Wallet Application Security Audit Report

Table Of Contents

1 Executive Summary

On 2025.02.10, the SlowMist security team received the 77wallet team's security audit application for 77wallet (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-box and grey-box" 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	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	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

iOS

DownLink: https://apps.apple.com/hk/app/77wallet/id6738638349

Version: 1.0.5

Sha256 Sum: 60b3c636563813e543e3ac663f39061df3c9b609794fc6919fc9a015517bd753

Fixed Version

iOS

DownLink: https://apps.apple.com/hk/app/77wallet/id6738638349

Version: 1.2.0

Sha256: 14a6ead6b6564e00925fb3294d05a48d235bd9f0f1df6e1f32bda38918f8d76b

3.2 Vulnerability Information

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

NO	Title	Category	Level	Status
N1	App runtime environment issue	App runtime environment detection	Suggestion	Fixed
N2	Business security issue	Business security audit	Low	Fixed
N3	SQLite storage issue	SQLite storage security audit	Low	Fixed
N4	Secret key storage issue	Secret key storage security audit	Low	Fixed
N5	Secret key destruction issue	Secret key destruction security audit	Low	Fixed
N6	Screenshot/screen recording issue	Screenshot/screen recording detection	Suggestion	Fixed
N7	Paste copy issue	Paste copy detection	Suggestion	Fixed
N8	Keyboard keystroke cache issue	Keyboard keystroke cache detection	Suggestion	Acknowledged
N9	User interaction issue	User interaction security	Suggestion	Acknowledged

3.3 Vulnerability Summary

[N1] [Suggestion] App runtime environment issue

Category: App runtime environment detection

Content

When running device detection, a message will appear indicating that only iPhone devices are supported if used on a MacOS simulator, iPad, or similar environments.

After decompilation and testing on actual jailbroken devices, no jailbreak detection or alerts were found.

During actual testing with Frida Hook, no related Hook detection or alerts were discovered.

Solution

It is recommended to consider adding jailbreak detection with appropriate alerts.

It is also recommended to include detection for common hooking frameworks like Frida in your security checks and issue warnings when detected. Reference information on implementing Frida detection can be found at:

https://web.archive.org/web/20181227120751/http://www.vantagepoint.sg/blog/90-the-jiu-jitsu-of-detecting-frida

Status

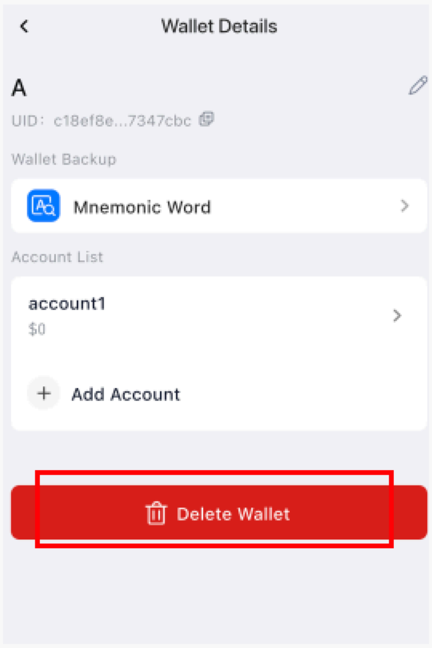
Fixed; The application has enhanced security measures, including detection for re-signing, simulators, and hooking tools, but it lacks detection and alerts for jailbroken devices.

