



Browser Extension Wallet 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 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

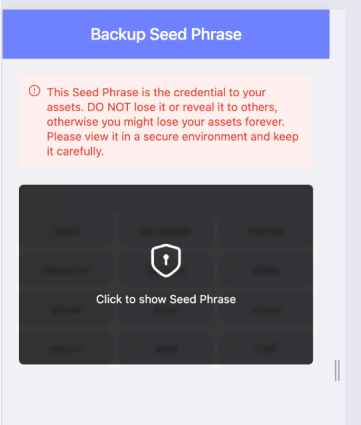
3.3 Vulnerability Summary

[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.



```
[Promise #883] FULFILL INPUT {"password":"12345678"}
[Promise #883] FULFILL OUTPUT undefined
[Promise #860] FULFILL INPUT {"password":"12345678"}
[Promise #884] CONSTRUCTOR
[Promise #885] CONSTRUCTOR
[Promise #886] FULFILL OUTPUT undefined
[Promise #886] FULFILL INPUT false
[Promise #886] FULFILL OUTPUT undefined
[Promise #887] CONSTRUCTOR
[Promise #888] CONSTRUCTOR
[Promise #889] FULFILL INPUT ""
[Promise #889] FULFILL OUTPUT undefined
[Promise #890] CONSTRUCTOR
[Promise #891] CONSTRUCTOR
[Promise #892] FULFILL INPUT "feed purpose merge observe faculty slide glove kiwi snow return add half"
[Promise #892] FULFILL OUTPUT undefined
[Promise #897] FULFILL INPUT {"brands":[{"brand":"Not A Brand","version":"8"}, {"brand":"Chromium","version":"138"}, {"brand":"Google Chrome","version":"138"}], "fullVersionList": [{"brand":"Not A Brand","version":"8.0.0.0"}, {"brand":"..."}]}
[Promise #897] FULFILL OUTPUT undefined
[Promise #899] CONSTRUCTOR
[Promise #893] FULFILL INPUT {}
[Promise #893] FULFILL OUTPUT undefined
[Promise #895] FULFILL INPUT {}
[Promise #895] FULFILL OUTPUT undefined
[Promise #900] FULFILL INPUT {}
[Promise #900] FULFILL OUTPUT {}
```

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

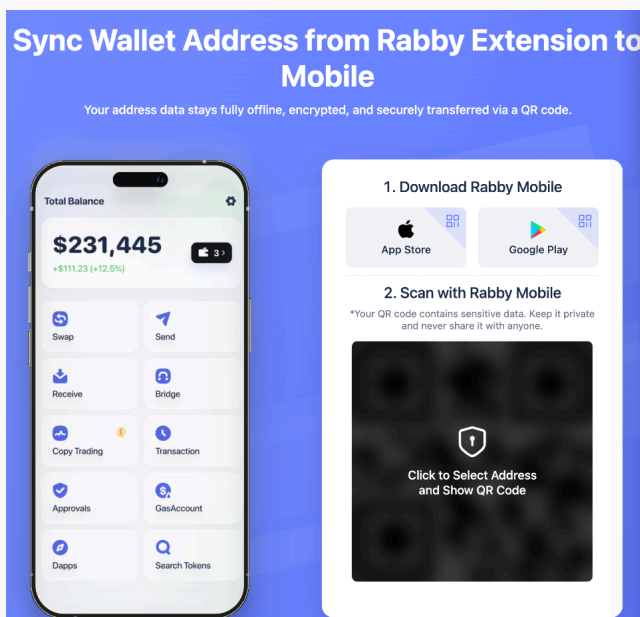
[N2] [Suggestion] Sync wallet function lacks password verification

