RSAConference2018
San Francisco | April 16 – 20 | Moscone Center

SESSION ID: HT-F03

LOST IN THE ETHER

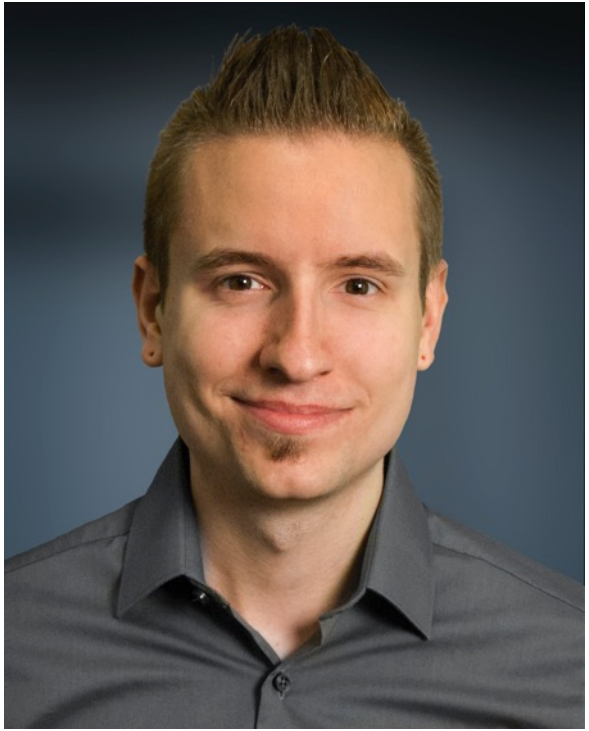
HOW ETHEREUM HACKS ARE SHAPING THE BLOCKCHAIN FUTURE

Marc Laliberte

Sr. Security Analyst
WatchGuard Technologies
@XORRO_

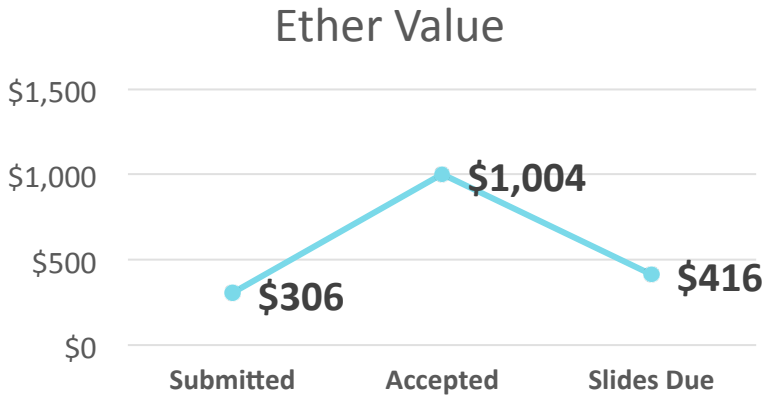
ABOUT ME

- Marc Laliberte
- Sr. Security Analyst
- 6 years at WatchGuard
- WatchGuard Threat Lab Lead



Presentation Overview

- What to expect:
  - Basic intro to cryptocurrency
  - Ethereum Virtual Machine primer
  - Discussion on EVM security



CRYPTOCURRENCY OVERVIEW

Intro Video

Blockchain, more than a buzzword

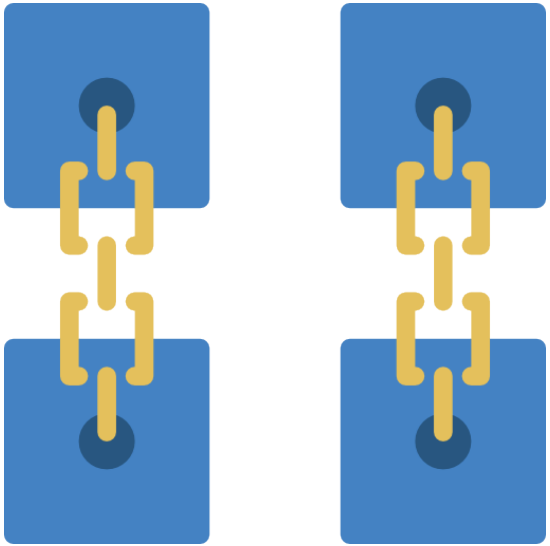
Kodak announces its own cryptocurrency and watches stock price skyrocket

