Who Spent My EOS?

On the (In)Security of Resource Management of EOS.IO

Sangsup Lee, Daejun Kim, Dongkwan Kim, Sooel Son, Yongdae Kim @KAIST
Abstract

2K+ Cryptocurrencies
Resource management of EOS.IO
Abstract

Resource management of EOS.IO

4 unique vulnerabilities
Abstract

Evaluated the impact of each vulnerability

4 unique vulnerabilities

Resource management of EOS.IO
Background
Overview of cryptocurrency components

- Consensus
- Mining
- P2P
- Blockchain-based cryptocurrency
- Smart contract
- Wallet
- Network
Background: Blockchain

Key components

- Consensus
- Smart contract
- Blockchain-based cryptocurrency
- Mining
- Wallet
- P2P
- Network
- ...
The fundamentals of blockchain
Background: Blockchain

Consensus algorithm

Data (Block)

Creating blocks
Background: Blockchain

Consensus algorithm

Creating blocks

Verifying blocks

Data (Block)
Background: Blockchain

Consensus algorithm

Creating blocks

Verifying blocks

Agreement on blocks
Consensus algorithm (PoW)

Numerous block producers

Create Agree

Data (Block)

Slow...

Bitcoin

Ethereum
Background: DPoS (Delegated Proof of Stake)

EOS.IO Consensus algorithm (DPoS)

Elected 21 Block producers (BP)

Create → Agree → Data (Block)

FAST! (0.5 sec / block)
But, resource management matters.
Resource management necessity

Background: DPoS (Delegated Proof of Stake)
Resource management necessity

User

Transaction requests

...
Resource management necessity

Transaction requests

User

Blockchain

Overload problem

Properly process request

Elected 21 Block producers (BP)

Background: DPoS (Delegated Proof of Stake)
Background: Smart contract

Smart contract

Use Case

- Exchange
- Gambling
- Auction
- Funding
- Bank
- And so on.

Alice $ Transaction Transaction Transaction Transaction Bob
Background: Smart contract on EOS.IO

- Target (Ex. eBay)
- Function (Ex. Bidding(), Selling())
- Permission (Ex. Alice@active)
Background: Smart contract on EOS.IO

Delegated execution

BP
Background: Smart contract on EOS.IO

Delegated Execution

Resource management matters
Background: Resource of EOS.IO

Transaction delivery  Program execution  Data storing

NET  CPU  RAM
Background: Resource of EOS.IO

- **Transaction delivery**
  - NET

- **Program execution**
  - CPU

- **Data storing**
  - RAM

**Staking**

**Individuals**

**Blockchain system**

Refreshed every day
Background: Resource of EOS.IO

Transaction delivery

Program execution

Data storing

NET

CPU

RAM

Buy

Individuals

RAM Market

Not refreshed every day.
Why EOS?
Why EOS?

**Smart contract research**
- Making smart contracts smarter (ACM CCS '16)
- ZEUS: Analyzing Safety of Smart Contracts (NDSS '18)
- teether: Gnawing at ethereum to automatically exploit smart contracts (USENIX '18)

**Consensus research**
- The miner's dilemma (IEEE S&P '15)
- Be Selfish and Avoid Dilemmas: Fork After Withholding (FAW) Attacks on Bitcoin (ACM CCS '17)
- Publish or perish: A backward-compatible defense against selfish mining in bitcoin (RSA '17)

**Other research work**
- Porosity: A decompiler for blockchain-based smart contracts bytecode (Defcon '17)
- Hijacking bitcoin: Routing attacks on cryptocurrencies (IEEE S&P '17)
- Eclipse attacks on bitcoin’s peer-to-peer network (USENIX '15)

**Market cap**
- #1
- #2

2K+ Cryptocurrencies
<table>
<thead>
<tr>
<th>Rank of marketcap</th>
<th>Name</th>
<th>Consensus algorithm</th>
<th>Smart contract platform</th>
<th>User accounts</th>
<th>Why EOS?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Bitcoin</td>
<td>PoW</td>
<td>X</td>
<td>\approx 1.3 M</td>
<td>But, no security research in academia.</td>
</tr>
<tr>
<td>2</td>
<td>Ethereum</td>
<td>PoW</td>
<td>O</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Ripple</td>
<td>PoS</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Litecoin</td>
<td>PoW</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Bitcoin cash</td>
<td>PoW</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Binance Coin</td>
<td>DPoS</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>EOS</td>
<td>DPoS</td>
<td>O</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
In our paper...
What are new attack targets?