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.



```
0840:531863c462088c92ede283216c789b5141d42c17759a95368085640b6a126698c783e42625d419d695a1eb8ed3f8a5a8c7f39cd7403dd1222e7724e418469;
d4780a9510d672d74131f49938a84bc4e2a11f0b1c7f0e0f180780e0d0037f8a8402e054f78623f3209f8224f6c2c32776d407f33050f10684f53990608c7c21
0200e404a2a9425036999d0b0e3d3ca3197dfe0ebc5aa799db4bce91d0c492431b08ad79b2bd2c2f3d1316e9dc7834056d8ba3285991096404f4f3347f19578a719f0;
8a38ec1c0667f9f50c929915ee199cfd477044a89631850b6c3110668232098d8a9967c44f0acab8c5b0f3aa102fccc33064208621809f1c3e829ec7757d2ab0e8f685;
70d0e len:602;
b64:421c06c0d3K/pZTU0Y0Y/Jc+mTBRXvbjYvShuW8XB32ytlUw42hz1EK0Mllw8Bvltztv/s313pDn0JVCaRnt+LHQeqZp039ML3Zv2wblp0WPhM3J7nzL0k6Pv1L
RY1MPCXG0Yf1nd8ee02HMS19633p2v130a11K702a20UpqE3p21rv0E1+4w0B0K9vUvM0cp71083J0M2E4TLvU5030a1420v+R3M412Ykq/0022M00/0w0/
H0QM0hJyML2hskJ3KT0w0v1F83B1891p0C2V5pPp+sqY110m1Lqurh1TfPwqM/zN0dAP0ELvck5BHG1BV/BP-ZTfU0KJk5549kxNT3BqM1t1c0L8s/6Z0Nk/y8
CfA1T2wuy22gTp1MAUE9T1KvZ01zPwLkrueVb0M6p0vKvdkU4Ag0B0Kp0550a2Z0YDTYj0K3C+y1gm0tLpH7Uk0M0K15A8bL20TfUrnng0M60A0hQ0M/L0P1
00N0w0Bm059y0Z0Y7YhC/c0B0M0M0w0e0G0F0M0Kp05B0Kp05q0v0qH7v0M2CAG1Q1H0B0Y0v0B0K0g0v03T0jYj1Y100P0C0v0500X0g4=
Plaintext
(hex:5b702274787065223a2484d404b65790254726565222c2264617461223a7b266e6e56d6f6e696223a2266656564207872727067f385206d65726765206f6273657;
0076c6f7665206b697f6920736e6f772072657473726e206164642068016c66222c2261637469705496465786573223a3b5d2c22686450617468223a26d21f434272f34
274223a74727652c22696e46578223a3802c226e656545061737370872617365223a6616c73652c2261636367f56e74727223a67223a087434333396163737383866313
5373634373835736164223a2c2261636367f56e7444657461696c73223a7b2230786434333396163737383866313066138353330639303353735765736336437
a22607f3424727363027f3807f38022c2268645061746845707865223a242405843422c226163646573223a307d707c2270756276498346570223a238703833396
7383643866337363173633396339373533861656663465663583763732633349643731636662376622c226973536c6978339223a6616c73657d742c7b2274
573722c2264617461223a7b2261636367f56e7472723a5b22387846393737383134653396441343464641383623623293541386313661383937343431636643225c
b64+4653dHw25161nE1EtL580cnW1Lw1z0FwS16ey1t0w1T25p1y16InC12M0g0Yv09Z50rZ2JhZ5BvYm1cnZ1L3ZhV3v0m0gczp2D0g2Zxv0m0p213a08zm031H
Z0V42M0101tdC1o2F0B0g010J1L0Qj02M0cV0KcV0mC1m0550w1b301p0cnV1L1Cp0m1Lc10M0cV0mV1Z7F0c3Nw0H3JcZU10mZ0M0L1L3H72W0v0d0cY10W0Iw0G0M0P
ZT2P0Q2M0U0M0230v1Y013v0J0E1u0F0p0W10n0J0M0H0P0F0H0T0c0d040F0M0T0A0Z00T0A0NT1C0E0J0N0M0L0N0F013p7T0m0H0B0c1G316m0D0L2L7w0Jy0Jy0J1w
Zg010/B95w1cW1b0J152V51J0J0p0M0Lm0D1L0E24Y02W0M7gyZ0K4Mzc4nn042H3Z03J03Z03M37M3T4M4VWwZJR1ZCY1M0J1H3J1Y0Q5Z0cY2Z2JN2Y1L3C3p1Ns0A0516Znf
