Safeheron MPC Node
Security Audit Report

Table Of Contents

1 Executive Summary

On 2024.01.15, the SlowMist security team received the Safeheron team's security audit application for Safeheron MPC Node, developed the audit plan according to the agreement of both parties and the characteristics of the project, and finally issued the security audit report.

The SlowMist security team adopts the strategy of "black/grey box lead, white box assists" to conduct a complete security test on the project in the way closest to the real attack.

The test method information:

Test method	Description
Black box testing	Conduct security tests from an attacker's perspective externally.
Grey box testing	Conduct security testing on code modules through the scripting tool, observing the internal running status, mining weaknesses.
White box testing	Based on the open source code, non-open source code, to detect whether there are vulnerabilities in programs such as nodes, SDK, etc.

The vulnerability severity level information:

Level	Description
Critical	Critical severity vulnerabilities will have a significant impact on the security of the project, and it is strongly recommended to fix the critical vulnerabilities.
High	High severity vulnerabilities will affect the normal operation of the project. It is strongly recommended to fix high-risk vulnerabilities.
Medium	Medium severity vulnerability will affect the operation of the project. It is recommended to fix medium-risk vulnerabilities.
Low	Low severity vulnerabilities may affect the operation of the project in certain scenarios. It is suggested that the project team should evaluate and consider whether these vulnerabilities need to be fixed.
Weakness	There are safety risks theoretically, but it is extremely difficult to reproduce in engineering.
Suggestion	There are better practices for coding or architecture.

2 Audit Methodology

The security audit process of SlowMist security team for wallet application includes two steps:

The codes are scanned/tested for commonly known and more specific vulnerabilities using automated analysis tools.

Manual audit of the codes for security issues. The wallet application is manually analyzed to look for any potential issues.

The following is a list of security audit items considered during an audit:

NO.	Audit Items	Result
1	App runtime environment detection	Passed
2	Code decompilation detection	Passed
3	App permissions detection	Passed
4	File storage security audit	Passed
5	Communication encryption security audit	Passed
6	Interface security audit	Fixed
7	Business security audit	Passed
8	WebKit security audit	Passed
9	App cache security audit	Passed
10	WebView DOM security audit	Passed
11	SQLite storage security audit	Passed
12	Deeplinks security audit	Passed
13	Client-Based Authentication Security audit	Passed
14	Signature security audit	Passed
15	Deposit/Transfer security audit	Passed
16	Transaction broadcast security audit	Passed

NO.	Audit Items	Result
17	Secret key generation security audit	Passed
18	Secret key storage security audit	Fixed
19	Secret key usage security audit	Passed
20	Secret key backup security audit	Passed
21	Secret key destruction security audit	Fixed
22	Screenshot/screen recording detection	Passed
23	Paste copy detection	Passed
24	Keyboard keystroke cache detection	Passed
25	Insecure entropy source audit	Passed
26	Background obfuscation detection	Passed
27	Suspend evoke security audit	Passed
28	AML anti-money laundering security policy detection	Passed
29	Others	Passed
30	User interaction security	Passed

3 Project Overview

3.1 Project Introduction

Audit Version

Website:

https://www.safeheron.com/

Module & Docker Image Version:

Safeheron MPC Relayer

Version: smn-relayer:1.0.0-20240105_145406

Safeheron MPC Node Service

Version: smn-service:1.0.0-20240112_150826

Safeheron Embedded MPC Node

libsmnSDK-JS: 1.0.0-20240111_145655

libsmnSDK-IOS: 1.0.0-20240124_214236

libsmnSDK-Android: 1.0.0-20240112_122815

Fixed Version

Safeheron MPC Node Service

Version: smn-relayer:1.0.0-20240126_151044

Safeheron MPC Node Service

Version: smn-service:1.0.0-20240129_133353

Safeheron Embedded MPC Node

libsmnSDK-JS: 1.0.0-20240129_120456

libsmnSDK-IOS: 1.0.0-20240128_180228

libsmnSDK-Android: 1.0.0-20240128_195256

3.2 Vulnerability Information

The following is the status of the vulnerabilities found in this audit:

NO	Title	Category	Level	Status
N1	The Interface Contains Unnecessary Paths	Interface security audit	Suggestion	Fixed
N2	The Interface Returns Unescaped Characters, Posing XSS Risk	Interface security audit	Low	Fixed

NO	Title	Category	Level	Status
N3	Secret key storage issue	Secret key storage security audit	Suggestion	Fixed
N4	Secret key destruction issue	Secret key destruction security audit	High	Fixed

