# Browser Extension Wallet Security Audit Report

# Table Of Contents

# 1 Executive Summary

On 2025.08.04, the SlowMist security team received the Rabby team's security audit application for Rabby Browser Extension Wallet, 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 browser extension wallet 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 browser extension wallets are manually analyzed to look for any potential issues.

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

| NO. | Audit Class | Audit Subclass |
|---|---|---|
| 1 | Transfer security | Signature security audit |
| | | Deposit/Transfer security audit |
| | | Transaction broadcast security audit |
| 2 | Secret key security | Secret key generation security audit |
| | | Secret key storage security audit |
| | | Secret key usage security audit |
| | | Secret key backup security audit |
| | | Secret key destruction security audit |
| | | Insecure entropy source security audit |
| | | Cryptography security audit |
| 3 | Web front-end security | Cross-Site Scripting security audit |
| | | Third-party JS security audit |
| | | HTTP response header security audit |
| 4 | Communication security | Communication encryption security audit |
| | | Cross-domain transmission security audit |
| 5 | Architecture and business logic security | Access control security audit |

| NO. | Audit Class | Audit Subclass |
|---|---|---|
| | | Wallet lock security audit |
| | | Business design security audit |
| | | Architecture design security audit |
| | | Denial of Service security audit |

# 3 Project Overview

## 3.1 Project Introduction

**Audit Version**

Source Code

Link: https://github.com/RabbyHub/Rabby/releases/tag/v0.93.42

Commit hash: d15d129281df9720affb261d761d8ae481474c4c

**Fixed Version**

Source Code

Link: https://github.com/RabbyHub/Rabby

Commit hash: 06cac9804bb1b94585a32a11cd22d1a11e5f64c8

## 3.2 Vulnerability Information

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

| NO | Title | Category | Level | Status |
|---|---|---|---|---|
| N1 | Lack of Runtime Protections Mechanism | User interaction security | Suggestion | Acknowledged |
| N2 | Sync wallet function lacks password verification | Secret key usage security audit | Suggestion | Acknowledged |

| NO | Title | Category | Level | Status |
|---|---|---|---|---|
| N3 | "Unknown Signature Type" lacks security reminder | User interaction security | Suggestion | Fixed |

## 3.3 Vulnerability Summary

### [N1] [Suggestion] Lack of Runtime Protections Mechanism

**Category: User interaction security**

**Content**

Due to the lack of runtime protection mechanisms in the Rabby Extension Wallet, third-party dependencies or attackers can steal sensitive data by hooking native objects. For example, popups usually don't store data such as private keys and mnemonic phrases. However, when backing up or exporting the wallet, it's necessary to send the mnemonic phrases or private keys from Service Workers (background) to the popup for display, using the "controller" communication type. In this scenario, the mnemonic phrases or private keys can be captured by hooking the native types of Promise.


**Solution**

It is recommended to use LavaMoat to freeze objects, preventing the modification of global variables through hooks in popups or Service Workers (background) to read wallet data.

Reference: https://github.com/LavaMoat/LavaMoat

**Status**

Acknowledged

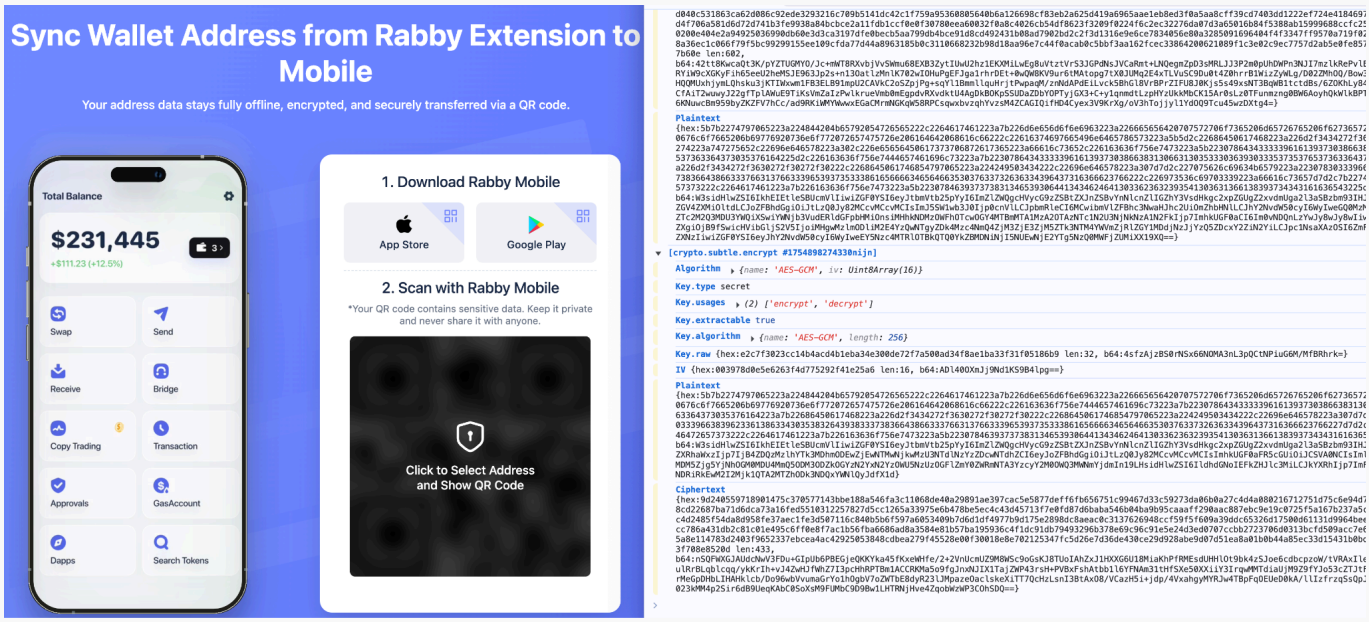
## [N2] [Suggestion] Sync wallet function lacks password verification

**Category: Secret key usage security audit**

**Content**

The sync wallet function is designed to synchronize wallet data to the rabby wallet's mobile app, with wallet data encrypted using the unlock password of the user's rabby browser extension wallet.

