SLOWMIST

Wallet Application Security Audit Report

Table Of Contents

1 Executive Summary

On 2024.06.24, the SlowMist security team received the Rabby team's security audit application for Rabby mobile wallet 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/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	Confirmed
2	Code decompilation detection	Passed
3	App permissions detection	Confirmed
4	File storage security audit	Passed
5	Communication encryption security audit	Confirmed
6	Interface security audit	Confirmed
7	Business security audit	Passed
8	WebKit security audit	Passed
9	App cache security audit	Passed
10	WebView DOM security audit	Confirmed
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	Confirmed
19	Secret key usage security audit	Passed
20	Secret key backup security audit	Passed
21	Secret key destruction security audit	Confirmed
22	Screenshot/screen recording detection	Confirmed
23	Paste copy detection	Passed
24	Keyboard keystroke cache detection	Confirmed
25	Insecure entropy source audit	Passed
26	Background obfuscation detection	Confirmed
27	Suspend evoke security audit	Confirmed
28	AML anti-money laundering security policy detection	Confirmed
29	Others	Passed
30	User interaction security	Confirmed

3 Project Overview

3.1 Project Introduction

Audit Version

Source Code

Link: https://github.com/RabbyHub/rabby-mobile

Commit hash: a424dbe54bba464da7585769140f6b7136c9108b

Android

App Link: - https://download.rabby.io/downloads/wallet-mobile-reg/android-20240618_091214-0.2.0.19/rabbymobile.apk

App Version: 0.2.0

Sha256: 04535033c95bd6f5c208818273d7482474146a68130d06689ec7f1f0f3b0e03b

Fixed Version

Source Code

Link: https://github.com/RabbyHub/rabby-mobile

Commit hash: 9b0d8ace864c221e00c9ef53a415309827de9cf8

Android

App Link: - https://download.rabby.io/downloads/wallet-mobile-reg/android-20240712_040246-0.2.1.21/rabby-mobile.apk

App Version: 0.2.1

Sha256: e1bf6ea39f3ecb2fe43a5ce71c04b37bf2d2a01ec900152355676df5cd1ea866

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 detection issue	App runtime environment detection	Suggestion	Confirmed
N2	App permissions detection issue	App permissions detection	Suggestion	Fixed
N3	The interface document has leaked	Interface security audit	Suggestion	Fixed
N4	Allow WebView to Enable Camera and Location Silently	WebView DOM security audit	Low	Fixed

NO	Title	Category	Level	Status
N5	Inappropriate Domain Access Control in Rabby WebView	WebView DOM security audit	Low	Fixed
N6	Lack of Phishing Website Detection	WebView DOM security audit	Suggestion	Confirmed
N7	Checking WebView URL Address Issue	WebView DOM security audit	Suggestion	Fixed
N8	Mobile Wallet Apps Connect Issue	WebView DOM security audit	Suggestion	Confirmed
N9	Mnemonic Phrase/Private Key Destroy Issue	Secret key destruction security audit	Low	Fixed
N10	Screenshot/screen recording Issue	Screenshot/screen recording detection	Suggestion	Fixed
N11	Keyboard keystroke Cache Issue	Keyboard keystroke cache detection	Suggestion	Confirmed
N12	Background Obfuscation Issue	Background obfuscation detection	Suggestion	Fixed
N13	Suspend Evoke Issue	Suspend evoke security audit	Suggestion	Fixed
N14	Missing AML (Anti-Money Laundering) Policy	AML anti-money laundering security policy detection	Suggestion	Confirmed
N15	User Interaction Issue	User interaction security	Suggestion	Confirmed
N16	Communication encryption Issue	Communication encryption security audit	Suggestion	Confirmed
N17	Secret Key Storage Issue	Secret key storage security audit	Medium	Fixed

3.3 Vulnerability Summary

[N1] [Suggestion] App runtime environment detection issue

Category: App runtime environment detection

Content

Wallet App lacks security alerts for rooted detection.

Solution

It is recommended to add an Android device rooted detection and reminder scheme.

Status

Confirmed; The project team has confirmed that due to business considerations, this design will not be implemented at this time.

[N2] [Suggestion] App permissions detection issue

Category: App permissions detection

Content

The permissions “android.permission.WRITE_EXTERNAL_STORAGE” and “android.permission.READ_EXTERNAL_STORAGE” allow the app to access and modify files on the external storage.