3.3 Vulnerability Summary

[N1] [Suggestion] The Interface Contains Unnecessary Paths

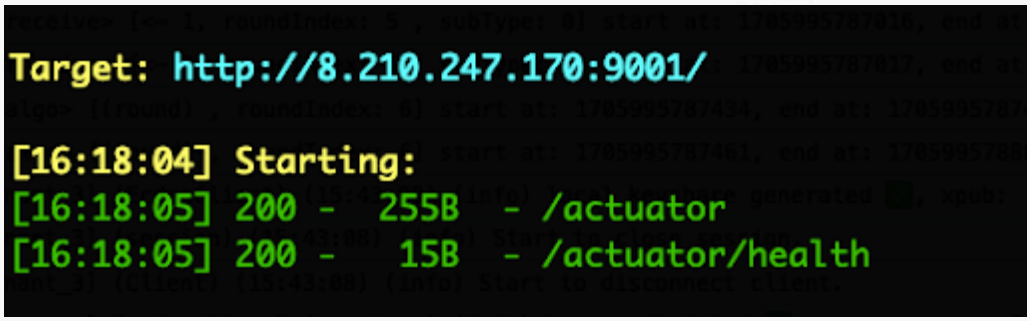
Category: Interface security audit

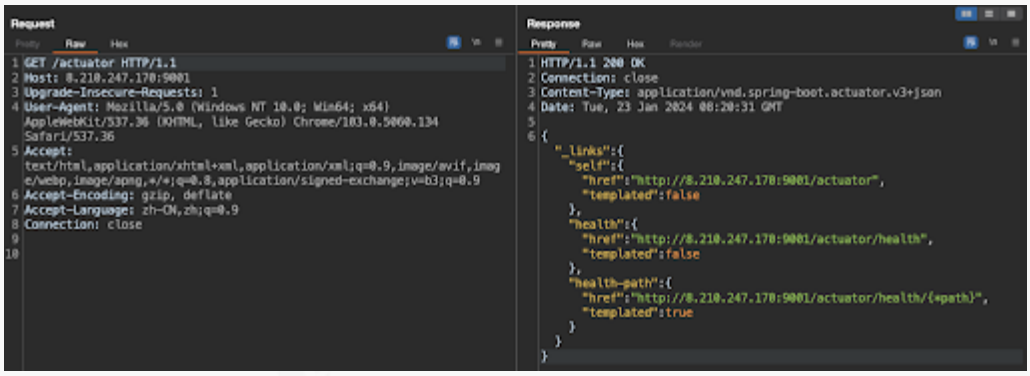
Content

The Relayer has unnecessary interfaces, such as:

http://8.210.247.170:9001/actuator

http://8.210.247.170:9001/actuator/health





Solution

Unnecessary interfaces that are not required for business purposes should not be exposed and should be blocked.

Status

Fixed

[N2] [Low] The Interface Returns Unescaped Characters, Posing XSS Risk

Category: Interface security audit

Content

MPC Node Service, due to developers lacking permission to make modifications, is responsible for handling all server-side character output escaping. For instance, the value of the sessionId parameter returned to the frontend in case of errors lacks proper character escaping, posing a risk of cross-site scripting.

URLs:

- http://8.210.247.170:8088/mpc/ecdsa/keygen
- http://8.210.247.170:8088/mpc/ecdsa/sign

Parameters:

- SessionId
- KeyId

The request package is as follows:

```
POST /mpc/ecdsa/keygen HTTP/1.1
Host: 8.210.247.170:8088

{"curveName":"Secp256k1","remoteParties":
[{"id":2,"tpk":"044e17df9d6b74f3f7d11bf0e27cad8f382f8f25e2158d9c7a987efbcd0f10873dc80
942995f81bb63fe7e227624e5f6578efc5f7e012e0a899828062fa973e77c"},
{"id":3,"tpk":"0499066c93b8cc30a196c1ace55fdeda8676ccc56491320d947ec98749187d8738d31d
28c5128c973586599e33bee76f6e619c04290b437098d189e86a8e236a34"}],"sessionId":"\u003cScR
iPt\u003eifijrbbyif\u003c/ScRiPt\u003e","threshold":2,"totalPartyNum":3}
```

The response package is as follows:

```
HTTP/1.1 200 OK

{
  "code": "500",
  "data": null,
  "errCode": "M201059",
  "errMsg": "The format of sessionId does not conform to the UUID.. sessionId=
<ScRiPt>ifijrbbyif</ScRiPt>",
  "success": false,
```

```
    "timestamp": "1706251948034"
}
```

Solution

It is recommended that the server performs escaping when returning special character outputs.