- **Transaction (User)**
- **Transaction (SC)**

Block producer

- **Users**
- **program code**

EOS structure
EOS structure

What are new attack targets?

Transaction (User)
Transaction (SC)

Block producer

Smart Contract

Smart Contract
What are new attack targets?

Transaction (User)
Transaction (SC)

Block producer

Smart Contract

Smart Contract

[]@eosio.code

Users
What are new attack targets?
What are new attack targets?

EOS structure
What are new attack targets?

Audit Target

Transaction (User)

Transaction (SC)

Block producer

Block creation time

Smart Contract Provider (SCP)

Users

[]@eosio.code
What are new attack targets?

- Transaction (User)
- Transaction (SC)

Block producer

- B
  - NET
  - CPU
  - RAM
- A
  - NET
  - CPU
  - RAM

[]@eosio.code

Pay for transaction

block creation time
Attack Target

What are new attack targets?

Transaction (User) → Block producer → Transaction (SC)

Block producer

B

NET
CPU
RAM

A

NET
CPU
RAM

= @eosio.code

Smart Contract Provider (SCP)

Users

CPU
RAM

CPU
RAM

Block creation time

Pay for transaction

To save data
What are new attack targets?

Transaction (User) → Block producer
Transaction (SC) → Block producer

Block creation time
Pay for transaction
To save data
grant permission to SCP

Smart Contract Provider (SCP)
CPU  RAM

Users
CPU  RAM

[ ]@eosio.code
We found …
We found …

Transaction (User) → Block producer → Transaction (SC)

Block delay attack

Smart Contract Provider (SCP)

Users

[@eosio.code]
We found ...

Transaction (User) → Block producer
Transaction (SC) → Block producer

Block delay attack

CPU-Drain attack

Smart Contract Provider (SCP) → Users

[@eosio.code]
Attack Models & Threat Models & Attacks!

We found …

Transaction (User) → Block producer → Transaction (SC)

Block delay attack

CPU-Drain attack

RAM-Drain attack

Smart Contract Provider (SCP)

Users

CPU
RAM
We found …

**Attack Models & Threat Models & Attacks!**

Transaction (User)  Transaction (SC)

Block producer

**Block delay attack**

- CPU-Drain attack
- RAM-Drain attack
- RAMsomware attack

Smart Contract Provider (SCP)

Users

- RAMcode

CPU  RAM

CPU  RAM
Attack
Block delay attack

Block delay attack | DoS by draining EOS resources | RAMsomware attack

Block producer

Timer (T) 0.5s 0.5s
Succeeded state
Exhausted state

Transactions (trx)

Queue
Block delay attack
Block delay attack

Block producer

Timer (T) | T+0.5s | T+0.5s | T+0.5s | T+0.5s | T+0.5s
---|---|---|---|---|---
Succeeded state
Exhausted state

Block delay attack | DoS by draining EOS resources | RAMsomware attack
# Block delay attack

Block delay attack | DoS by draining EOS resources | RAMsomware attack

<table>
<thead>
<tr>
<th>Timer (T)</th>
<th>T+0.5s</th>
<th>T+0.5s</th>
<th>T+0.5s</th>
<th>T+0.5s</th>
<th>T+0.5s</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Succeeded state</strong></td>
<td><img src="image" alt="Block producer" /></td>
<td><img src="image" alt="Queue" /></td>
<td><img src="image" alt="Transactions" /></td>
<td><img src="image" alt="Block" /></td>
<td></td>
</tr>
<tr>
<td><strong>Exhausted state</strong></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
<td><img src="image" alt="Image" /></td>
</tr>
</tbody>
</table>
Block delay attack

