



# Wallet Application Security Audit Report



# Table Of Contents

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

# 1 Executive Summary

On 2025.02.10, the SlowMist security team received the 77wallet team's security audit application for 77wallet (Android), 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

#### Android

DownLink: [https://download.77.im/1.0.5/1/77wallet\\_1.0.5\\_466\\_0207\\_prod\\_company-sign.apk](https://download.77.im/1.0.5/1/77wallet_1.0.5_466_0207_prod_company-sign.apk)

Version: 1.0.5

Sha256 Sum: 5650f50952b59986f4d8a6ee05865175454ceda67cadecaffcfe00119dd6053c

## Fixed Version

### Android

DownLink: <https://play.google.com/store/apps/details?id=com.sevenwallet.app>

Version: 1.2.0

Sha256: 80d2b88430c794795d444e26776b433f763cf7d4f3d34c5c604f0c0aa33da97d

DownLink: [https://download.77.im/1.2.0/1/77wallet\\_1.2.0\\_807\\_0404\\_prod\\_company\\_signed.apk](https://download.77.im/1.2.0/1/77wallet_1.2.0_807_0404_prod_company_signed.apk)

Version: 1.2.0

Sha256: 1675932826cf3013639b6016811d9048615d846e3ee04f8c23093e7156d4df2c

## 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	Code decompilation issue	Code decompilation detection	Suggestion	Fixed
N3	App permissions issue	App permissions detection	Suggestion	Acknowledged
N4	Business security issue	Business security audit	Low	Fixed
N5	SQLite storage security issue	SQLite storage security audit	Low	Fixed
N6	Secret key storage issue	Secret key storage security audit	Low	Fixed
N7	Secret key destruction issue	Secret key destruction security audit	Low	Fixed
N8	Screenshot/screen recording issue	Screenshot/screen recording detection	Suggestion	Fixed
N9	Paste copy issue	Paste copy detection	Suggestion	Fixed

NO	Title	Category	Level	Status
N10	Keyboard keystroke cache issue	Keyboard keystroke cache detection	Suggestion	Acknowledged
N11	Background obfuscation issue	Background obfuscation detection	Suggestion	Fixed
N12	User interaction issue	User interaction security	Suggestion	Acknowledged

### 3.3 Vulnerability Summary

#### [N1] [Suggestion] App runtime environment issue

##### Category: App runtime environment detection

##### Content

##### 1. Device Compatibility

When running on MacOS simulators or iPad environments, you'll get a notification that the app only supports iPhone devices.

##### 2. Jailbreak Detection

After decompiling the app and testing on actual jailbroken devices, we didn't find any jailbreak detection mechanisms or warnings.

##### 3. Hook Detection

During our practical testing with Frida Hook, we couldn't find any Hook detection features or alerts in the app.

##### Solution

It is recommended to consider adding jailbreak detection and notifications to the app.

It is also recommended to include detection for common hooking frameworks like Frida in your security checks, and issue appropriate warnings when detected. For implementation guidance, you can refer to:

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

## Status

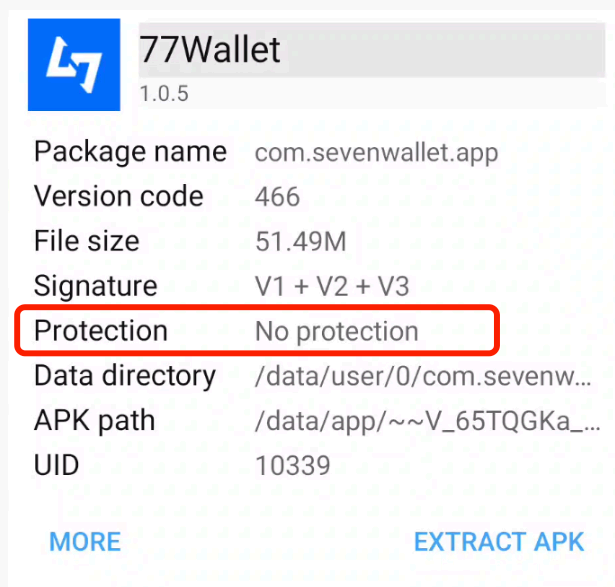
Fixed; The APK has been hardened with protection measures and implements detection for root access, emulators, and hooking techniques.

