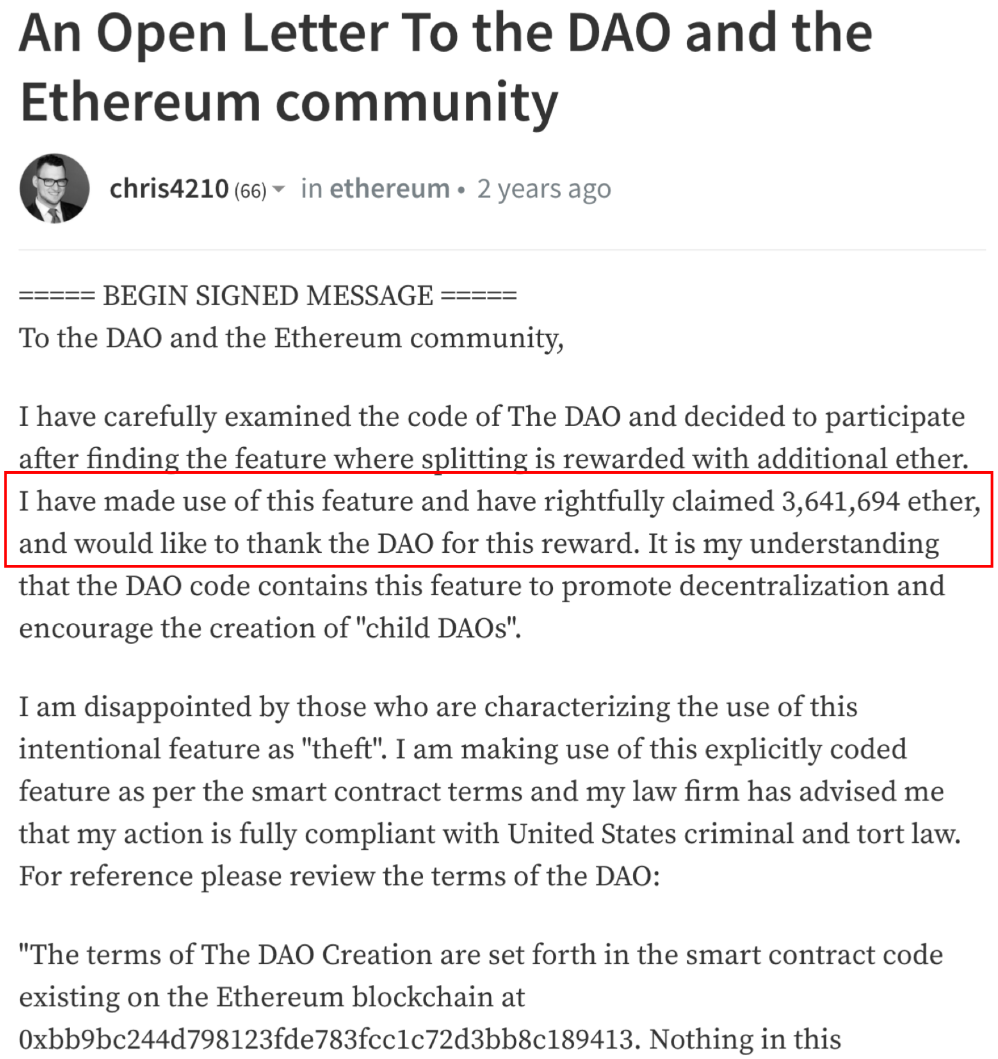

Look how much I have.
Can I hold it?
crypto-comics.com

The solution?

Rewrite the past and pretend it didn’t happen

Attacker got away with his ETH Classic

worth $67.4 million – 337 Lambos



An Open Letter To the DAO and the Ethereum community

chris4210 (66) in ethereum • 2 years ago

===== BEGIN SIGNED MESSAGE =====
To the DAO and the Ethereum community,

I have carefully examined the code of The DAO and decided to participate after finding the feature where splitting is rewarded with additional ether. I have made use of this feature and have rightfully claimed 3,641,694 ether, and would like to thank the DAO for this reward. It is my understanding that the DAO code contains this feature to promote decentralization and encourage the creation of "child DAOs".

