Publicchain security audit report

零时科技
NONEAGE

DeepBrainChain Audit Report

Audit Team：Noneage security team

Audit Date：May 19, 2021

DeepBrainChain Security Audit Report

1. Overview

The security team of Zero Hour Technology conducted a security audit on the **DeepBrainChain** project from May 10 to May 14, 2021. This audit mainly focused on the code itself, the security of the code content, reference libraries, and dependent libraries Analyzed. In this audit, there were no serious security issues in the code itself, and no security vulnerabilities that could be directly exploited and generated security issues were found, and it passed the security audit.

The safety audit result of the **DeepBrainChain** project: **Passed the audit**.

Audit Report MD5：9CBF12C2D32B862D8D73597341E58F2E

2. Project Background

2.1 Project Description

> Project Name：DeepBrainChain
>
> Project official website：https://www.deepbrainchain.org/
>
> Code warehouse：https://github.com/DeepBrainChain/DeepBrainChain-MainChain/tree/v0.2
>
> Audit version：commit e97c29ab9f2c5a10a51247d34497cb1851091250
>
> Main coding language：Rust

2.2 Audit scope

> Code warehouse：https://github.com/DeepBrainChain/DeepBrainChain-MainChain

2.3 Security audit items

The security team of Zero Hour Technology conducts security audits on the agreed security audit projects. The scope of this security audit does not include new attack methods that may appear in the future, updated or tampered code, project front-end code security, and project platform server security.

1. The security audit items include the following:
    1. Code compliance audit

       Code similarity audit

       Code patch audit

       Roadmap audit

       Recharge program audit
    2. P2P security

Node connection number audit

Node performance audit

Message format verification

Elimination policy audit

Communication encryption audit

"Alien Attack" Audit

3. RPC security

Remote call permission audit

Malformed data request audit

Communication encryption audit

Same-origin policy audit

4. Encryption signature security

Random number generation algorithm audit

Key storage audit

Cryptographic component call audit

Hash strength audit

Transaction malleability audit

Encryption and decryption fuzzing

5. Account and transaction model security

Transaction verification audit

Transaction replay audit

"Fake top-up" audit

6. Static code inspection

Built-in function safety

Standard library security audit

Third-party library security audit

Injection audit

Serialization algorithm audit

Memory leak audit

Arithmetic operation audit

Resource consumption audit

Exception handling audit

Log security audit

7. Python script security audit

8. Android APP security test

3. Architecture analysis

3.1 Directory Structure

```
    ├─bench
    |  └─src
    ├─browser-testing
    |  └─src
    ├─cli
    |  ├─bin
    |  ├─browser-demo
    |  ├─doc
    |  ├─res
    |  ├─src
    |  └─tests
    ├─docs
    |  ├─freq_ask_questions
    |  |  └─freq_ask_questions.assets
    |  ├─How_to_rent_supernode.assets
    |  ├─join_dbc_network_vm.assets
    |  ├─join_dbc_testnet.assets
    |  ├─join_dbc_testnet_EN.assets
    |  ├─prepare_vm_EN.assets
    |  └─staking_dbc_and_voting.assets
    ├─executor
    |  ├─benches
    |  ├─src
    |  └─tests
    ├─inspect
    |  └─src
    ├─pallets
    |  ├─dbc-staking
    |  |  ├─fuzzer
    |  |  |  └─src
    |  |  ├─reward-curve
    |  |  |  ├─src
    |  |  |  └─tests
    |  |  └─src
    |  └─dbc-testing
    |      └─src
    ├─primitives
    |  └─src
    ├─rpc
    |  └─src
    ├─rpc-client
    |  └─src
    ├─runtime
    |  └─src
    ├─scripts
    ├─testing
    |  └─src
    └─traits
        └─phase-reward
            └─src
```

4. Audit details

4.1 Public chain code audit

No security vulnerabilities that can be directly exploited and generate security issues are found, and the security audit is passed.

5. Security Audit Tool

Tool name	Function
Zero Hour internal toolkit	Zero Hour (Hawkeye System) self-developed toolkit
Codeql	libraries and queries that provide support for global security researchers

6. Vulnerability risk assessment standard

- **Higher Hazard**

  High hazard means that the vulnerability occurs in the core system business logic (blocks, transactions, funds, consensus verification processing and other logic involving core assets and data), causing a lot of economic losses, large areas of confusion, or obtaining nodes for the entire blockchain system Serious and most irreversible harms such as host permissions.

  including but not limited to:

  -Remote command execution on any node -Blockchain network fork -Tampering with historical block data -Forge and replay any transaction or block and benefit a lot -Get the private key hosted by any node -Arbitrary minting and stealing coins -Cause a loss of funds to any account -Tampering with core system logic such as authentication, charging, and transfer -Destroy the confidential design on the chain

  Moderate hazard

  Moderate harm refers to the problem that loopholes cause serious harm to some nodes or accounts, which can cause some blockchain systems to stagnate and cause greater chaos or economic losses.

  including but not limited to:

  -Any node program crashes or becomes unresponsive -Any node host crashes or becomes unresponsive -Make any node unable to accept legal transactions -Make any node unable to maintain any valid connection with other nodes -Disconnect any node from other nodes -Forgery and replay any transaction or block but cannot benefit a lot -Forge signatures and obtain the ability to use other people's private keys to sign arbitrary data -Get the private key of some accounts -Obtain a small amount of unexpected capital gains -Cause a loss of funds to certain accounts -Unauthorized modification of account address or permission settings

  Low Hazard

  Low-level hazards refer to problems where vulnerabilities cause a certain degree of confusion or economic loss to some nodes or accounts, and will not cause substantial damage to the blockchain system, nodes or accounts, but still need to be improved and have potential risks.

  including but not limited to:

-Replay specific transactions or blocks -Fail to start any node -Make any node unable to establish a valid connection with other nodes -Significantly reduce the difficulty of exploiting other attacks -Disable the server RPC interface

-Leakage of sensitive information that will not directly cause economic losses -Reduce the difficulty of using other attacks to a certain extent

Disclaimer:

Horizon Technology only issues a report and assumes corresponding responsibilities for the facts that occurred or existed before the issuance of this report. Since the facts that occurred after the issuance of the report cannot judge the safety status of the project, it is not responsible for this. The subsequent on-chain deployment and operation methods of the project party are outside the scope of this audit. This report only conducts a security audit based on the information provided by the information provider to ZEISS at the time the report is issued. If the information of this project is concealed or the situation reflected is inconsistent with the actual situation, ZEISS will be responsible for the losses caused thereby And no responsibility for adverse effects.

There are risks in the market and investment needs to be cautious. This report only conducts security audits and results announcements on the project code, and does not make investment recommendations and basis.

NONEAGE

Telephone：86-17391945345 18511993344

Email ：support@noneage.com

Site ：www.noneage.com

Weibo ：weibo.com/noneage