DoS by draining EOS resources | RAMsomware attack

Succeeded state

Exhausted state

Timer (T)  T+0.5s  T+0.5s  T+0.5s  T+0.5s  T+0.5s

Queue

Block producer

Transactions (trx)

Block delay attack
Block delay attack

- DoS by draining EOS resources
- RAMsome attack

Block producer

<table>
<thead>
<tr>
<th>Timer (T)</th>
<th>T+0.5s</th>
<th>T+0.5s</th>
<th>T+0.5s</th>
<th>T+0.5s</th>
</tr>
</thead>
<tbody>
<tr>
<td>Succeeded status</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Exhausted status</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

A

- transaction Call(A)
- transaction Call(A)

SC1
- Action (sc)
- Action (sc)

SC2
- Action (sc)
- Action (sc)

SC3
- Action (sc)
- Action (sc)

SC4
- Action (sc)
- Action (sc)

SC5
- Action (sc)
- Action (sc)

SC6
- Action (sc)
- Action (sc)

SC7
- Action (sc)
- Action (sc)
Block delay attack

DoS by draining EOS resources | RAMsomware attack

- Timer (T)
- Succeeded state
- Exhausted state

Expected time $T = T + 0.5^x$

Real time: $T = T + 0.2^x$

Block creation delay time

$0.5^x - 0x2^x$
## Block delay attack

### Estimated financial loss via block delay attack

<table>
<thead>
<tr>
<th>Block Count</th>
<th>Time (min)</th>
<th>Eos-CPU (min)</th>
<th>EOS-NET (MiB)</th>
<th>Cost (EOS)</th>
<th>Delay Time (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>376</td>
<td>0.92</td>
<td>1.23</td>
<td>16.13</td>
<td>480</td>
<td>2.05</td>
</tr>
<tr>
<td>704</td>
<td>2.06</td>
<td>2.32</td>
<td>34.72</td>
<td>910</td>
<td>3.56</td>
</tr>
<tr>
<td>1106</td>
<td>3.02</td>
<td>3.65</td>
<td>50.82</td>
<td>1,426</td>
<td>5.67</td>
</tr>
<tr>
<td>1471</td>
<td>4.00</td>
<td>4.85</td>
<td>65.53</td>
<td>1,894</td>
<td>7.46</td>
</tr>
<tr>
<td>1840</td>
<td>5.04(min)</td>
<td>6.07</td>
<td>79.69</td>
<td>2,368</td>
<td>9.12(min)</td>
</tr>
</tbody>
</table>

181,518 EOS == $880,000 USD  
(29/7/2019)

Average of EOS transfer volume 
(01/04/2019~30/04/2019)
## Block delay attack

**Block delay attack** | DoS by draining EOS resources | RAMsomware attack

### Estimated financial loss via block delay attack

<table>
<thead>
<tr>
<th>Block Count</th>
<th>Time (min)</th>
<th>Eos-CPU (min)</th>
<th>Cost (EOS)</th>
<th>Delay Time (min)</th>
<th>Loss (EOS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>376</td>
<td>0.92</td>
<td>1.23</td>
<td>16.13</td>
<td>2.05</td>
<td>40,802</td>
</tr>
<tr>
<td>704</td>
<td>2.06</td>
<td>2.32</td>
<td>34.72</td>
<td>3.56</td>
<td>70,856</td>
</tr>
<tr>
<td>1106</td>
<td>3.02</td>
<td>3.65</td>
<td>50.82</td>
<td>5.67</td>
<td>112,851</td>
</tr>
<tr>
<td>1471</td>
<td>4.00</td>
<td>4.85</td>
<td>65.53</td>
<td>7.46</td>
<td>148,478</td>
</tr>
<tr>
<td>1840</td>
<td>5.04(min)</td>
<td>6.07</td>
<td>79.69</td>
<td>9.12(min)</td>
<td>181,518</td>
</tr>
</tbody>
</table>