[N2] [Low] Business security issue

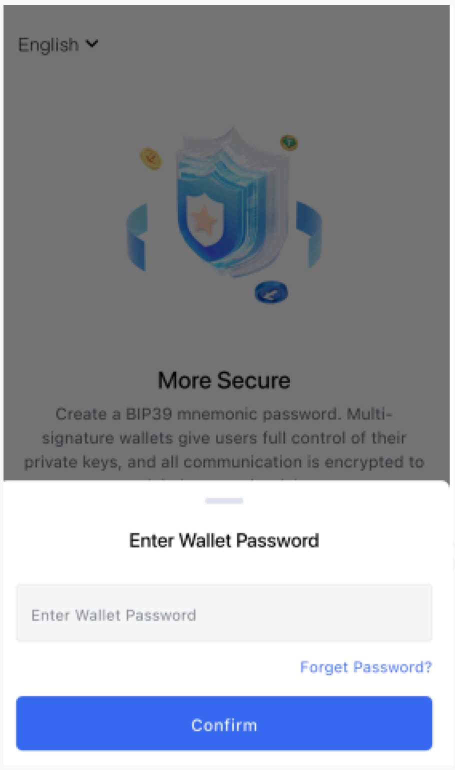
Category: Business security audit

Content

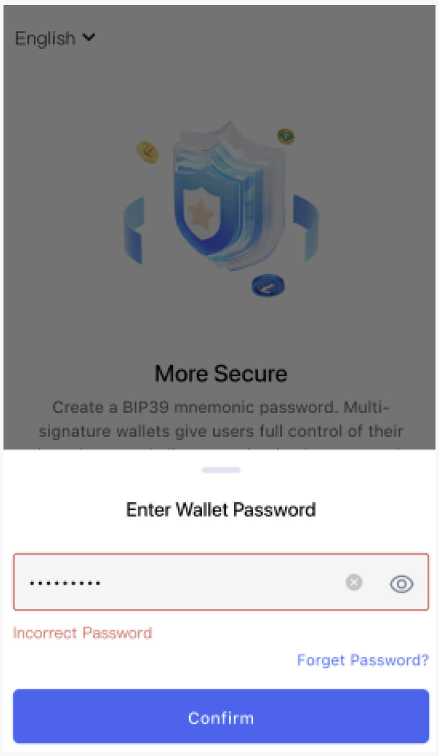
After deleting the last wallet and reopening the app, users are redirected to the wallet creation page. However, there's a misleading behavior here, as the system still requires verification with the original wallet password before importing a mnemonic phrase again. As shown, this occurs during the wallet deletion process.



At this point, importing a wallet again requires password verification.



Create a BIP39 mnemonic password. Multi-signature wallets give users full control of their private keys, and all communication is encrypted to



English

More Secure

Create a BIP39 mnemonic password. Multi-signature wallets give users full control of their

Enter Wallet Password

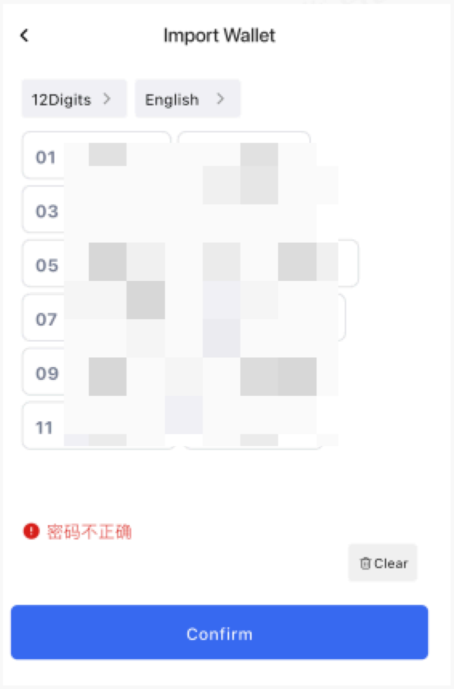
••••••••••

Incorrect Password

Forget Password?

Confirm

However, if you close the app and reopen it, the password verification popup mentioned above won't appear anymore, and you can import your mnemonic phrase normally. But after setting a wallet password and reaching the mnemonic phrase import confirmation screen, if you enter anything other than the original password, you'll get a password error message.



Import Wallet

12Digits English

01 03 05 07 09 11

密码不正确

Clear

Confirm

Solution

It is recommended to optimize the user flow after wallet deletion by clearing the password hash information from the database when the last wallet is removed. This would create a more intuitive experience and eliminate the confusion caused by the current logic.

Status

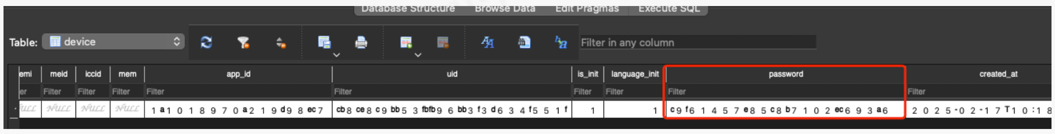
Fixed

[N3] [Low] SQLite storage issue

Category: SQLite storage security audit

Content

In the SQLite files within the app's sandbox, the Device-related tables are storing both the device serial number and the user's wallet password.



emi	meid	iccid	mem	app_id	uid	is_init	language_init	password	created_at
NULL	NULL	NULL	NULL	1a1018970a219d98ec7	cb8ce8c9bb53fbfb96bb3f3d634f551f	1	1	c9f61457e85c8b7102ec693a6	2025-02-17T10:18

Solution

It is recommended to avoid storing the user's wallet password hash on the device, as this creates a risk of rainbow table enumeration attacks.

Status

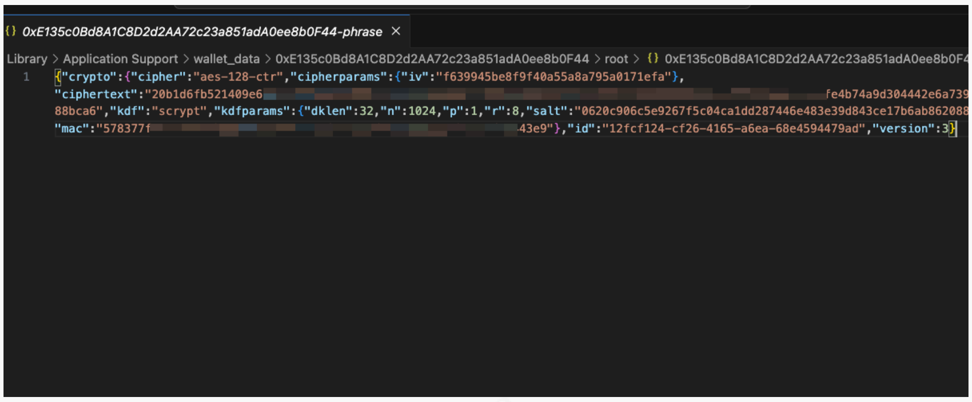
Fixed; The SQLite database file in the sandbox directory of the fixed version no longer stores password hashes.