## [N2] [Suggestion] Code decompilation issue

**Category: Code decompilation detection**

### Content

The App lacks proper APK obfuscation and anti-reversing safeguards.



The screenshot shows the details of the 77Wallet app. The app icon is a blue square with a white 'L7' logo. The app name is '77Wallet' and the version is '1.0.5'. Below the app name, there is a table of app details. The 'Protection' row is highlighted with a red border, showing 'No protection'. At the bottom of the screenshot, there are two buttons: 'MORE' and 'EXTRACT APK'.

Package name	com.sevenwallet.app
Version code	466
File size	51.49M
Signature	V1 + V2 + V3
Protection	No protection
Data directory	/data/user/0/com.sevenw...
APK path	/data/app/~~V_65TQGKa_...
UID	10339

[MORE](#) [EXTRACT APK](#)

## Solution

It's recommended to include APK obfuscation and anti-reversing protection.

Reference link:

<https://jiagu.360.cn/#/global/index>

<https://dun.163.com/product/android-reinforce>

## Status

Fixed; The APK uses iJiami's protection solution, making it impossible to decompile directly.

## [N3] [Suggestion] App permissions issue

**Category: App permissions detection**

### Content

The App contains several high-risk permissions that aren't necessary for the business functionality.



```
uses-permission: name='android.permission.READ_MEDIA_IMAGES'  
uses-permission: name='android.permission.WRITE_EXTERNAL_STORAGE'  
uses-permission: name='android.permission.READ_EXTERNAL_STORAGE' maxSdkVersion='32'  
uses-permission: name='android.permission.MANAGE_EXTERNAL_STORAGE'  
uses-permission: name='android.permission.REQUEST_INSTALL_PACKAGES'
```

## Solution

It is recommended to follow the principle of least privilege, where any high-risk permissions that aren't essential for the business functions mentioned above should be removed.

## Status

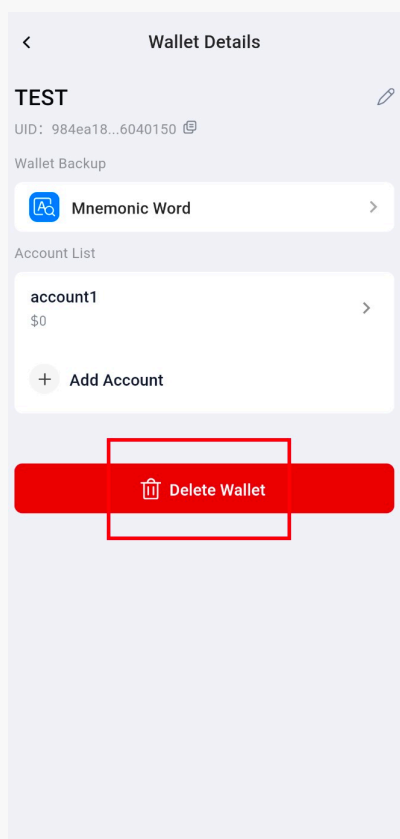
Acknowledged

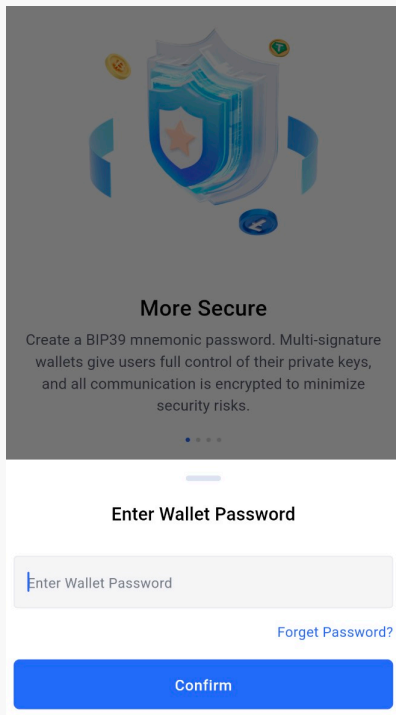
## [N4] [Low] Business security issue

**Category: Business security audit**

## Content

When a user deletes their last wallet and reopens the App, they're directed to the wallet creation page. However, this is somewhat misleading, as they still need to enter their original wallet password for verification before they can import a mnemonic phrase again.





