



Wallet Application Security Audit Report



Table Of Contents

1 Executive Summary	_____
2 Audit Methodology	_____
3 Project Overview	_____
3.1 Project Introduction	_____
3.2 Vulnerability Information	_____
3.3 Vulnerability Summary	_____
4 Audit Result	_____
5 Statement	_____

1 Executive Summary

On 2023.03.07, the SlowMist security team received the Sender Wallet team's security audit application for Sender Wallet iOS, 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	Confirmed
3	App permissions detection	Passed
4	File storage security audit	Passed
5	Communication encryption security audit	Fixed
6	Interface security audit	Passed
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	Passed
19	Secret key usage security audit	Passed
20	Secret key backup security audit	Passed
21	Secret key destruction security audit	Passed
22	Screenshot/screen recording detection	Confirmed
23	Paste copy detection	Passed
24	Keyboard keystroke cache detection	Confirmed
25	Background obfuscation detection	Passed
26	Suspend evoke security audit	Passed
27	AML anti-money laundering security policy detection	Confirmed
28	Others	Fixed
29	User interaction security	Confirmed

3 Project Overview

3.1 Project Introduction

Audit Version

<https://github.com/SenderWallet/sender-wallet-mobile/tree/slowmist-multi-chain>

commit: 973e529554053c15ac660a996965dfe17d037233

iOS sender-mobile-multichain.ipa V2.0.1

(SHA256:61ec98f52d628b023abc25b0cb5a4dcc651e0e9d2758b3fc188faf151cf3912)

Fixed Version

<https://github.com/SenderWallet/sender-wallet-mobile/tree/slowmist-multi-chain>

commit: 6ad3714c31fbae81c695ab846c42c50e2f92b21f

iOS sender-mobile-multichain.ipa V2.0.1

(SHA256:36b7f291fcf01601524affad3b057184c1560b91fbb264c4729d12fb912d0c47)

3.2 Vulnerability Information

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

NO	Title	Category	Level	Status
N1	The information leakage of developers	Code decompilation detection	Suggestion	Confirmed
N2	Verify Phrase issue	Others	Low	Confirmed
N3	App Transport Security AllowsArbitraryLoads is allowed	Communication encryption security audit	Low	Fixed
N4	React Native code decompilation issue	Code decompilation detection	Suggestion	Confirmed
N5	Missing screenshot/screen recording detection	Screenshot/screen recording detection	Suggestion	Confirmed
N6	Lack of secure keyboard	Keyboard keystroke cache detection	Suggestion	Confirmed
N7	Lack of AML security policy	AML anti-money laundering security policy detection	Suggestion	Confirmed
N8	User interaction security suggestions	User interaction security	Suggestion	Confirmed
N9	Address book issue	Others	Low	Fixed
N10	Address book issue	Others	Low	Fixed

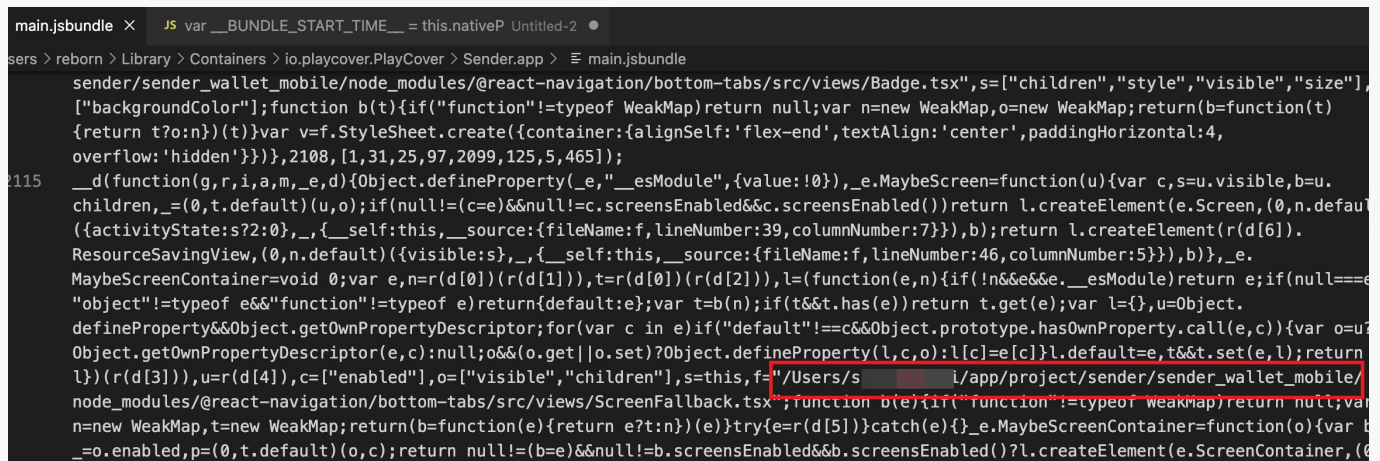
3.3 Vulnerability Summary

[N1] [Suggestion] The information leakage of developers

Category: Code decompilation detection

Content

The "main.jsbundle" file has leaked information about the developers.