[N4] [Low] Secret key storage issue

Category: Secret key storage security audit

Content

The current system has a security vulnerability when using the scrypt encryption algorithm, due to the work factor N being set too low (N = 1024, i.e. 2^10), which significantly reduces the computational cost of brute-force cracking. It is recommended to increase the N value to at least 2^13 (8MB), with an even better choice being to use 2^17 (128MB).



```
{} 0xE135c0Bd8A1C8D2d2AA72c23a851adA0ee8b0F44-phrase ×
Library > Application Support > wallet_data > 0xE135c0Bd8A1C8D2d2AA72c23a851adA0ee8b0F44 > root > {} 0xE135c0Bd8A1C8D2d2AA72c23a851adA0ee8b0F4
{"crypto":{"cipher":"aes-128-ctr","cipherparams":{"iv":"f639945be8f9f40a55a8a795a0171efa"},
"ciphertext":"20b1d6fb521409e6[redacted]fe4b74a9d304442e6a739
88bca6","kdf":"scrypt","kdfparams":{"dklen":32,"n":1024,"p":1,"r":8,"salt":"0620c906c5e9267f5c04ca1dd287446e483e39d843ce17b6ab862088
"mac":"578377f[redacted]43e9"},"id":"12fcf124-cf26-4165-a6ea-68e4594479ad","version":3}
```

Solution

It is recommended to increase the N value when using scrypt.

Refer：https://cheatsheetseries.owasp.org/cheatsheets/Password_Storage_Cheat_Sheet.html#scrypt

Status

Fixed; The fixed version implements a new password protection scheme using argon2id.

[N5] [Low] Secret key destruction issue

Category: Secret key destruction security audit

Content

Deleting the last wallet removes the KeyStore file from the sandbox, but doesn't delete the password hash from the database file.

Solution

It is recommended to delete the password hash from the wallet database when removing the last wallet.

Status

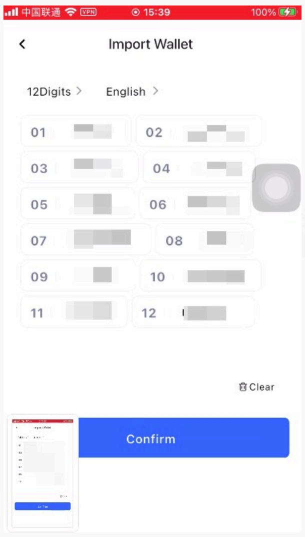
Fixed

[N6] [Suggestion] Screenshot/screen recording issue

Category: Screenshot/screen recording detection

Content

Pages containing sensitive information, such as the mnemonic phrase export page, have screenshot protection enabled.



However, some pages lack this screenshot prevention feature, like the wallet import page.

Solution

It is recommended to consistently implement anti-screenshot and anti-recording protection across all pages that display mnemonic phrases and private keys.

Status

Fixed

[N7] [Suggestion] Paste copy issue

Category: Paste copy detection

Content

When copying the private key, users are warned about the risks. However, after completing the paste action, the clipboard isn't cleared. For example, after pasting a mnemonic phrase during wallet import, the app doesn't automatically clear the clipboard.

Solution

It is recommended to clear the clipboard after copying and pasting mnemonic phrases.

Status

Fixed

[N8] [Suggestion] Keyboard keystroke cache issue

Category: Keyboard keystroke cache detection

Content

The app doesn't come with a secure keyboard.

Solution

It is recommended to use a secure keyboard, as third-party input methods may collect user input, potentially leading to mnemonic phrase leakage.

Status

Acknowledged

[N9] [Suggestion] User interaction issue

Category: User interaction security

Content

Functionality	Support	Notes
WYSIWYS	✗	Signature not supported.
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.
Password complexity requirements	✓	There is a password complexity limit.

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

Solution

It is recommended to improve the related user interactions.

Status

Acknowledged

4 Audit Result

Audit Number	Audit Team	Audit Date	Audit Result
0X002502190002	SlowMist Security Team	2025.02.10 - 2025.02.19	Passed

Summary conclusion: The SlowMist security team employs a manual approach along with the SlowMist team's analysis tool to conduct an audit of the project. During the audit process, four low-risk issues and five suggestions were identified. Additionally, four low-risk issues and three suggestions have been fixed. All other findings have been acknowledged.

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.