Kodak stock jumps 60 percent after the surprise announcement

By Shannon Liao | @Shannon_Liao | Jan 9, 2018, 3:22pm EST

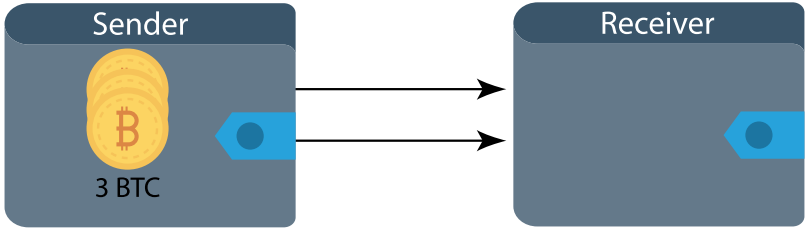
Blockchain, more than a buzzword

- Distributed public ledger
- Public Key Encryption
- Immutable*

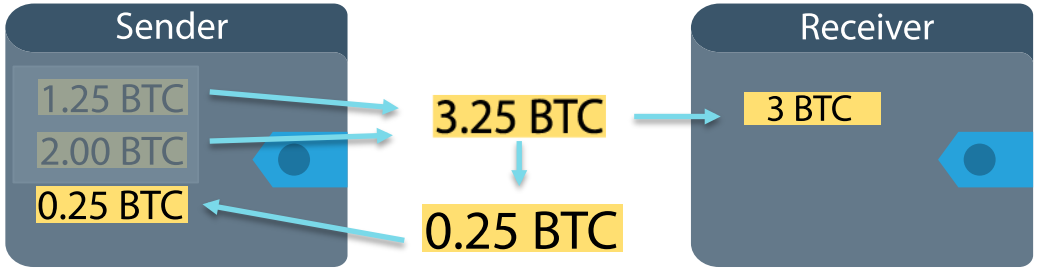


Example Bitcoin Transaction

- Basic Concept:
  - From
  - To
  - Value



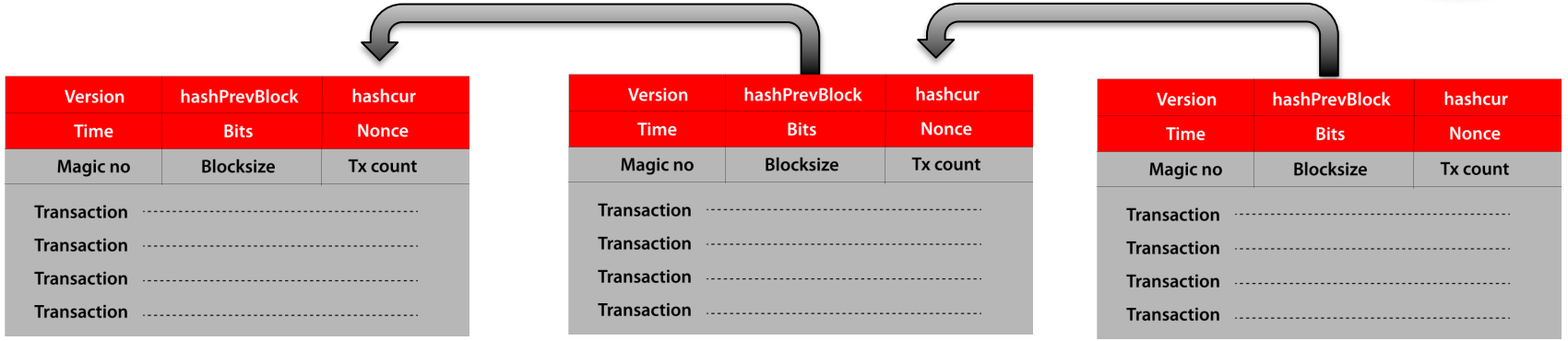
- Actual Implementation:
  - Inputs
  - Outputs

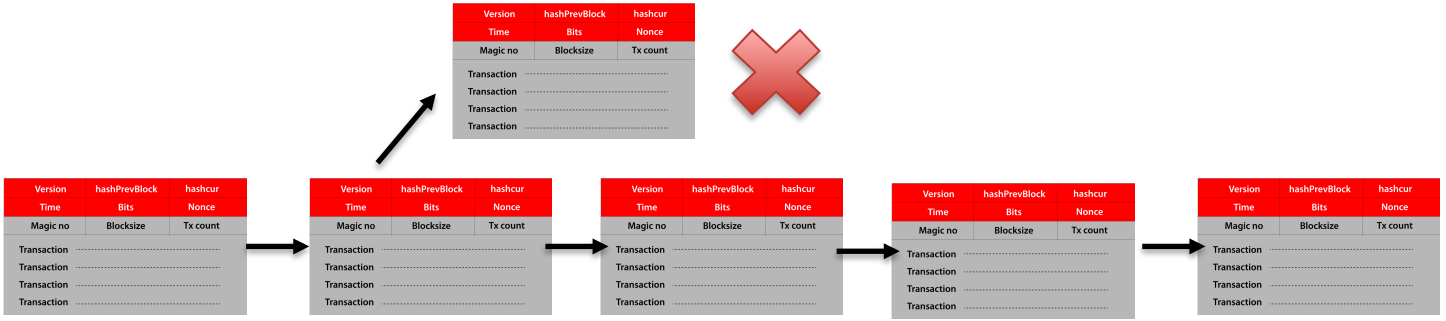


Example Bitcoin Block

Version	hashPrevBlock	hashcur
Time	Bits	Nonce
Magic no	Blocksize	Tx count
Transaction		
Transaction		
Transaction		
Transaction		