Solution

It's important to evaluate whether the app actually requires this permission.

Status

Fixed

[N3] [Suggestion] The interface document has leaked

Category: Interface security audit

Content

The leaked interface list:

- https://alpha.rabby.io/swagger.json
- https://app-api.rabby.io/swagger.json

Solution

It is recommended to avoid exposing JSON interface documents unless absolutely necessary for complete public disclosure of interfaces.

Status

Fixed

[N4] [Low] Allow WebView to Enable Camera and Location Silently

Category: WebView DOM security audit

Content

The Rabby Wallet's WebView allows Dapps to access camera and GPS location permissions by default after opening.

Solution

It is recommended that when using Rabby to open a Dapp and utilizing camera and location permissions, the WebView component should prompt users to authorize relevant privacy permissions for the Dapp.

Status

Fixed

[N5] [Low] Inappropriate Domain Access Control in Rabby WebView

Category: WebView DOM security audit

Content

Improper domain access control in Rabby WebView results in continued signature requests from Dapps after switching from Explore to Home tabs, increasing the risk of phishing due to user misclicks.

Solution

It is recommended that after leaving the Dapp tab using the Rabby wallet, any requests from the Dapp should be blocked.

Status

Fixed

[N6] [Suggestion] Lack of Phishing Website Detection

Category: WebView DOM security audit

Content

Rabby's WebView component lacks phishing link detection.

Solution

It is suggested to maintain a list of malicious phishing website addresses and perform checks when accessing them.

You can refer to: Integrate: eth-phishing-detect.

Status

Confirmed; The project team has confirmed that due to business considerations, this design will not be implemented at this time.

[N7] [Suggestion] Checking WebView URL Address Issue

Category: WebView DOM security audit

Content

Accessing a Dapp with Rabby Wallet allows validation when the URL format appears as x.x.x.x. For example: "file:///etc/passwd/?x.x.x.x".

Currently, although the validation check is bypassed, requests are still enforced with HTTPS. However, the regex matching mentioned above can be strengthened and optimized.

Solution

It is recommended to strengthen URL address validation.

Status

Fixed

[N8] [Suggestion] Mobile Wallet Apps Connect Issue

Category: WebView DOM security audit

Content

Rabby can connect to other wallets via Mobile Wallet Apps. However, using this method defaults to connecting to the Dapp previously connected to Rabby. Typically, other wallets require reauthorization for connection.

Solution

It is recommended that after connecting to other wallets via Mobile Wallet Apps, connecting to the Dapp previously linked with Rabby should require reauthorization for connection confirmation.

Status

Confirmed; The project team has confirmed that due to business considerations, this design will not be implemented at this time.

[N9] [Low] Mnemonic Phrase/Private Key Destroy Issue

Category: Secret key destruction security audit

Content

After removing the wallet, password verification is required, but the contents of the mmkv.default file are not cleared.

Solution

It is recommended to delete the encrypted information stored in mmkv when removing the wallet's mnemonic and address.

Status

Fixed

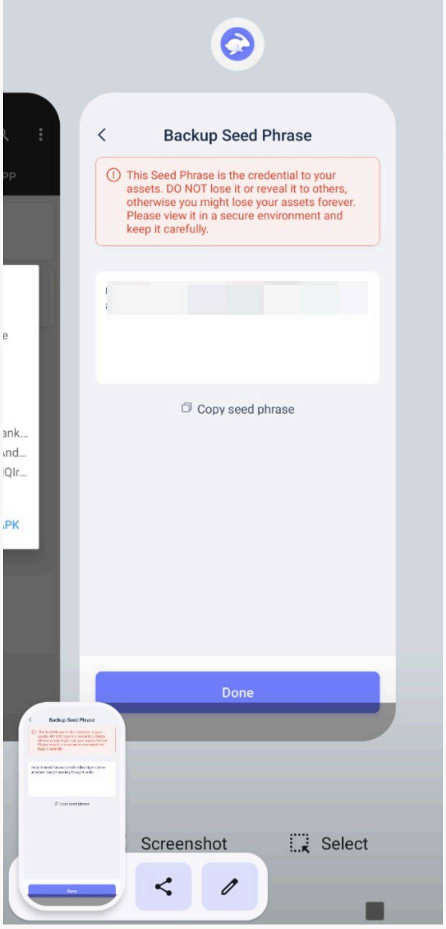
[N10] [Suggestion] Screenshot/screen recording Issue

Category: Screenshot/screen recording detection

Content

Backup of mnemonic lacks a security reminder against screenshots/recording.