I am disappointed by those who are characterizing the use of this intentional feature as "theft". I am making use of this explicitly coded feature as per the smart contract terms and my law firm has advised me that my action is fully compliant with United States criminal and tort law. For reference please review the terms of the DAO:

"The terms of The DAO Creation are set forth in the smart contract code existing on the Ethereum blockchain at 0xbb9bc244d798123fde783fcc1c72d3bb8c189413. Nothing in this

Multi-signature wallets

“Captain planet, the world’s first multi-factor authentication” © dnet

Shared vulnerable library + reinit - 2017 July 20

$31M stolen – 155 Lambos

- A lot more was in danger, but good guys were faster

Lot of shared libraries exists in the blockchain

- Save gas
- Contracts now share the same vulnerabilities

Parity multi-signature wallets

Teh code

NON LIBRARY CODE

```
function() payable {  // someone called a function we don’t have?
  if (msg.value > 0)            // some ether is sent
    ...
  else if (msg.data.length > 0)         //ether is not sent, but some data is
    _walletLibrary.delegatecall(msg.data);                  //let’s check if we can execute this code via shared library
}
```

- If the method name is not defined on this contract…
- And there’s no ether being sent in the transaction…
- And there is some data in the message payload…

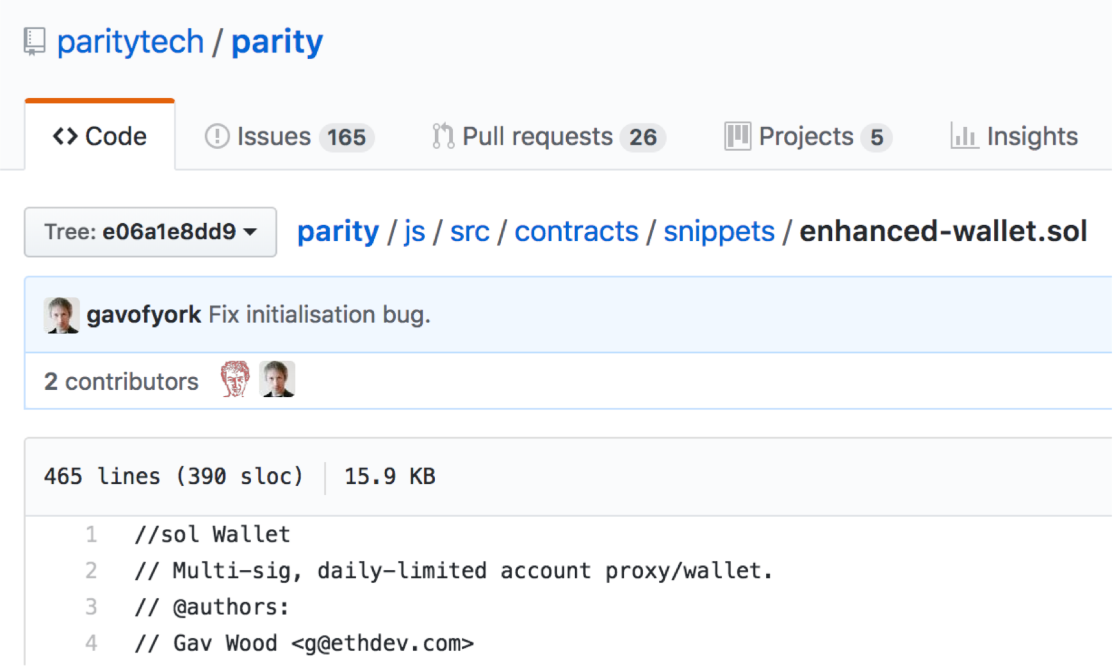
for whatever method that calls DELEGATECALL, it will call the same method on the contract you're delegating to, but using the context of the current contract

Teh library codez

```
function initWallet(address[] _owners, uint _required, uint _daylimit) {
          //the shared library has initWallet and it is public !


  initDaylimit(_daylimit);
  initMultiowned(_owners, _required);
}
```

initWallet is not in the non-library code, but is called in the shared library

So some random guys don’t know how to code Smart Contracts …



paritytech / parity

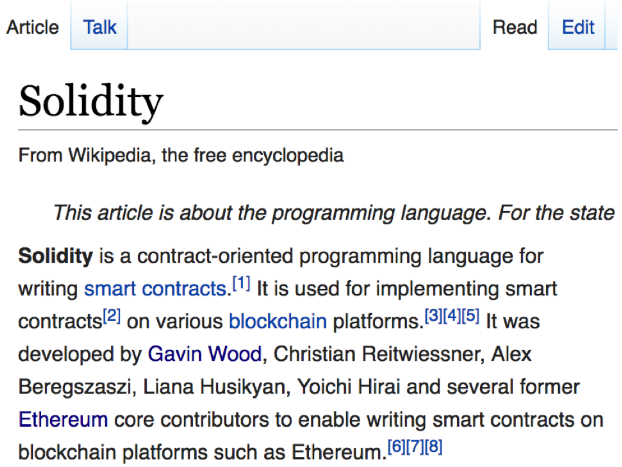
parity / js / src / contracts / snippets / enhanced-wallet.sol

gavofyork Fix initialisation bug.