Example Bitcoin Blockchain





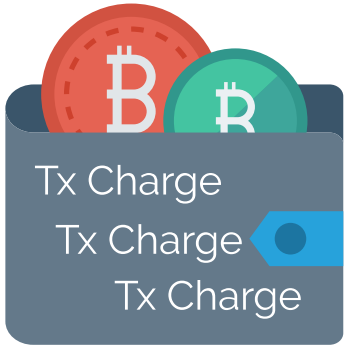
ENTER, ETHEREUM

What's Ethereum?

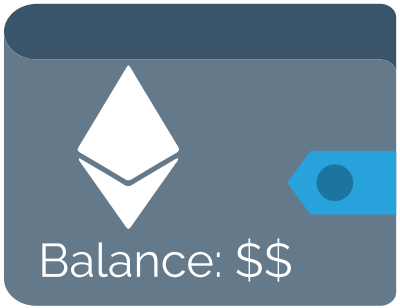
- Terminology:
  - Ethereum = the platform
  - Ether = the cryptocurrency
  - Wei = 1,000,000,000,000,000,000 Ether
  - Smart contract = fancy transaction
  - Gas = cost (in GWei) to execute

Bitcoin and Ethereum User Accounts

- Bitcoin:
  - Input and output states
  - Wallet value is an accumulation of inputs and leftover “change” from outputs



- Ethereum:
  - Simple transactions
  - Wallet is an account that holds a value amount

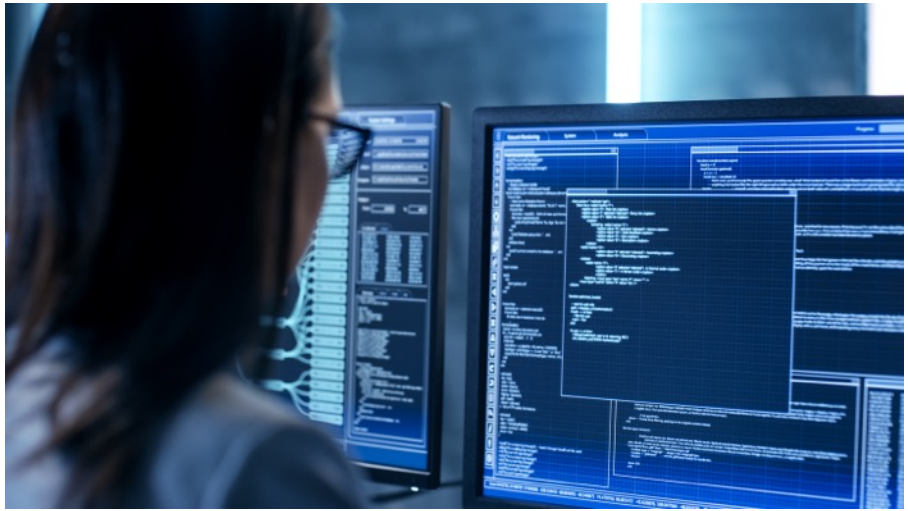


Ethereum Virtual Machine

- Instead of pre-defined actions (like bitcoin transactions), allows full programming using Solidity language
- Securely executes untrusted code
- Execution results compared to all other nodes on the network

What is a smart contract?

- A type of account, just like user accounts
  - User accounts also called Externally Owned Accounts (EOAs)
- A collection of code and data



Smart Contract Example: High-Five Coin


```
contract HFCoin {
    string public name;
    string public symbol;
    mapping (address => uint256) public balanceOf;

    function HFCoin(
        uint256 initialSupply,
        string tokenName,
        string tokenSymbol
    ) public
    {
        balanceOf[msg.sender] = initialSupply;
        name = tokenName;
        symbol = tokenSymbol;
    }

    function transfer(address _to, uint256 _value) public {
        require(balanceOf[msg.sender] >= _value);
        require((balanceOf[_to] += _value) >= balanceOf[_to]);
        balanceOf[msg.sender] -= _value;
        balanceOf[_to] += _value;
    }
}
```

Smart Contract Example: High-Five Coin


```
contract HFCoin {
    string public name;
    string public symbol;
    mapping (address => uint256) public balanceOf;

    function HFCoin(
        uint256 initialSupply,
        string tokenName,
        string tokenSymbol
    ) public
    {
        balanceOf[msg.sender] = initialSupply;
        name = tokenName;
        symbol = tokenSymbol;
    }

    function transfer(address _to, uint256 _value) public {
        require(balanceOf[msg.sender] >= _value);
        require((balanceOf[_to] += _value) >= balanceOf[_to]);
        balanceOf[msg.sender] -= _value;
        balanceOf[_to] += _value;
    }
}
```