ZM01Lw1Z0FwS16ey1t0w1T25p1y16InC12M0g0Yv09Z50rZ2JhZ5BvYm1cnZ1L3ZhV3v0m0gczp2D0g2Zxv0m0p213a08zm031H
Algorithm > {name: 'AES-GCM', iv: Uint8Array(16)}
Key.type secret
Key.usages > (2) ['encrypt', 'decrypt']
Key.extractable true
Key.algorithm > {name: 'AES-GCM', length: 256}
Key.raw (hex:e2c7f3823cc14b4acd41eba34e300de727fa508ad34f8ae1ba33f31f0518609 len:32, b64:4sf2A1zB50rNSv8M6X0M0A3p0QC1nP1uG0M/MTBRhKx=)
IV (hex:00397808e6263f4d7592f41e25a6 len:16, b64:AD1400XnJ19M1KS981Lpg=)
Plaintext
(hex:5b702274787065223a2484d404b65790254726565222c2264617461223a7b266e6e56d6f6e696223a2266656564207872727067f385206d65726765206f6273657;
0076c6f7665206b697f6920736e6f772072657473726e206164642068016c66222c2261636367f56e7444657461696c73223a7b2230786434333396163737383866313
03396633962336136634383538353834393633738366438643376333766333965393735338616566634656646538763732633396441343464641383623623293541386313661383937343431636643225c
b64+4653dHw25161nE1EtL580cnW1Lw1z0FwS16ey1t0w1T25p1y16InC12M0g0Yv09Z50rZ2JhZ5BvYm1cnZ1L3ZhV3v0m0gczp2D0g2Zxv0m0p213a08zm031H
Z0V42M0101tdC1o2F0B0g010J1L0Qj02M0cV0KcV0mC1m0550w1b301p0cnV1L1Cp0m1Lc10M0cV0mV1Z7F0c3Nw0H3JcZU10mZ0M0L1L3H72W0v0d0cY10W0Iw0G0M0P
ZT2P0Q2M0U0M0230v1Y013v0J0E1u0F0p0W10n0J0M0H0P0F0H0T0c0d040F0M0T0A0Z00T0A0NT1C0E0J0N0M0L0N0F013p7T0m0H0B0c1G316m0D0L2L7w0Jy0Jy0J1w
Zg010/B95w1cW1b0J152V51J0J0p0M0Lm0D1L0E24Y02W0M7gyZ0K4Mzc4nn042H3Z03J03Z03M37M3T4M4VWwZJR1ZCY1M0J1H3J1Y0Q5Z0cY2Z2JN2Y1L3C3p1Ns0A0516Znf
ZM01Lw1Z0FwS16ey1t0w1T25p1y16InC12M0g0Yv09Z50rZ2JhZ5BvYm1cnZ1L3ZhV3v0m0gczp2D0g2Zxv0m0p213a08zm031H
CipherText
(hex:9624059718981475c37877143bbe188a546fa3c11068de4ba20801ae397cac5e87def6f656751c946d7d3c59273da06b0a72c44a088016712751473c6e94d;
8cd2807ba71d06ca73a16fe0551831225787d5cc1265a339756b478be5ec4c43645137f08f07f0b0ba34084b09b95caaff729baa88f0eb9e19c8725f5a1070723f5c
c62405f54a0a0f0f0737aec1fe0597116c40b03d6f591a663400b0d6d1d0f709d1c200801d0a0c813762040c4cf59f5f08a3060c5326a175080b1131d0640e0
cc786a431d02c810e495c6ff0e087f1a156fba6686a8a3584e8157ba195936c4f1dc91d0794932963786e9c96c91e5e24d3e0d07cc0b273786d0313cfd590acc7ef
5a0e14735a34031965327f3ceca48c4295853848c4b0ea279f45528e0f30018e0c70212347f1c5d26e7d36e4c308c299208ba0e0f051eaa0ab0b44a5e3301543110b0c
37f808582d len:433;
b64:rs0F0K0G0A0U0cW03F0dU0ip06P8E0e00KXka45Kxk0MfE/z+2VnUcU2M9M5G0uG3K70U1A0Z11HXG0U18M1a0P1R0E5d0H0190k445J0e0c0d0c0z0v1V0A11e
u10R1q1L0q1y0K1Thv1w14A0d1T0b1713p0m0T0M1AC0R0050c0v0m0J1E1T01Z0P45r0v0P0F0A0A0b015V0M0311H5X0e00X0L1V31r0m0F01u0M0251y055Z2T1H
r0f0G0P0H1LH0K1cb0d056W0v0na0r0y010hg0v070ZM7EB0dYR231Mpa2e0Ac1sk01T770cH2Ln13B1A08/YCazH51+jdp/4Vxahy0MYR3w4T8Pqf0U0e0B0K1L1Hfzq550p;
023k0M4p251c6d90UeK0c50k0u09F0M0c9090v1LH7R1Hve4Zg0w0zW0P0C00S00=)
```