Status

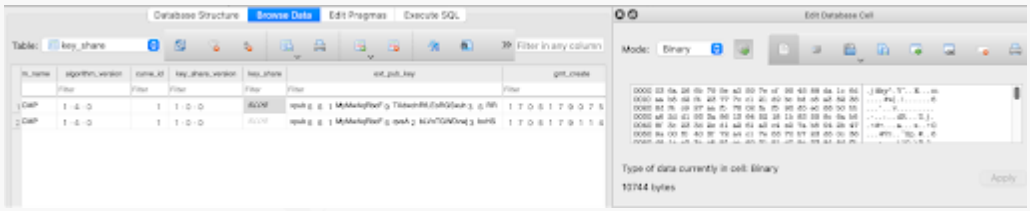
Fixed

[N3] [Suggestion] Secret key storage issue

Category: Secret key storage security audit

Content

The private key is stored in encrypted fragments in an SQLite file, and currently, in the SDK demo, the encryption key is a fixed value. The management of the encryption key is a risk point, as it is currently left to the discretion of the developer to define how to manage the encryption key.



Solution

It is recommended that in the delivery document of Safeheron MPC Node, it should be stated and emphasized that the management of encryption keys requires user participation through a password, and appropriate protection against brute-force attacks (such as PBKDF) should be implemented. Encryption should be stored in a secure device area (such as KeyChain).

Status

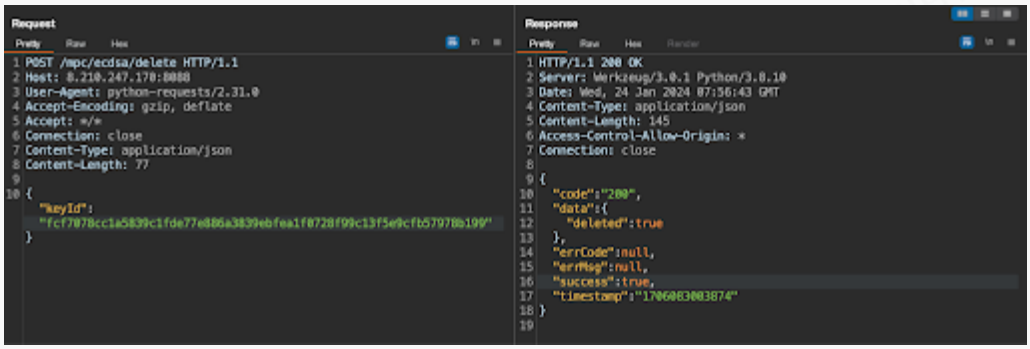
Fixed

[N4] [High] Secret key destruction issue

Category: Secret key destruction security audit

Content

The interface for deleting private key fragments is "mpc/ecdsa/keys/delete".



The server does not verify whether the deletion action comes from the owner of the keyId when deleting stored private key fragments from the database.

Solution

It is recommended that if this part requires developers to write additional business code to restrict permissions, the delivery document of SMN should clearly state and emphasize the potential risk of malicious deletion if permission restrictions are not properly implemented.

If this part requires the SMN SDK to perform permission checks for deletion, it is recommended that the interface for deleting fragments synchronously verifies the signature before executing the deletion operation.

Status

Fixed

4 Audit Result

Audit Number	Audit Team	Audit Date	Audit Result
0X002401310001	SlowMist Security Team	2024.01.15 - 2024.01.31	**Passed**

Summary conclusion: The SlowMist security team use a manual and SlowMist team's analysis tool to audit the project, during the audit work we found 1 high risk, 1 low risk, 2 suggestion vulnerabilities. And all vulnerabilities were Fixed. We extend our gratitude for Safeheron team recognition of SlowMist and hard work and support of relevant staff.

5 Statement

SlowMist issues this report with reference to the facts that have occurred or existed before the issuance of this report, and only assumes corresponding responsibility based on these.

For the facts that occurred or existed after the issuance, SlowMist is not able to judge the security status of this project, and is not responsible for them. The security audit analysis and other contents of this report are based on the documents and materials provided to SlowMist by the information provider till the date of the insurance report (referred to as "provided information"). SlowMist assumes: The information provided is not missing, tampered with, deleted or concealed. If the information provided is missing, tampered with, deleted, concealed, or inconsistent with the actual situation, the SlowMist shall not be liable for any loss or adverse effect resulting therefrom. SlowMist only conducts the agreed security audit on the security situation of the project and issues this report. SlowMist is not responsible for the background and other conditions of the project.

Official Website

www.slowmist.com

E-mail

team@slowmist.com

Twitter

@SlowMist_Team

Github

https://github.com/slowmist