*We get maximum bug bounty ($10,000 USD)*
*From EOSIO foundation*

---

표의 맨 아래쪽을 보면 5분동안 dummy transaction을 잔류 시키면, 약 9분간 block이 생성이 멈추는 결과를 얻을 수 있었다. block 생성이 멈추면 EOS의 모든 transaction 처리가 되지 않으며, 그 잠재적 손실은 191,518 EOS 이다.
DoS by draining EOS resources: RAM-drain attack

EOS-RAM is purchase resource not stake, so EOS-RAM doesn’t return until finishing their propose
DoS by draining EOS resources: RAM-drain attack

EOS-RAM is purchase resource not stake, so EOS-RAM doesn't return until finishing their propose
DoS by draining EOS resources: RAM-drain attack

The consuming time for RAM is depends on Smart contract’s source code.
DoS by draining EOS resources: CPU-drain attack

Users

EOS-CPU: 1,000 ms
EOS-RAM: 1 MB
EOS-NET: 1,000 bps

Create new transaction

EOS-CPU: 500 ms
EOS-RAM: 1 MB
EOS-NET: 500 bps

Block delay attack | DoS by draining EOS resources | RAMsomware attack
DoS by draining EOS resources: CPU-drain attack

Users

EOS-CPU: 700 ms
EOS-RAM: 1 MB
EOS-NET: 700 bps

Create new transaction

EOS-CPU: 0 ms
EOS-RAM: 1 MB
EOS-NET: 0 bps

Cost {EOS-CPU of $A} + Cost {EOS-NET of $A} < Cost {EOS-CPU of $B} + Cost {EOS-NET of $B}

Block delay attack | DoS by draining EOS resources | RAMsomware attack
# DoS by draining EOS resources: CPU-drain attack

<table>
<thead>
<tr>
<th>Attack Count</th>
<th>EOS-NET (KiB)</th>
<th>EOS-CPU (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.137</td>
<td>0.400</td>
</tr>
<tr>
<td>10</td>
<td>1.329</td>
<td>4.366</td>
</tr>
<tr>
<td>20</td>
<td>2.655</td>
<td>8.352</td>
</tr>
<tr>
<td>50</td>
<td>6.626</td>
<td>20.47</td>
</tr>
<tr>
<td>100</td>
<td>13.21</td>
<td>41.19</td>
</tr>
</tbody>
</table>

Over x3 (times)

**Attacker partially make DoS to victim while a day**
RAMsomware attack

Block delay attack | DoS by draining EOS resources | RAMsomware attack

Users
Give EOSIO.CODE

Block producer

A
Call(B)

B
Normal Action
RAMsomware attack

[User]@eosio.code

Smart Contract Provider (SCP)

Users

CPU

RAM
RAMsomware attack

Block delay attack | DoS by draining EOS resources | RAMsomware attack

Users

JUST SEND TRANSACTION

Block producer

A

Call(A)

loop

SAVE DATA

A

Use EOS-RAM

User EOS-RAM is drained

Attacker demand ransom of EOS-RAM to the victim
RAMsomware attack

The user who have the largest EOS-RAM have 2GB EOS RAM

2GB EOS-RAM == $800,000 USD
Defense
Defense

Trivial solution

- Block delay attack
- CPU/RAM drain attack
- RAMsomware attack

- Patched by EOSIO developers
- Do access control
- Do check smart contract version
Defense

**Trivial solution**
- Block delay attack
- CPU/RAM drain attack
- RAMsomware attack

**Design solution**
- Fine graind permission
  : eosio.code Expire Time, Maximum EOS Coin per a transaction
  : EOS-CPU permission, EOS-NET permission, EOS-RAM permission etc...

- Totally payment of transaction fee to the first transaction creator
  : Every transaction that purpose a role, is payed by the users who start trx.

- Patched by EOSIO developers
- Do access control
- Do check smart contract version
Conclusion & Future work

- Conclusion
  - Analyzed new threats from the view point of new resources in EOS.IO
  - Found 4 new attack methodologies and verified them
  - Proposed new security features to prevent our attacks

- Future work
  - Make an automatic auditing tool for our attacks
  - Design a web assembly analyzer
Thank you

{k1rh4, reset, dkay, sl.son, yongdaek}@kaist.ac.kr