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

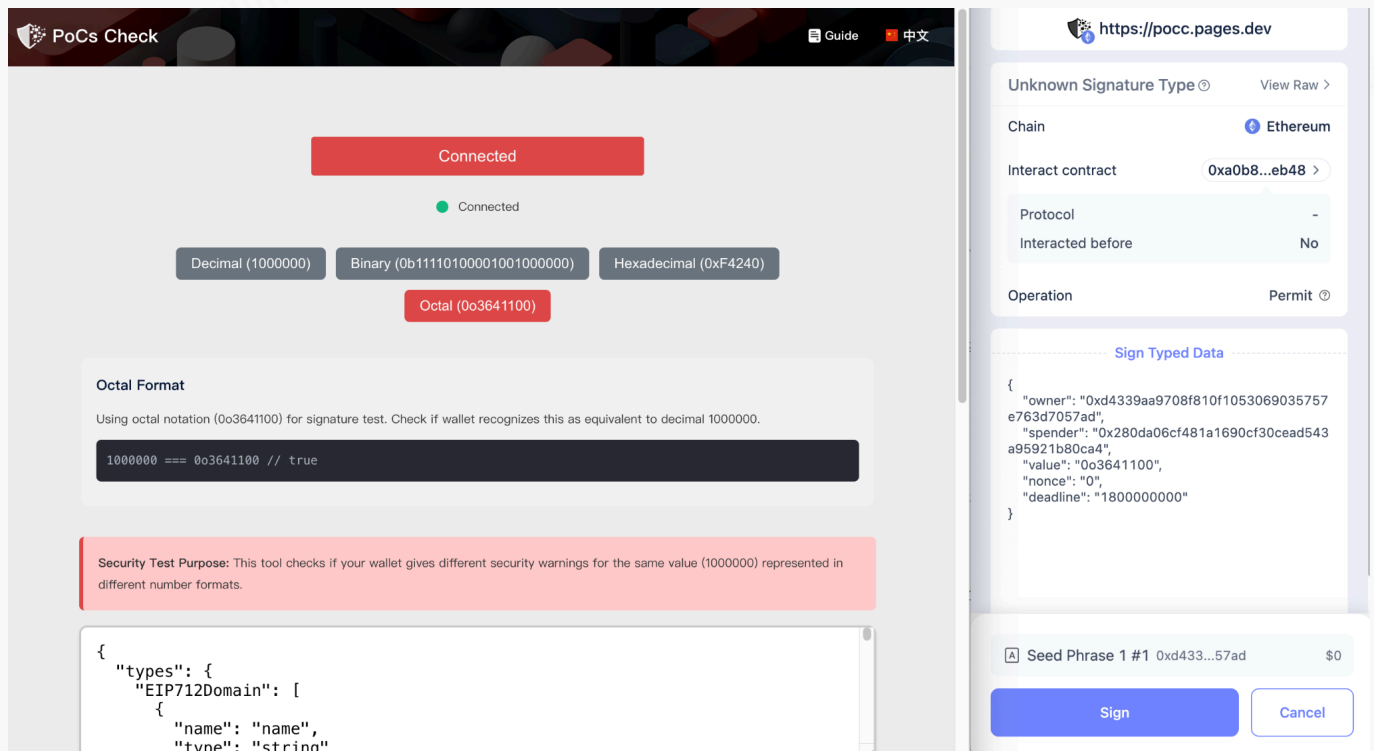
[N3] [Suggestion] "Unknown Signature Type" lacks security reminder

Category: User interaction security

Content

Rabby browser 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](https://twitter.com/SlowMist_Team)



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