Smart Contract Example: High-Five Coin

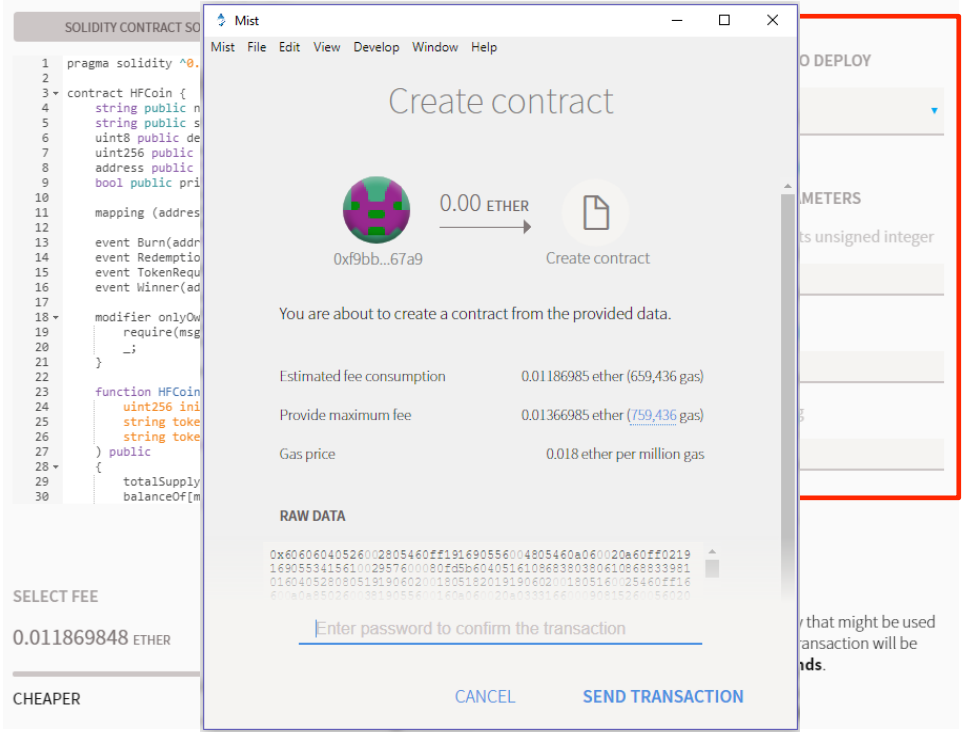

```
contract HFCoin {
    string public name;
    string public symbol;
    mapping (address => uint256) public balanceOf;

    function HFCoin(
        uint256 initialSupply,
        string tokenName,
        string tokenSymbol
    ) public
    {
        balanceOf[msg.sender] = initialSupply;
        name = tokenName;
        symbol = tokenSymbol;
    }

    function transfer(address _to, uint256 _value) public {
        require(balanceOf[msg.sender] >= _value);
        require((balanceOf[_to] += _value) >= balanceOf[_to]);
        balanceOf[msg.sender] -= _value;
        balanceOf[_to] += _value;
    }
}
```

Smart Contract Example: High-Five Coin


```
contract HFCoin {
    string public name;
    string public symbol;
    mapping (address => uint256) public balanceOf;

    function HFCoin(
        uint256 initialSupply,
        string tokenName,
        string tokenSymbol
    ) public
    {
        balanceOf[msg.sender] = initialSupply;
        name = tokenName;
        symbol = tokenSymbol;
    }

    function transfer(address _to, uint256 _value) public {
        require(balanceOf[msg.sender] >= _value);
        require((balanceOf[_to] += _value) >= balanceOf[_to]);
        balanceOf[msg.sender] -= _value;
        balanceOf[_to] += _value;
    }
}
```

Deploying Smart Contracts

- Call initialization function with starting values
- Send transaction containing bytecode
- Every transaction costs a fee
  - Discourages frivolous or malicious transactions
- Add Contract address to wallet for tracking