2 contributors

465 lines (390 sloc) | 15.9 KB

```
//sol Wallet
// Multi-sig, daily-limited account proxy/wallet.
// @authors:
// Gav Wood <g@ethdev.com>
```



Solidity

From Wikipedia, the free encyclopedia

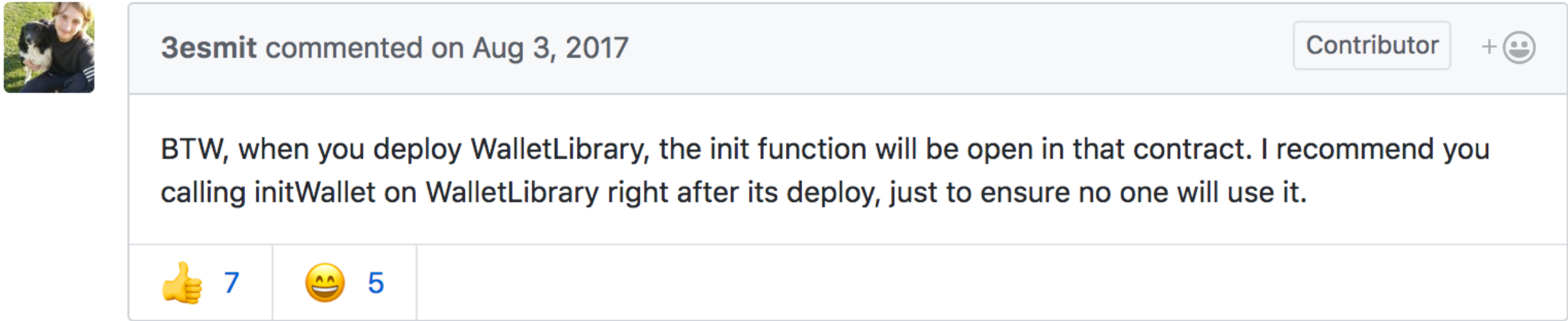
This article is about the programming language. For the state

Solidity is a contract-oriented programming language for writing smart contracts.[1] It is used for implementing smart contracts[2] on various blockchain platforms.[3][4][5] It was developed by Gavin Wood, Christian Reitwiessner, Alex Beregszaszi, Liana Husikyan, Yoichi Hirai and several former Ethereum core contributors to enable writing smart contracts on blockchain platforms such as Ethereum.[6][7][8]

Fixing the Parity bug

Parity fixed previous bug

and introduced a new one



3esmit commented on Aug 3, 2017

BTW, when you deploy WalletLibrary, the init function will be open in that contract. I recommend you calling initWallet on WalletLibrary right after its deploy, just to ensure no one will use it.

👍 7 😄 5

Library contract was not initialized properly. That allowed anyone to turn the library contract into a multi-sig wallet

