New issue   Jump to bottom

The node may have a bug when dealing with unformatted packet and lead to a crash #1951

Closed   **fCorleone** opened this issue on Jun 15, 2021 · 2 comments

Assignees

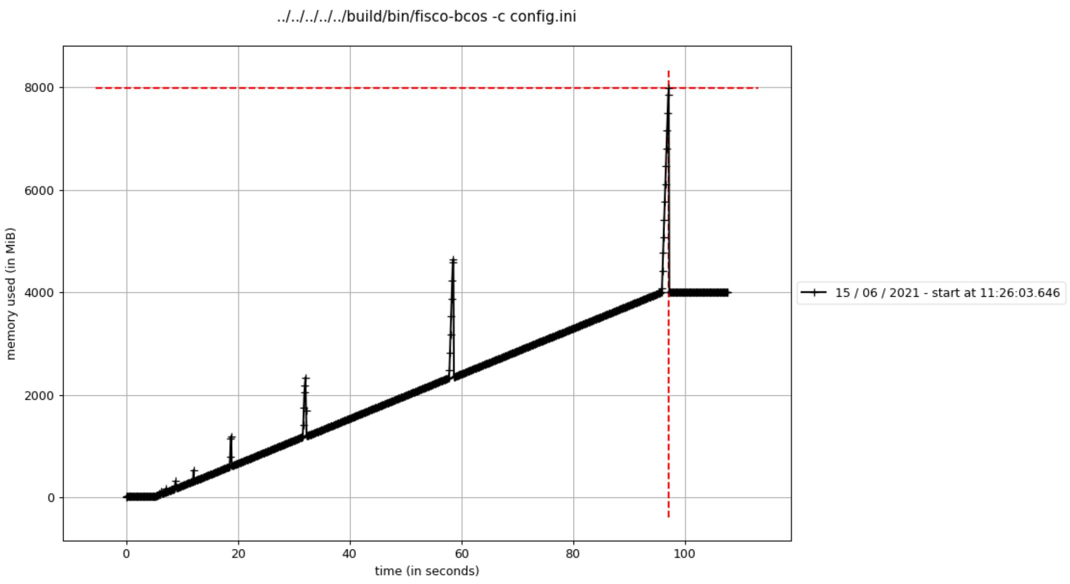
Labels   release-2.8.0-fixed   vulnerabilities   wontfix

Projects   FISCO BCOS

fCorleone commented on Jun 15, 2021

Describe the bug

A malicious node can send a packet continuously. The packet is in an incorrect format and cannot be decoded by the node correctly. As a result, the node may consume the memory sustainably, as the flowing figure shows:



After 100 seconds, over 4000 MB memory has been consumed. If I continue sending the packet, the node will consume all the memory. At last it be killed by the OS.

In order to analyze the reason for this bug, I try to debug the code of the node. Here is what I found:
First, I found that in the file `libp2p/P2PMessageRC2.cpp` , at line 109 in the function `decode` :

```
ssize_t P2PMessageRC2::decode(const byte* buffer, size_t size)
{
    ...
    m_length = ntohl(*((uint32_t*)&buffer[offset]));
    if (size < m_length)    {
       return dev::network::PACKET_INCOMPLETE;
    }
    ...
}
```

the variable `size` is 72 and the variable `m_length` is a very big number under my packet. So the function will return `dev::network::PACKET_INCOMPLETE` whose value is 0.
The variable which accepts the return value is `result` in `libnetwork/Session.cpp` at line 421 in the function `doRead` :

```
ssize_t result = message->decode(s->m_data.data(), s->m_data.size());
```

and the program will enter into a if-else cluse:

```
if (result > 0){
    ...
}
else if (result == 0)  {
    s->doRead();
    break;
}
else {
    ...
}
```

Because the value of `result` is 0, so here the program will call the function `doRead` recursively. If I delete this call, the problem will not occur anymore.

```
else if (result == 0)  {
    // s->doRead();
```

```
        break;
    }
```

So I think the reason maybe the developers forget to release certain memory before the return statement if the packet is not decoded correctly!

To Reproduce
Steps to reproduce the behavior:

1. Construct a P2P packet which claims to have a big length (set a big value for variable `m_length`)
2. Continuously send the packet to a running node
3. The node will consume the memory continuously and crash.

Expected behavior
By handling the abnormal packets correctly, the memory cost will not sustainably increase and the node will not crash.

Screenshots
I have give the screenshots of the memory usage of the node in the description part.

Environment (please complete the following information):

- OS: Ubuntu 16.04
- FISCO BCOS Version: v2.7.2

Additional context
None!

cyjseagull commented on Jun 24, 2021 Contributor

We will follow up on this issue and resolve it in 2.8.0.

cyjseagull self-assigned this on Jun 28, 2021

cyjseagull added the **vulnerabilities** label on Jun 28, 2021

cyjseagull mentioned this issue on Jun 28, 2021

Fix the problem of memory growth caused by forged P2P message packets #1958

Merged

cyjseagull added the release-2.8.0-fixed label on Jun 28, 2021

stale bot commented on Jan 8

This issue has been automatically marked as stale because it has not had recent activity. It will be closed if no further activity occurs. Thank you for your contributions.

stale bot added the **wontfix** label on Jan 8

cyjseagull added this to **Done** in **FISCO BCOS** on Feb 15

stale bot closed this as completed on Mar 2

Assignees

cyjseagull

Labels

release-2.8.0-fixed vulnerabilities wontfix

Projects

FISCO BCOS
Done

Milestone

No milestone

Development

No branches or pull requests

2 participants