Advanced Smart Contracts - CryptoKitties

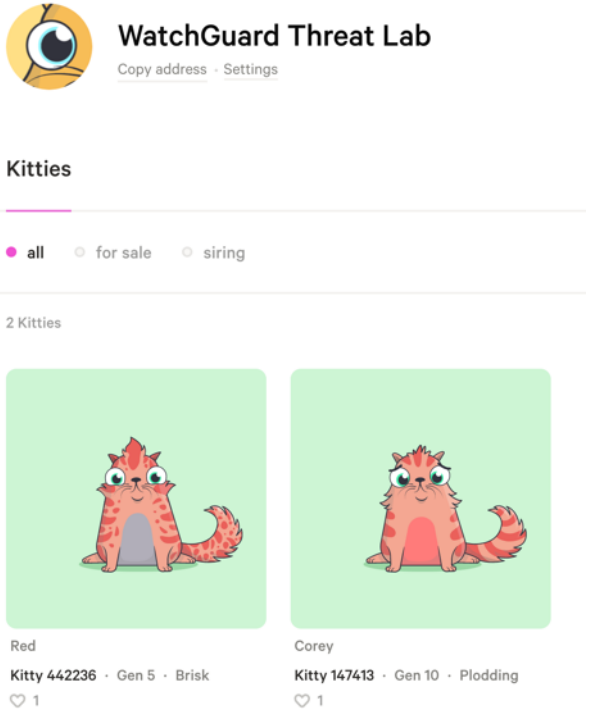
- More than just currency
- Complete applications

Upsides:

- Public
- Immutable

Downsides:

- Public
- Inefficient



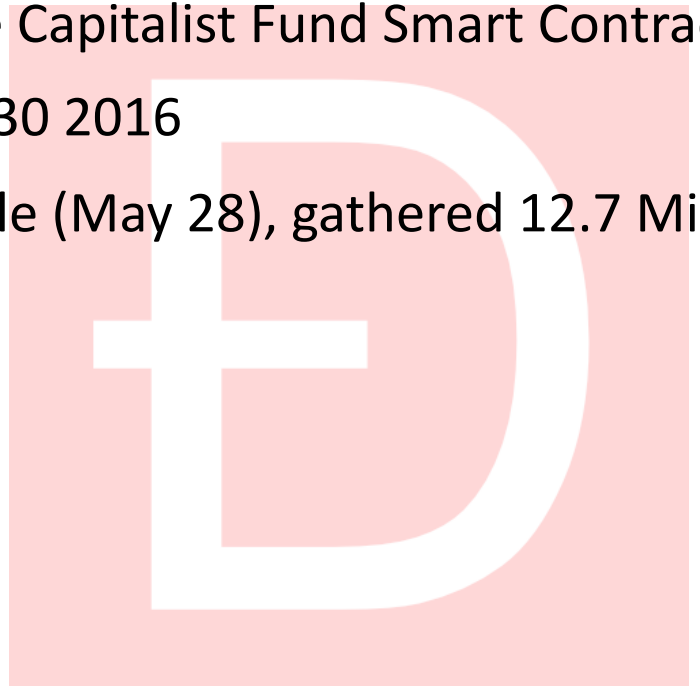
Advanced Smart Contracts - CryptoKitties


```
contract KittyBase is KittyAccessControl {

    event Birth(address owner, uint256 kittyId, uint256 matronId, uint256 sireId, uint256 genes);
    event Transfer(address from, address to, uint256 tokenId);

    struct Kitty {
        uint256 genes;
        uint64 birthTime;
        uint64 cooldownEndBlock;
        uint32 matronId;
        uint32 sireId;
        uint32 siringWithId;
        uint16 cooldownIndex;
        uint16 generation;
    }

    Kitty[] kitties;

    mapping (uint256 => address) public kittyIndexToOwner;
    mapping (address => uint256) ownershipTokenCount;
    mapping (uint256 => address) public kittyIndexToApproved;
    mapping (uint256 => address) public sireAllowedToAddress;
    SaleClockAuction public saleAuction;
    SiringClockAuction public siringAuction;

    function _transfer(address _from, address _to, uint256 _tokenId) internal {
        ownershipTokenCount[_to]++;
        kittyIndexToOwner[_tokenId] = _to;
        if (_from != address(0)) {
            ownershipTokenCount[_from]--;
            delete sireAllowedToAddress[_tokenId];
            delete kittyIndexToApproved[_tokenId];
        }
        Transfer(_from, _to, _tokenId);
    }
```

ETHEREUM VULNERABILITIES AND HACKS

Advanced Smart Contracts - The DAO

- Blockchain Venture Capitalist Fund Smart Contract
- Launched on April 30 2016
- By end of crowd sale (May 28), gathered 12.7 Million Ether ($150MM back then)



The DAO SplitDAO() function


```
function splitDAO(
  uint _propsalID,
  address _newCurator
) noEther onlyTokenholders returns (bool _success) {
...
  withdrawRewardFor(msg.sender);
  totalSupply -= balanes[msg.sender];
  balances[msg.sender] = 0;
...
}
```

The DAO SplitDAO() function


```
function splitDAO(
  uint _propsalID,
  address _newCurator
) noEther onlyTokenholders returns (bool _success) {
...
  withdrawRewardFor(msg.sender);
  totalSupply -= balanes[msg.sender];
  balances[msg.sender] = 0;
...
}
```

The DAO SplitDAO() function


```
function withdrawRewardFor(address _account) noEther internal returns
(bool _success) {
...
if(!rewardAccount.payOut(_account, reward))
  throw;
...
}
```