The next Parity hack

November 2017 - $300M lost – 1500 Lambos

@devops199 “accidentally” called initWallet() method to own the library

@devops199 “accidentally” called kill() method to self-destruct it

It was planned to be fixed – forking EIP-999. Community voted no

Conclusion

Blockchain, Ethereum, Smart Contracts are here to hack

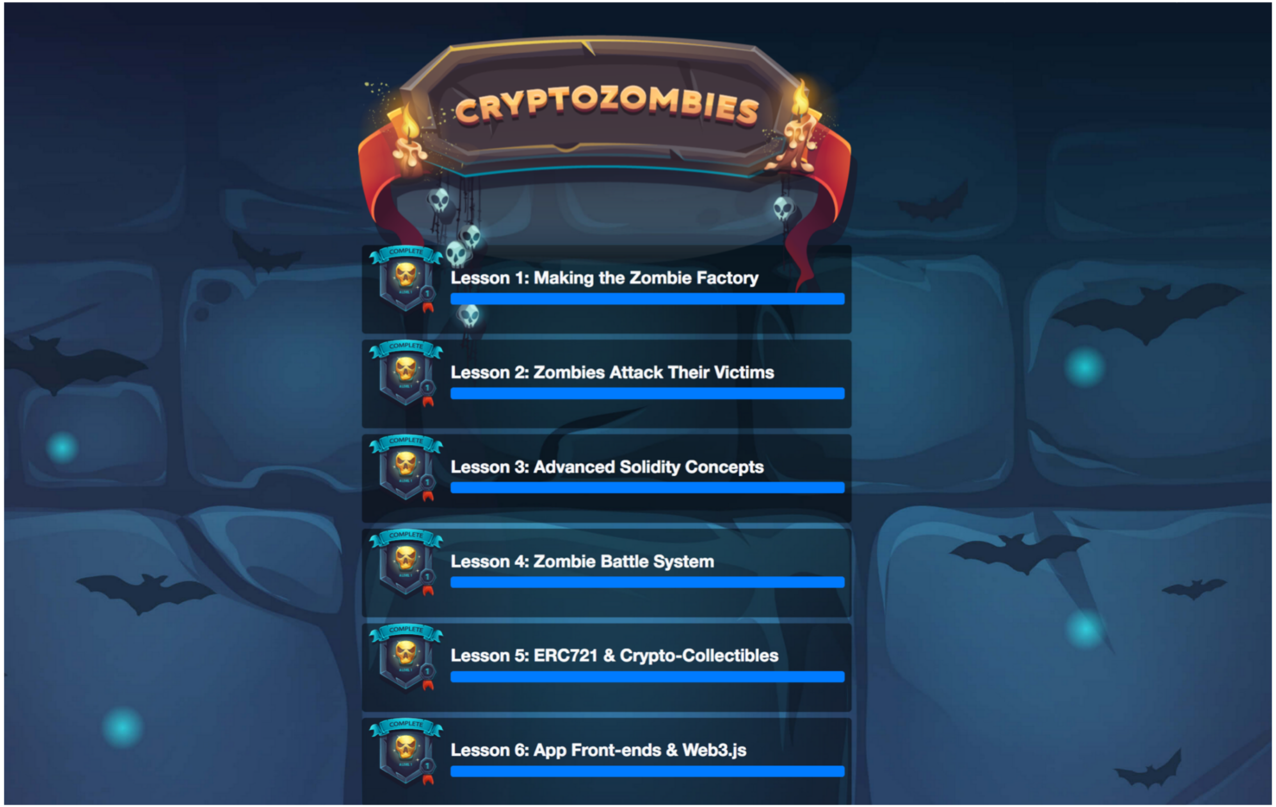
Writing secure Smart Contracts is hard

Ethereum is still in beta

Hacking Smart Contracts is possible, fun, but probably illegal

- Hacking your own smart contract is probably not illegal
- Hacking in test blockchain is not illegal