The plaintext data of the wallet is usually stored in memory. If the user needs to export the wallet, a password verification is required. However, the Sync wallet function lacks password verification. The encryption and decryption logic of the Sync wallet function can be intercepted by hooking `crypto.subtle.encrypt` and `crypto.subtle.decrypt`, thereby obtaining the wallet data.




**Solution**

It is recommended not only to use LavaMoat to protect the key global objects but also to add password verification in the Sync Wallet function to ensure the same level of security verification as the wallet export function.

**Status**

Acknowledged

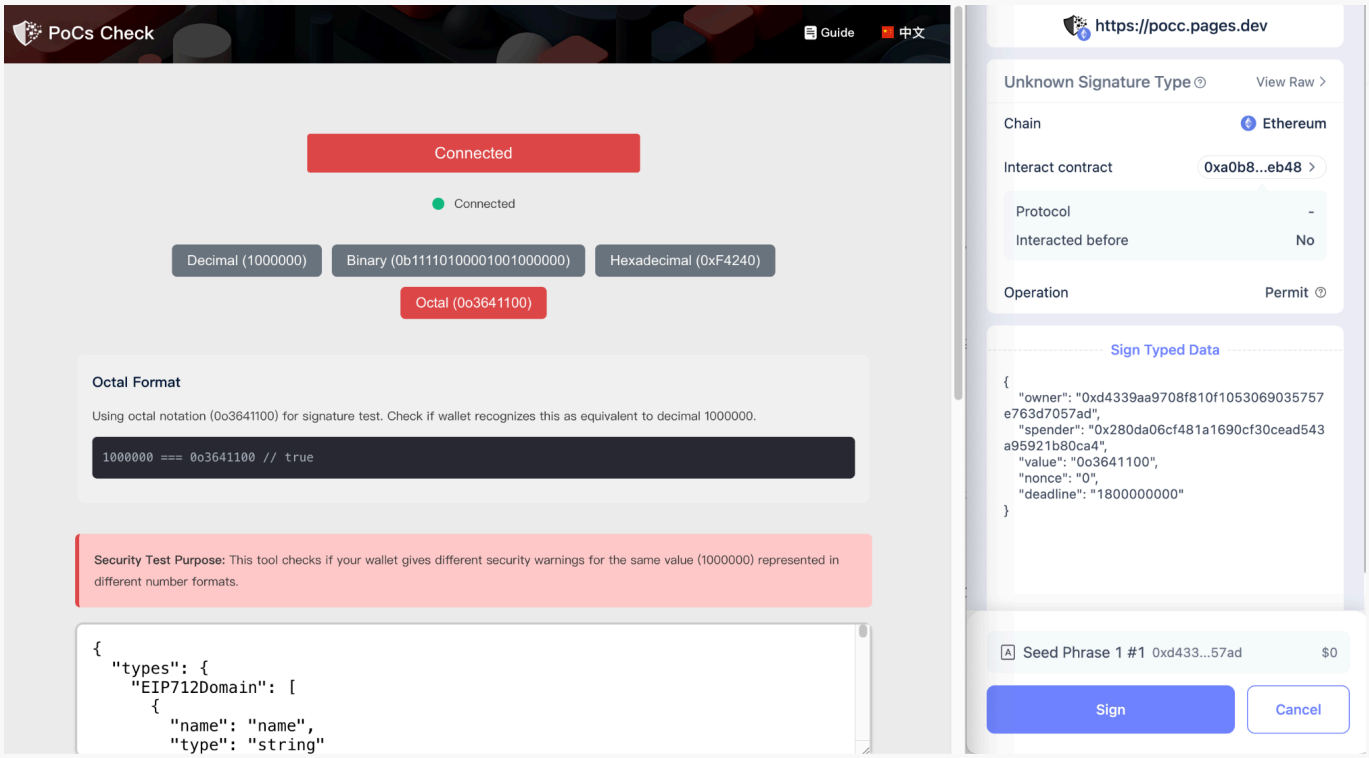
## [N3] [Suggestion] "Unknown Signature Type" lacks security reminder

**Category: User interaction security**

**Content**

Rabby browser extension wallet will mark transactions with unrecognized signature types as "Unknown Signature

Type" and there will be no reminder of security risks. However, these transactions can be normally parsed on the blockchain.




**Solution**

It is recommended to remind users of the security risks associated with these unrecognizable signature information, and to allow users to confirm the signature information carefully before proceeding with the signature.

**Status**

Fixed

# 4 Audit Result

| Audit Number | Audit Team | Audit Date | Audit Result |
|---|---|---|---|
| 0X002508150004 | SlowMist Security Team | 2025.08.04 - 2025.08.15 | Passed |

Summary conclusion: The SlowMist security team use a manual and SlowMist team's analysis tool to audit the project. During the audit, we identified four issues, all categorized as 'Suggestion' level. One of these issues has been fixed, while the remaining 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

**E-mail**

team@slowmist.com

**Twitter**

@SlowMist_Team

**Github**

https://github.com/slowmist