The DAO SplitDAO() function


```
function payOut(address _recipeint, uint _amount) returns (bool) {
...
  if (_recipient.call.value(_amount)()) {
    PayOut(_recipient, _amount);
    return true;
  }
...
}
```

The DAO SplitDAO() function


```
function() {
  call TheDAO.splitDAO(...)
}
```

The DAO SplitDAO() function


```
function splitDAO(
  uint _propsalID,
  address _newCurator
) noEther onlyTokenholders returns (bool _success) {
...
  withdrawRewardFor(msg.sender);
  totalSupply -= balanes[msg.sender];
  balances[msg.sender] = 0;
...
}
```

The DAO Hacked


Digital Money Fund Hacked

Roughly $55 million tal currency called Ethereum stakeholder to create an identi- early-morning hours on Friday

- June 18th
- Hacker drained 3.6 Million Ether ($70MM)
- Got around maximum transaction stack size

The DAO Hack Fallout

- Splitting from the DAO triggered a 28-day waiting period
- The Ethereum community had time to decide how to handle it
- 89% of community voted to hard-fork
  - All stolen Ether forcibly transferred to a new contract account
  - Victims could withdraw their stolen Ether from the new account
- Un-forked blockchain renamed Ethereum Classic

Initial Coin Offering

- Similar to an IPO
- Used to raise capital for projects
- Participants exchange Ether for a token at an increasing price
- Participants can cash out their token for the product in the future or trade them for Ether at a higher price
- $3,700,628,293 raised through ICOs in 2017

Insurex ICO Hack

- July 13, 2017
- Just prior to ICO, hacker compromised Insurex's Twitter account
- Posted fake ICO pre-sale address
- Stole 1106 Ether ($409K in today's valuation)

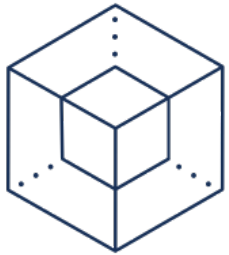
CoinDash ICO Hack

- July 17, 2017
- Hacker modified the ICO address on CoinDash's website
- 37,000 Ether sent to fake address ($13.7MM in today's valuation)


ASHLEY MADISON®
Life is short. Have an affair.®
SEE YOUR MATCHES

Enigma ICO Hack

- Attackers hacked Enigma's Slack and website
  - Used the CEO's credentials, leaked in the Ashley Madison hack…
- Convinced participants to send Ether to the wrong address for a special pre-sale
- Stole 1492 Ether ($550K in today's valuation)



enigma

Multi-Signature Wallets

- Normal wallets can sign transactions with approval of their owner
- Multi-Sig wallets require multiple approvals before signing transactions
- Uses:
  - Rudimentary multifactor for transactions
  - Can be used for company accounts with multiple controllers

Parity Multi-Sig Wallet Hack


```
function initWallet(address[] _owners, uint _required, uint _daylimit) {
    initDaylimit(_daylimit);
    initMultiowned(_owners, _required);
  }
```

Parity Multi-Sig Wallet Hack

- Attacker found vulnerability that let them re-initialize wallets with them as the owner.
- Manually stole over 150,000 Ether from several wallets, starting with most valuable
- White Hat Group wrote a script to exploit the same vulnerability and drain funds from all remaining wallets
  - Drained 377,105 Ether
  - $122MM (at the time) in secondary tokens

White Hat Group

- Returned 100% of the funds by July 31 2017
- Paid transaction fees with donations

WHG **A Modified Version of a Common Multisig Had A Vulnerability - The WHG Took Action & Will Return the Funds** (self.ethereum)

submitted 8 months ago * (last edited 8 months ago) by jbaylina

The White Hat Group were made aware of a vulnerability in a specific version of a commonly used multisig contract. This vulnerability was trivial to execute, so they took the necessary action to drain every vulnerable multisig they could find as quickly as possible. Thank you to the greater Ethereum Community that helped finding these vulnerable contracts.

The White Hat account currently holding the rescued funds is https://etherscan.io/address/0x1dba1131000664b884a1ba238464159892252d3a[1]

If you hold a multisig contract that was drained, please be patient. We will be creating another multisig for you that has the same settings as your old multisig but with the vulnerability removed and we will return your funds to you there. We will be using the donations sent to us from The DAO Rescue to pay for gas.

Effectively we will upgrade your multisig contract for you, all you will have to do is, be patient, find your new multisig address once we have finished, and it will be like nothing happened.

We will not be responding to any social media posts.