If the user closes the App and reopens it at this point, the password verification popup won't appear anymore, and they can import the mnemonic phrase normally. But after setting a wallet password and reaching the mnemonic phrase import confirmation screen, they'll get a "wrong password" error message if they don't enter their original password.



## 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 deleted. This would create a more intuitive experience for users starting fresh after removing all their wallets.

## Status

Fixed

## [N5] [Low] SQLite storage security issue

**Category: SQLite storage security audit**

## Content

The SQLite database includes a table named `device`, which holds information related to the device, such as the device's serial number (SN) and password.

	is_init # INTEGER	language_init # INTEGER	password B <sub>C</sub> TEXT	created_at B <sub>C</sub> TIMESTAMP
ad5d868837d2	1	1	586653314581d6b43bee925d723843b9a431a0a307be9894b8b22a0e0791f205	2025-02-10T04:30

In this `device` table, the password stored is actually a hashed value generated by running the user's input password through the PBKDF2 algorithm with 100,000 iterations.

## 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.

## [N6] [Low] Secret key storage issue

**Category: Secret key storage security audit**

## Content

The current system has a security vulnerability when using the script 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).

```
1 {"crypto":{"cipher":"aes-128-ctr","cipherparams":{"iv":"ae29332b6ce969912bb5e9aad4696082"},
  "ciphertext":"c5386fd0474bc89a612ea186655d32c957bc7e245ae0802294c28c6c187a25a2
  884ccfa4ede58b5dae5648321f2e7801d410dfbf6028fa271230950740e716670d10e6dff7",
  "kdf":"scrypt","kdfparams":{"dklen":32,"n":1024,"p":1,"r":8,
  "salt":"b4d7e693a31a72092b75451726ba46e0414fc3f516aee46ec5f9310d67de13e1"},
  "mac":"0c5465f7ddd40a43fdc0330997a1c21e092df331f97e4225895dffe81a96e0ac"},
  "id":"4c963485-3689-4d96-bf23-490f405d5d29","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](https://cheatsheetseries.owasp.org/cheatsheets/Password_Storage_Cheat_Sheet.html#scrypt)

### Status

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

## [N7] [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

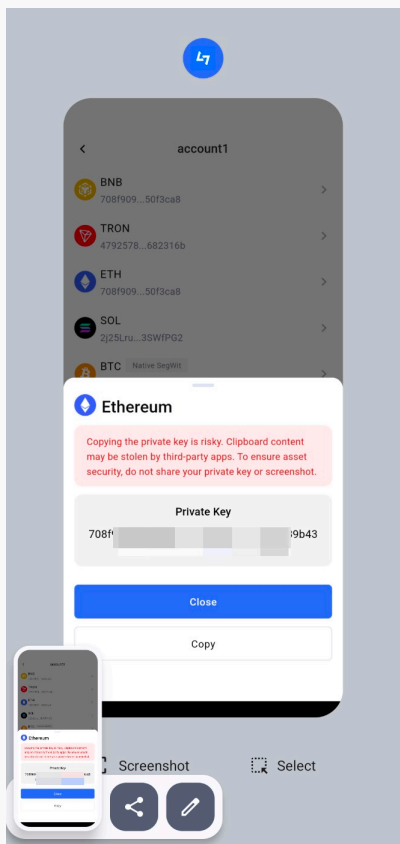