Where to learn to code? cryptozombies.io



Lesson 1: Making the Zombie Factory

Lesson 2: Zombies Attack Their Victims

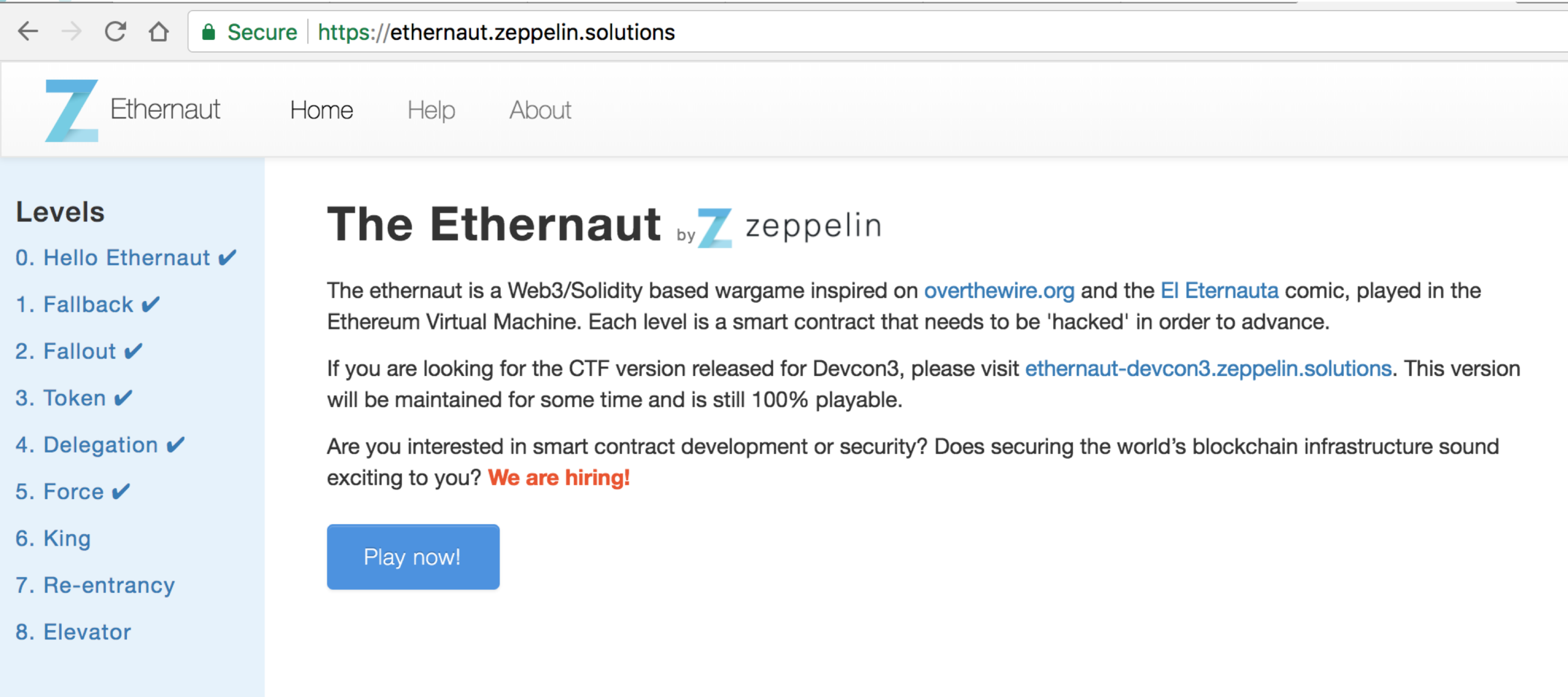
Lesson 3: Advanced Solidity Concepts

Lesson 4: Zombie Battle System

Lesson 5: ERC721 & Crypto-Collectibles

Lesson 6: App Front-ends & Web3.js

Where to learn to hack?



Levels

0. Hello Ethernaut ✔
1. Fallback ✔
2. Fallout ✔
3. Token ✔
4. Delegation ✔
5. Force ✔
6. King
7. Re-entrancy
8. Elevator

The Ethernaut by zeppelin

The ethernaut is a Web3/Solidity based wargame inspired on overthewire.org and the El Eternauta comic, played in the Ethereum Virtual Machine. Each level is a smart contract that needs to be 'hacked' in order to advance.

If you are looking for the CTF version released for Devcon3, please visit ethernaut-devcon3.zeppelin.solutions. This version will be maintained for some time and is still 100% playable.

Are you interested in smart contract development or security? Does securing the world's blockchain infrastructure sound exciting to you? **We are hiring!**

References

Nick Szabo: The idea of smart contracts 1997 https://perma.cc/V6AZ-7V8W

https://www.reddit.com/r/explainlikeimfive/comments/12knie/eli5_bitcoins/?st=IZW0ENOG&sh=d566a3ee

https://medium.freecodecamp.org/smart-contracts-for-dummies-a1ba1e0b9575

https://www.reddit.com/r/explainlikeimfive/comments/4lz9t4/eli5_ethereum/

http://hackingdistributed.com/2016/06/18/analysis-of-the-dao-exploit/

https://github.com/b-mueller/smashing-smart-contracts/blob/master/smashing-smart-contracts-1of1.pdf

https://medium.freecodecamp.org/a-hacker-stole-31m-of-ether-how-it-happened-and-what-it-means-for-ethereum-9e5dc29e33ce

https://www.stateofthedapps.com/

Cryptozombies.io - best tutorial

Latest hype and scams: https://boards.4chan.org/biz/

PRESTO

Whoami?

zoltan.balazs@mrg-effitas.com

https://hu.linkedin.com/in/zbalazs

Twitter – @zh4ck

www.slideshare.net/bz98

HACKERSULI !!!1!

https://JumpESPJump.blogspot.com

Zombie Browser Toolkit

https://github.com/Z6543/ZombieBrowserPack

HWFW Bypass tool

Similar stuff was used in PacketRedirect in Danderspritz FlewAvenue by EQGRP

https://github.com/MRGEffitas/hwfwbypass

Malware Analysis Sandbox Tester tool

https://github.com/MRGEffitas/Sandbox_tester

Played with crappy IoT devices – my RCE exploit code running on ~600 000 IP cameras via Persirai

https://jumpespjump.blogspot.hu/2015/09/how-i-hacked-my-ip-camera-and-found.html

https://jumpespjump.blogspot.hu/2015/08/how-to-secure-your-home-against.html

Invented the idea of encrypted exploit delivery via Diffie-Hellman key exchange, to bypass exploit detection appliances

Implemented by Angler and Nuclear exploit kit developers

https://www.mrg-effitas.com/generic-bypass-of-next-gen-intrusion-threat-breach-detection-systems/

HACKTIVITY