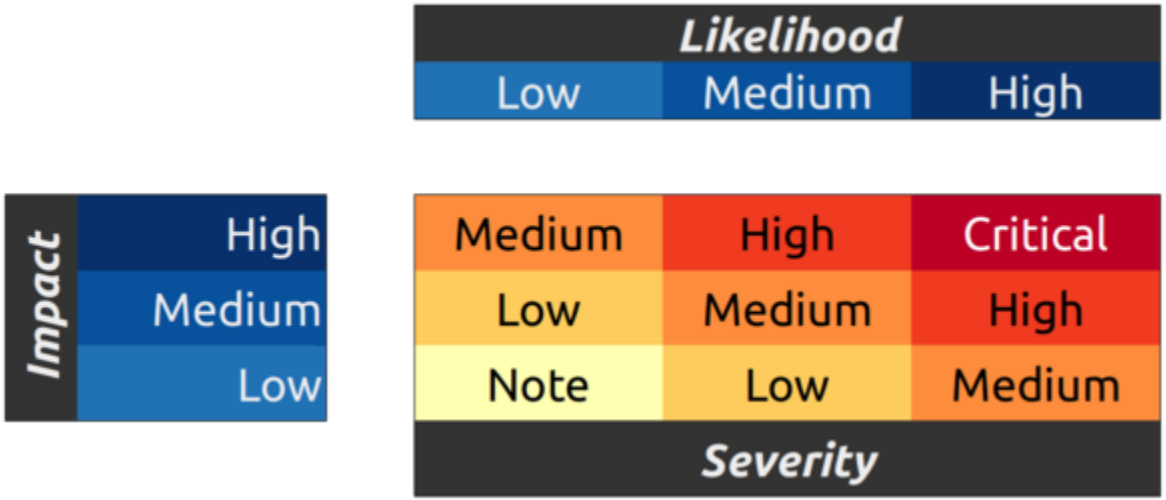
Edit: Do not trust any address posted below as a ¨donation address.¨ There are a lot of phishers in the community right now. In general always verify any address or link you find on reddit.

Parity Multisig Wallet Hack 2.0

- November 6, 2017
- Attacker re-initialized the shared Parity code library used by wallets
- Issued a self-destruct command, terminating the library
- Locked 519,774 Ether (near $200MM!!!)

HOW DO WE SECURE ETHEREUM?

Bug Bounties – Ethereum Foundation



- **Critical**: up to 25,000 points
- **High**: up to 15,000 points
- **Medium**: up to 10,000 points
- **Low**: up to 2,000 points
- **Note**: up to 500 points

1 point = $1 USD in ETH or BTC

Bug Bounties – Third Party

- Most ICOs now include a bug bounty before any presales
  - Usually $100 to $10,000 in ETH as rewards

- Major applications (e.g. Parity) have substantial bug bounties now

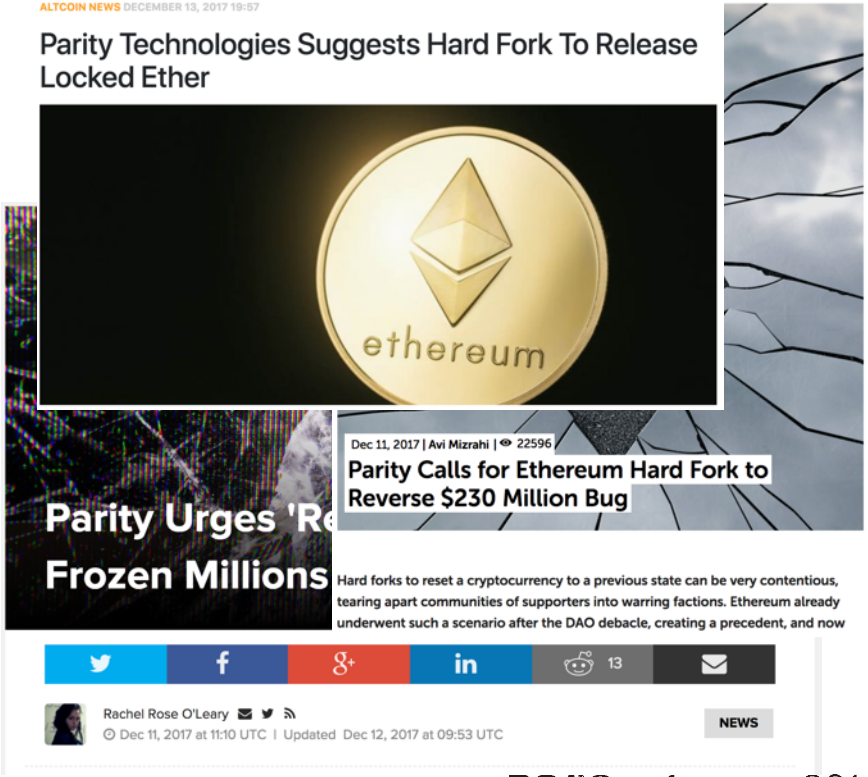
EtherscamDB

- EtherScamDB.info
- Open source database of ongoing scams relating to Ethereum
  - Created by the MyEtherWallet team
  - Keeps track of active phishing scams

MyEtherWallet

To Fork or Not To Fork

- The DAO
  - Hard fork in the early life of Ethereum
  - Concerns about setting a precedent
  - Concerns about censorship
- Parity Wallet
  - No plans for hard fork
  - Fewer people comparatively affected



Is Code Law?