Solution

It is recommended to check for information leaks in the code and remove them before publishing the sender wallet app.

Status

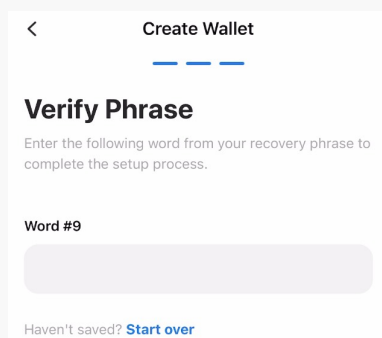
Confirmed

[N2] [Low] Verify Phrase issue

Category: Others

Content

Only one phrase was validated, not all the phrases.



Solution

To ensure complete user backup, it is recommended to validate all the phrases. It's better to use the complete

phrase and allow users to verify it by shuffling the word order.

Status

Confirmed

[N3] [Low] App Transport Security AllowsArbitraryLoads is allowed

Category: Communication encryption security audit

Content

App Transport Security restrictions are disabled for all network connections. Disabling ATS means that unsecured HTTP connections are allowed. HTTPS connections are also allowed, and are still subject to default server trust evaluation. However, extended security checks like requiring a minimum Transport Layer Security (TLS) protocol version—are disabled. This setting is not applicable to domains listed in NSExceptionDomains.

- ios/sender_wallet_mobile/Info.plist#Line31-43

```
<key>NSAppTransportSecurity</key>
<dict>
  <key>NSAllowsArbitraryLoads</key>
  <true/>
  <key>NSExceptionDomains</key>
  <dict>
    <key>localhost</key>
    <dict>
      <key>NSExceptionAllowsInsecureHTTPLoads</key>
      <true/>
    </dict>
  </dict>
</dict>
```

Solution

Avoid using the NSAllowsArbitraryLoads element and setting its value to as much as possible. Instead, use the default ATS settings and secure HTTPS connections whenever possible. Only use exceptions to allow insecure HTTP connections when it is truly necessary.

Avoid using the NSExceptionAllowsInsecureHTTPLoads element to enable insecure HTTP loads. Instead, use secure protocols and encryption algorithms supported by ATS as much as possible to protect the communication between the app and server.

Status

Fixed

[N4] [Suggestion] React Native code decompilation issue

Category: Code decompilation detection

Content

The Sender wallet is developed using the React Native framework, with the main logic located in the "sender_wallet_mobile.app/main.jsbundle" file. As it has not been fortified or code-verified, it can be easily debugged and tampered with.

Solution

It is recommended to add verification to the main.jsbundle file in the code to prevent tampering.

It is also suggested to use the Hermes optimization solution, which is a JavaScript optimization engine for React Native that is open-sourced by Facebook.

The original index.android.bundle file can be directly read as JavaScript text, but after being optimized by Hermes, it is converted into bytecode that cannot be directly read.

Reffer: [Hermes](#) is an open-source JavaScript engine optimized for running React Native apps on Android. For many apps, enabling Hermes will result in improved start-up time, decreased memory usage, and smaller app size. At this time Hermes is an opt-in React Native feature, and this guide explains how to enable it.