The code layer for pages displaying mnemonics and private keys does not prohibit screenshots/recording.



Backup Seed Phrase

This Seed Phrase is the credential to your assets. DO NOT lose it or reveal it to others, otherwise you might lose your assets forever. Please view it in a secure environment and keep it carefully.

Copy seed phrase

Solution

It is recommended to disable screenshots/recording at the code layer of pages displaying mnemonic and private key information.

Status

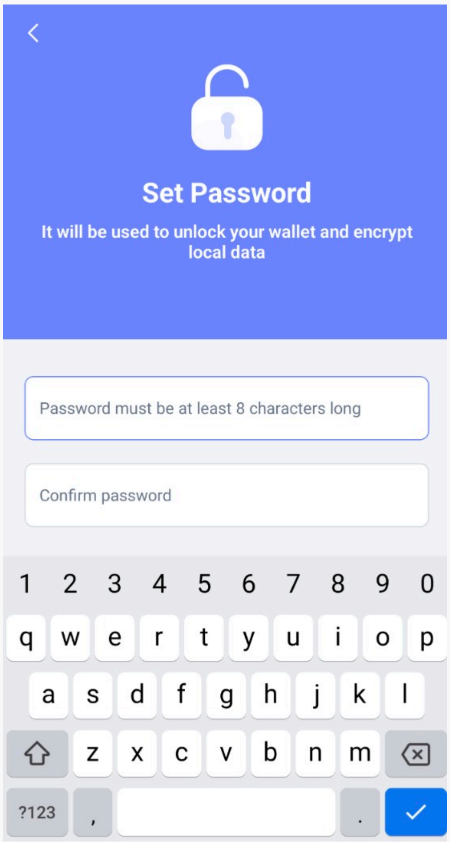
Fixed

[N11] [Suggestion] Keyboard keystroke Cache Issue

Category: Keyboard keystroke cache detection

Content

Rabby Wallet uses the system keyboard for password and other privacy inputs; the app does not have its own secure keyboard.



Set Password

It will be used to unlock your wallet and encrypt local data

Password must be at least 8 characters long

Confirm password

Solution

It is recommended that the app integrate a small keyboard to prevent input information such as mnemonic phrases and passwords from being cached by third-party keyboards.

Status

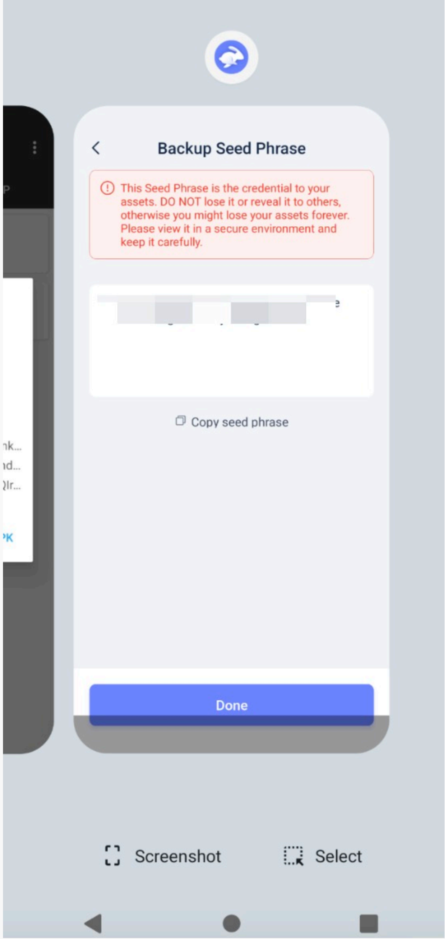
Confirmed; The project team has confirmed that due to business considerations, this design will not be implemented at this time.

[N12] [Suggestion] Background Obfuscation Issue

Category: Background obfuscation detection

Content

When the Rabby Wallet app runs and is sent to the background, the wallet pages are not blurred, which could potentially expose data to malicious apps when switching between apps.



Backup Seed Phrase

This Seed Phrase is the credential to your assets. DO NOT lose it or reveal it to others, otherwise you might lose your assets forever. Please view it in a secure environment and keep it carefully.