- Traditional Contract
  - Letter of the law (verbiage) is more malleable
  - Precedents to solve contract conflict in court

- Digital Smart Contract
  - Programming (verbiage) is of upmost importance
  - Who solves conflict in a decentralized system?

WHAT ABOUT OTHER CRYPTOCURRENCIES?

Bitcoin Issues – Arbitrary Data Insertion

- OP_RETURN allows embedded data

```
static const unsigned int MAX_OP_RETURN_RELAY = 80;      //! bytes
```

Research by RWTH Aachen University

Bitcoin Issues – Arbitrary Data Insertion

- OP_RETURN allows embedded data

```
"vout" : [
    {
        "value" : 0.00000000,
        "n" : 0,
        "scriptPubKey" : {
            "asm" : "OP_RETURN 636861726c6579206c6f766573206865696469",
            "hex" : "6a13636861726c6579206c6f766573206865696469",
            "type" : "nulldata"
        }
    },
```

Research by RWTH Aachen University

Bitcoin Issues – Arbitrary Data Insertion

- OP_RETURN allows embedded data

```
"vout" : [
    {
        "value" : 0.00000000,
        "n" : 0,
        "scriptPubKey" : {
            "asm" : "OP_RETURN 636861726c6579206c6f766573206865696469",
            "hex" : "6a13636861726c6579206c6f766573206865696469",
            "type" : "nulldata"
        }
    },
```

Research by RWTH Aachen University

Bitcoin Issues – Arbitrary Data Insertion

- OP_RETURN allows embedded data

```
"vout" : [
    {
        "value" : 0.00000000,
        "n" : 0,
        "scriptPubKey" : {
            "asm" : "OP_RETURN 636861726c6579206c6f766573206865696469",
            "hex" : "6a13636861726c6579206c6f766573206865696469",
            "type" : "nulldata"
        }
    },
```

Research by RWTH Aachen University

Bitcoin Issues – Arbitrary Data Insertion

- OP_RETURN allows embedded data

```
"vout" : [
    {
        "value" : 0.00000000,
        "n" : 0,
        "scriptPubKey" : {
            "asm" : "OP_RETURN 636861726c6579206c6f766573206865696469",
            "hex" : "6a13636861726c6579206c6f766573206865696469",
            "type" : "nulldata"
        }
    },
```

636861726c6579206c6f766573206865696469 == charley loves heidi

Research by RWTH Aachen University

Bitcoin Issues – Arbitrary Data Insertion

- OP_RETURN allows embedded data

> **Illegal and Condemned Content.** Bitcoin's blockchain contains **at least eight files with sexual content**. While five files only show, describe, or link to mildly pornographic content, we consider the remaining three instances objectionable for almost all jurisdictions: Two of them are backups of **link lists to child pornography**, containing 274 links to websites, 142 of which refer to Tor hidden services. The remaining instance is an **image depicting mild nudity of a young woman.** In an online forum this image is claimed to show child pornography, albeit this claim cannot be verified (due to ethical concerns we refrain from providing a citation). Notably, two of the explicit images were only detected by our suspicioustransaction detector, i.e., they were not inserted via known services. While largely harmless, potentially objectionable blockchain content is infrequently inserted, e.g., links to alleged child pornography or privacy violations. We thus believe that future blockchain designs must proactively cope with objectionable content. Peers can, e.g., filter incoming transactions or revert contentholding transactions [11,51], but this must be scalable and transparent.

Research by RWTH Aachen University

VERGE – 51% Attack

- VERGE
  - #22 in market cap
  - Privacy-focused cryptocurrency
  - Uses rotating algorithms for each block
- April 4, 2018
- Attacker spoofed timestamps of mined blocks
  - Used same algorithm (scrypt) over and over
  - Gave attacker disproportionate mining power (>51%)
- VERGE hard-fork fix

WRAPPING UP

Takeaways

Developers

- Avoid External Calls
  - Untrusted Contracts = Unexpected Errors
- Mark Visibility In Contract Functions
  - Be aware of who can call functions

Analysts

- Look for Bug Bounties
  - Great way to contribute to the community
- Test Everything!
  - Every function
  - Every input

High Five Coin

- Its real – **0x191a70e9808c8d89Be289Cfe9001A7010Dc3D78c**
  - Twitter: **@Xorro_**
- You can request up to 10 coins
- You can redeem coins for a crisp high five
- There is a vulnerability in the smart contract
- First person to exploit and redeem **1337** coins in one transaction earns some Ether

Contact Info

- Email: Marc.Laliberte@WatchGuard.com
- Twitter: @XORRO_
- LinkedIn: /in/marc-Laliberte
- Secplicity.org