Status

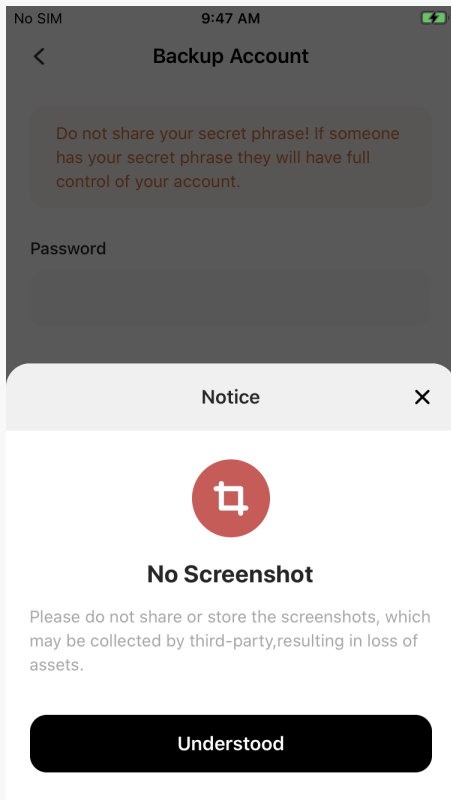
Confirmed

[N5] [Suggestion] Missing screenshot/screen recording detection

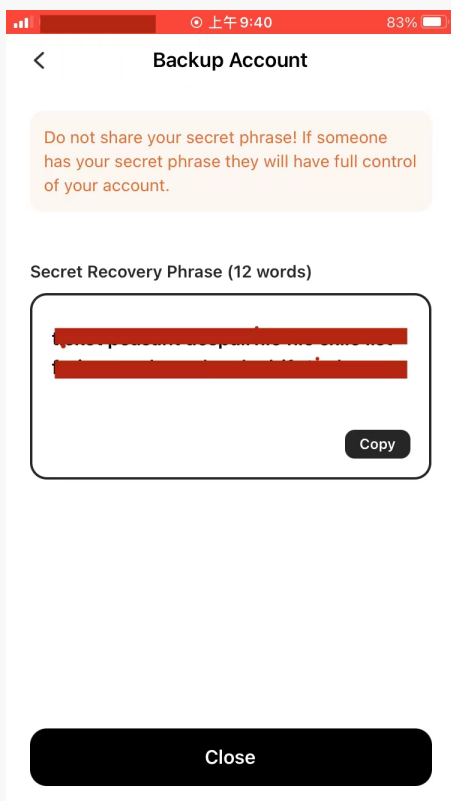
Category: Screenshot/screen recording detection

Content

When performing backup operations, users are reminded not to take screenshots or pictures.



However, the app does not have the ability to detect if the user takes screenshots or records the screen during backup operations.



Solution

We recommend implementing a screen capture and recording detection to prompt a safety reminder when users attempt to capture or record the app's content.

Status

Confirmed

[N6] [Suggestion] Lack of secure keyboard

Category: Keyboard keystroke cache detection

Content

Sender wallet uses the keyboard settings that come by default with iOS and does not have its own secure keyboard design.

Solution

It is recommended to use the built-in wallet keyboard to enter sensitive information such as passwords and mnemonics, and to avoid using third-party secure keyboards as they may store sensitive data in their cache, which can be easily stolen.

Status

Confirmed

[N7] [Suggestion] Lack of AML security policy

Category: AML anti-money laundering security policy detection

Content

Sender Wallet has not configured AML security policies and is unable to promptly synchronize malicious addresses with users.

Solution

It is recommended to configure AML security policies to remind users to avoid interacting with malicious addresses.

Status

Confirmed

[N8] [Suggestion] User interaction security suggestions

Category: User interaction security

Content

Functionality	Support	Notes
WYSIWYS	✓	There is friendly parsing of the data.
AML	✗	AML strategy is not supported.
Anti-phishing	✗	Phishing detect warning is not supported.
Pre-execution	✗	Pre-execution result display is not supported.
Contact whitelisting	•	The contact whitelisting is not supported, causing similar address attacks.
Password complexity requirements	✗	Missing password complexity requirements.

Tip: ✓ Full support, • Partial support, ✗ No support

Solution

It is recommended to increase the complexity of the password.

It is recommended to remind users to double-check the accuracy of the transfer destination address when it is not in their address book.

Status

Confirmed

[N9] [Low] Address book issue

Category: Others

Content

Sender wallet's "Remove" function for addresses in the address book is not working and addresses cannot be deleted.

- [src/redux/sagas/near.js#Line275-287](#)

```
function* removeAddressSaga(action) {
  try {
    const { index } = action;
    const nearStore = yield select(getNearStore);
    const { addressBook } = nearStore;

    const newAddressBook = _.filter(addressBook, (item) => `${item.index}` !==
```

```
`${index}`);  
  yield put(updateAddressBook(newAddressBook));  
  RootNavigation.goBack();  
} catch (error) {  
  console.log('remove address error: ', error);  
}  
}
```

Solution

It is recommended optimizing the address removal function.

Status

Fixed

[N10] [Low] Address book issue

Category: Others

Content

Sender wallet's address book does not perform format validation on the added addresses.



Solution

It is recommended to validate the format of the added addresses to ensure they are correct.

Status

Fixed

4 Audit Result

Audit Number	Audit Team	Audit Date	Audit Result
0X002303130003	SlowMist Security Team	2023.03.07 - 2023.03.13	Low Risk

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 4 low risks, and 6 suggestion vulnerabilities. And 3 low risks vulnerabilities were confirmed and being fixed. All the findings have been confirmed. We extend our gratitude for Sender Wallet 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](https://twitter.com/SlowMist_Team)



Github
<https://github.com/slowmist>