[redacted]

Solution

It is recommended to blur the wallet pages when the Rabby Wallet app is sent to the background.

Status

Fixed

[N13] [Suggestion] Suspend Evoke Issue

Category: Suspend evoke security audit

Content

No timeout mechanism was found in the wallet app, as testing showed that after being suspended for a considerable period, it did not prompt for password reauthentication.

Solution

It is recommended to add a timeout mechanism to the wallet app, so that it automatically logs out after a period of inactivity when running without user interaction.

Status

Fixed

[N14] [Suggestion] Missing AML (Anti-Money Laundering) Policy

Category: AML anti-money laundering security policy detection

Content

Rabby Wallet lacks support for Anti-Money Laundering (AML) services.

Solution

It is recommended to add Anti-Money Laundering (AML) services.

Status

Confirmed; The project team has confirmed that due to business considerations, this design will not be implemented at this time.

[N15] [Suggestion] User Interaction Issue

Category: User interaction security

Content

Functionality	Support	Notes
WYSIWYS	✓	There is no 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	✗	The application is designed with passwordless.

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

Solution

It is recommended to enhance security by integrating Anti-Money Laundering (AML) and anti-phishing capabilities. Additionally, enforcing password complexity requirements is highly advised.

Status

Confirmed

[N16] [Suggestion] Communication encryption Issue

Category: Communication encryption security audit

Content

1. Communication encryption is carried out using the HTTPS protocol for transmission.
2. Communication encryption performs certificate verification on the client-side and does not employ mutual authentication.

Solution

It is recommended to use two-way certificate binding or certificate whitelist for communication encryption.

Status

Confirmed; The project team has confirmed that due to business considerations, this design will not be implemented at this time.

[N17] [Medium] Secret Key Storage Issue

Category: Secret key storage security audit

Content

The feature to generate mnemonic phrases initially does not guide users to set a password but instead uses a default password for encryption storage, which exposes the app to all the elements needed to unlock secrets.

- apps/mobile/src/core/apis/lock.ts#130-156

```
export async function tryAutoUnlockRabbyMobile() {
  // // leave here for debugging
  if (__DEV__) {
    console.debug(
      'tryAutoUnlockRabbyMobile:: RABBY_MOBILE_KR_PWD',
      RABBY_MOBILE_KR_PWD,
    );
  }

  if (!keyringService.isBooted()) {
    await keyringService.boot(RABBY_MOBILE_KR_PWD);
  }
  const lockInfo = await getRabbyLockInfo();

  try {
    if (lockInfo.isUseBuiltInPwd && !keyringService.isUnlocked()) {
      await keyringService.submitPassword(RABBY_MOBILE_KR_PWD);
    }
  } catch (e) {
    console.error('[tryAutoUnlockRabbyMobile]');
    console.error(e);
  }

  return {
    lockInfo,
  };
}
```

MMKV continues to store data persistently without clearing previous entries. For example, when changing a password and storing newly encrypted mnemonic information, the old information remains uncleared.

- packages/service-keyring/src/keyringService.ts#131-152

```
private async _setupBoot(password: string) {
    this.password = password;
    const encryptBooted = await this.encryptor.encrypt(password, 'true');
    this.store.updateState({ booted: encryptBooted });
  }

  async boot(password: string) {
    await this._setupBoot(password);
    this.memStore.updateState({ isUnlocked: true });
  }

  async updatePassword(oldPassword: string, newPassword: string) {
    await this.verifyPassword(oldPassword);

    this.emit('beforeUpdatePassword', {
      keyringState: this.store.getState(),
    });

    // reboot it
    await this._setupBoot(newPassword);
    this.persistAllKeyrings();
  }
```

Solution

It is recommended to adjust the storage approach to avoid using incremental updates. When storing new encrypted data, old data should be deleted.

It is also recommended to guide users to set up a password before creating a wallet and generating mnemonic phrases.

Status

Fixed

4 Audit Result

Audit Number	Audit Team	Audit Date	Audit Result
0X002407030004	SlowMist Security Team	2024.06.24 - 2024.07.03	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 medium risk, 3 low risk, 13 suggestion vulnerabilities. And 7 suggestions were confirmed. All other findings were fixed. We extend our gratitude for Rabby mobile 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

Github

https://github.com/slowmist