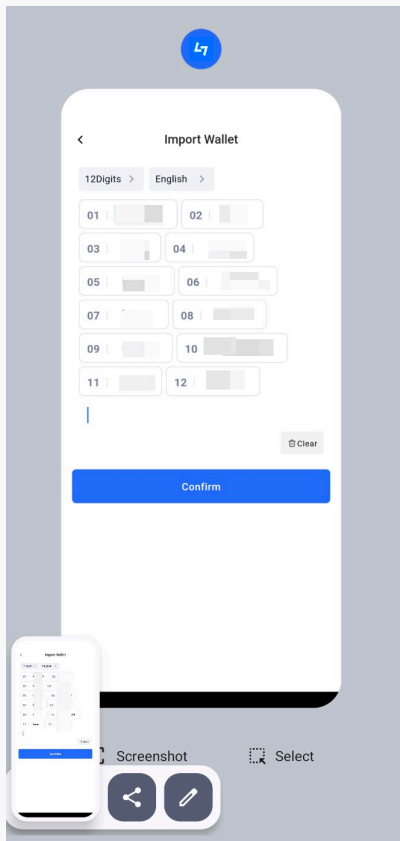
Fixed

## [N8] [Suggestion] Screenshot/screen recording issue

**Category:** Screenshot/screen recording detection

### Content

The mnemonic phrase backup export page has screenshot protection. However, both the mnemonic phrase import page and private key backup page lack protection against screenshots and screen recordings.



## 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

**[N9] [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**

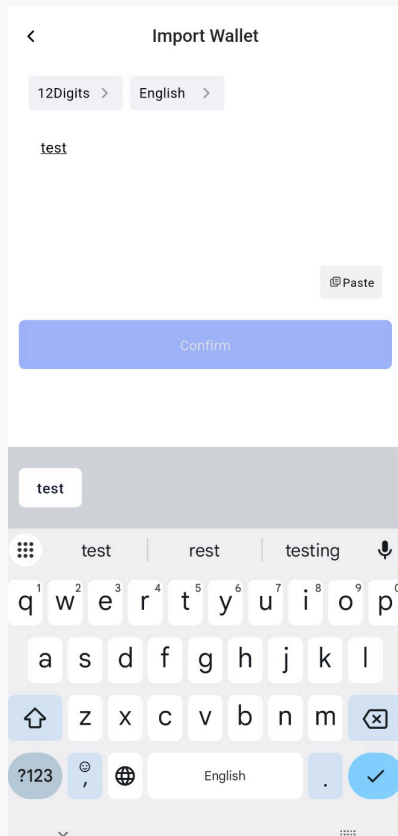
Fixed

**[N10] [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

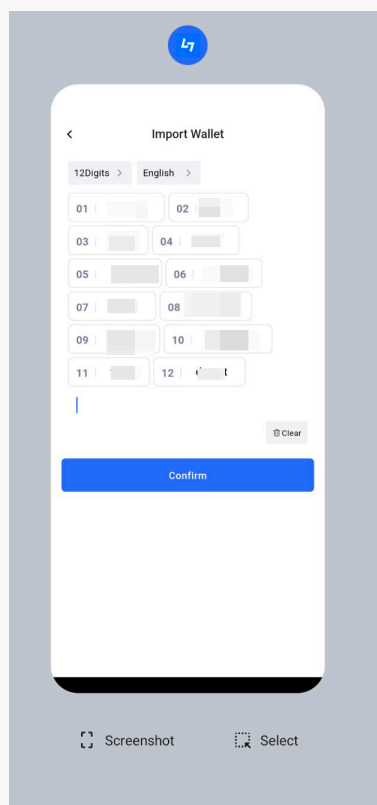
Acknowledged

## [N11] [Suggestion] Background obfuscation issue

### Category: Background obfuscation detection

### Content

When using the recent tasks interface for sliding up, the blur effect does not take effect immediately. Only after switching to the desktop completely and then switching back to the application, the display will turn completely black.



## Solution

It is recommended to blur wallet screens when the app is running in the background. This prevents sensitive data from being exposed when switching between apps while the wallet is displaying sensitive operation interfaces.

## Status

Fixed

## [N12] [Suggestion] User interaction issue

Category: User interaction security

## Content

Functionality	Support	Notes
<a href="#">WYSIWYS</a>	X	Signature not supported.
AML	X	AML strategy is not supported.
Anti-phishing	X	Phishing detect warning is not supported.
Pre-execution	X	Pre-execution result display is not supported.
Contact whitelisting	•	The contact whitelisting is not supported.



Functionality	Support	Notes
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
0X002502190001	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 eight suggestions were identified. Additionally, four low-risk issues and five 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.



**Official Website**  
[www.slowmist.com](http://www.slowmist.com)



**E-mail**  
[team@slowmist.com](mailto:team@slowmist.com)



**Twitter**  
[@SlowMist\\_Team](https://twitter.com/SlowMist_Team)



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