Contents

Introduction.......................................................................................................................... 3
Description of the vulnerability......................................................................................... 3
Exploitation ........................................................................................................................ 3
Threat ................................................................................................................................ 4
Solution to the vulnerability ............................................................................................. 4
References ......................................................................................................................... 6
Introduction

ColossusXT is an open source cryptocurrency which allows people to make fast and anonymous transactions that are non-government controlled [1]. ColossusXT is a proof-of-stake cryptocurrency which is chain-based [2].

In proof-of-stake based cryptocurrencies the next blocks creator is selected by different combinations of random selection and age or wealth [3].

ColossusCoinXT is a desktop wallet for the ColossusXT cryptocurrency [6].

Description of the vulnerability

CVE-2018-19158 allows a remote denial of service which is exploitable if the attacker has some stake in the system. The vulnerability affected ColossusXT version 1.0.5 and versions prior to it. [2]

The attack is a resource exhaustion attack. It consists of sending malformed chains of invalid headers or blocks which are stored either on RAM or on the disk without any proper validation. [4]

A node must determine if the blocks received are valid to mine the valid block chain which is the largest. The node maintains the headers in an in-RAM data structure mapBlockIndex and the blocks are stored in block files which are on the disk. [4]

Exploitation

Attack on RAM – It is necessary to create fake block headers that are stored in mapBlockIndex. To do that an arbitrary fork point in the blockchain has to be picked. Next a header which extends the block must be constructed. To avoid being disconnected while doing so the chain of false headers has to be strictly shorter than the current main chain. [4]

![Algorithm 1](image)

Figure 1. Algorithm 1 – RAM attack [4]
Attack on disk – It is necessary to create a chain of fake blocks that are stored in mapBlockIndex and in the blocks database. To do that a chain of blocks has to be created which starts from an arbitrary fork point in history and matches exactly the length of the main chain. Then the blocks must be broadcast but before doing that it is necessary to broadcast only the headers, so the victim would request all the blocks. [4]

Algorithm 2: Disk attack nothing at stake

```plaintext
Algorithm 2: Disk attack nothing at stake
1: procedure DISK_ATTACK(target_peer)
2: block ← empty
3: blockcount ← getblockcount()
4: depth ← rand(1, MAX_HEADERS_DEPTH)
5: pastblock_header ← getblockheader(blockcount - depth)
6: nTime ← pastblock_header.nTime
7: while target_peer.alive() do:
8:     prehash ← pastblock_header.hash
9:     headers, blocks ← [], []
10:     for d in range(depth) do:
11:         nTime ← nTime + block_interval*d
12:         nVersion ← CURRENT_BLOCK_VERSION
13:         nBits ← get_next_difficulty_bits()
14:         merklehash, nonce ← rand32_bytes(), rand4_bytes()
15:         block ← block_header(nVersion, prehash, merklehash, nTime, nBits, nonce)
16:         for j in range(MAX_TX) do:
17:             prev_tx, prev_index ← rand32_bytes(), 0
18:             scriptPubKey, amount ← b” “, rand_amount
19:             tx ← create_transaction(prev_tx, prev_index, scriptPubKey, amount)
20:             block.append(tx)
21:         block.rehash()
22:         prehash ← block.hash
23:         headers.append(block:header)
24:         blocks.append(block)
25:         send_msg_headers(target_peer, headers)  // Wait for peer to request blocks
26:     send_msg_blocks(target_peer, blocks[-1])
```

Figure 2. Algorithm 2 – Disk attack

Threat

Resource exhaustion attacks are considered critical vulnerabilities [4]. It is a threat because an attacker can overwhelm the existing nodes and crash them, so his node could have the majority of stake [5].

A successful attacker can limit the competition pool of proof-of-stake, so that he gains disproportionate wealth, the attacker can increase his own stake through someone else’s account and by doing so they make the victims account less profitable for the victim. [5]

Solution to the vulnerability

The vulnerability was fixed in build 1.2.1 which was released on 29th January 2019 and the last modification took place on 25th February 2019. [7]

The vulnerability is checked in method UpdateState which will return false if the node is an attacking one and then it will discard the node and return an error. [7]
Figure 3. UpdateState method. [7]

The received node is checked with banNode which will be assigned the value true if the node is an attacking one. BanNode can only be true if one of three given conditions hold. The given conditions check whether the node is formed correctly, if its size, average value, number of blocks is within the correct range, whether they exceed the max size or the max average. [7]
References

1. https://colossusxt.io/
6. https://github.com/ColossusCoinXT
7. https://github.com/ColossusCoinXT/ColossusCoinXT/compare/0223904...9666bb8