verichains

SECURITY AUDIT OF

RONIN WALLET EXTENSION

Ronin

Public Report

Mar 18, 2022

Verichains Lab

info@verichains.io

https://www.verichains.io

Driving Technology > Forward

ABBREVIATIONS

Name	Description
Ethereum	An open source platform based on blockchain technology to create and distribute smart contracts and decentralized applications.
Ether (ETH)	A cryptocurrency whose blockchain is generated by the Ethereum platform. Ether is used for payment of transactions and computing services in the Ethereum network.

EXECUTIVE SUMMARY

This Security Audit Report prepared by Verichains Lab on September 28, 2021. We would like to thank Sky Mavis for trusting Verichains Lab in auditing the wallet extension. Delivering high-quality audits is always our top priority.

This audit focused on identifying security flaws in code and the design of the Ronin Wallet Extension. The scope of the audit is limited to the source code files provided to Verichains. Verichains Lab completed the assessment using manual, static, and dynamic analysis techniques.

During the audit process, the audit team had identified no vulnerable issues in the extension code, along with some recommendations.

TABLE OF CONTENTS

1. MANAGEMENT SUMMARY

1.1. About Sky Mavis

Sky Mavis is a blockchain gaming studio that built the successful NFT-based online game Axie Infinity. The team has a long history in gaming, they lauched Axie Infinity in 2018 with the idea of a blockchain-based play-to-earn model could create more aligned incentives between game creators and game players in long-term.

The company built Ronin Sidechain to solve network congestion problem for Axie, which provides:

- Fast & seamless transactions with almost instant confirmation.
- Drastically reduced gas fees. In addition, rather than paying Ethereum miners - the gas fees could be retained by the community and used for things like tournaments & bounties.
- The ability to withdraw Axie assets back to Ethereum Mainnet (eventually).
- Simplified on-boarding for new users, through a customized wallet solution.
- A block explorer for transparency and data accessibility.

1.2. About Ronin Wallet Extension

Ronin Wallet is the official wallet for Sky Mavis's Ronin sidechain.

This extension allows users to play Axie Infinity and other decentralized applications running on Ronin, an Ethereum sidechain built specifically for Blockchain games. Users can use **Ronin Wallet** to:

- Manage digital identity, experience 100% true ownership of their assets.
- Send transactions without paying expensive gas fees.

1.3. Audit scope

In this particular project, a timebox approach was used to define the consulting effort. This means that **Verichains Lab** allotted a prearranged amount of time to identify and document vulnerabilities. Because of this, there is no guarantee that the project has discovered all possible vulnerabilities and risks.

Furthermore, the security check is only an immediate evaluation of the situation at the time the check was performed. An evaluation of future security levels or possible future risks or vulnerabilities may not be derived from it.

1.4. Audit methodology

Verichains Lab's audit team mainly used the list below to check for Wallet Extension security:

- Transaction signature
- Transfer assets
- Transaction broadcast
- DApp communication
- Private Key/Mnemonic Phrase generation/destruction
- Private Key/Mnemonic Phrase secure storage
- Private Key/Mnemonic Phrase backup/restore
- Cryptography
- XSS
- Third-party JS
- HTTP Response Header
- Communication encryption
- Cross-domain transmission
- Access control
- Business design
- Architecture design

During the audit process, we also used tools for viewing, finding and verifying security issues of the app, such as following:

#	Name	Version
1	Chrome	92.0.4515.159
2	Visual Studio Code	1.60.0

Table 1. Tools used for audit

For vulnerabilities, we categorize the findings into categories as listed in table below, depending on their severity level:

SEVERITY LEVEL	DESCRIPTION
CRITICAL	A vulnerability that can disrupt the application functioning; creates a critical risk to the application; required to be fixed immediately.

SEVERITY LEVEL	DESCRIPTION
HIGH	A vulnerability that could affect the desired outcome of executing the application with high impact; needs to be fixed with high priority.
MEDIUM	A vulnerability that could affect the desired outcome of executing the application with medium impact in a specific scenario; needs to be fixed.
LOW	An issue that does not have a significant impact, can be considered as less important.

Table 2. Severity levels

1.5. Disclaimer

Please note that security auditing cannot uncover all existing vulnerabilities, and even an audit in which no vulnerabilities are found is not a guarantee for a 100% secure application. However, auditing allows discovering vulnerabilities that were unobserved, overlooked during development and areas where additional security measures are necessary.

2. AUDIT RESULT

2.1. Overview

The initial review was conducted in September 2021 and a total effort of 2 weeks was dedicated to identifying and documenting security issues in the code base of the Ronin Wallet Extension.

The following files were made available in the course of the review:

FILE	SHA1 SUM
extension-wallet-main.zip	2898AAA4FAD899A0FC4C1C59DCA1758C296E509D

2.2. Findings

During the audit process, the audit team did not discover any security vulnerability issues in the Ronin Wallet Extension. Only some minor issues in coding were found as in the following table:

#	Issue	Severity
1	Unvalidated parameter	INFORMATIVE
2	Double calling of error handler	INFORMATIVE
3	Wrong early-return condition	INFORMATIVE

Audit team also suggested some possible enhancements.

Sky Mavis fixed the code, according to Verichains's private report.

2.3. Issues

2.3.1. Unvalidated parameter INFORMATIVE

Consider the following snippet from *extension-wallet-main/api/index.ts*, networkId obtains its value from user request's parameter, the handler then checks if networks contains that property and returns the value.

```
app.use(`/networks/:networkId`, (req, res) => {
  const networkId = req.params['networkId'];
  const maybeNetwork = networks[networkId];

  if (maybeNetwork) {
    res.send(networks[networkId]);
  } else {
    res.status(404).send('Not found');
  }
});

const { PORT = 4003 } = process.env;
const httpServer = createServer(app);

httpServer.listen({ port: PORT }, () => {
  // eslint-disable-next-line no-console
  console.log(`Server ready at port ${PORT}`);
});
```

Snippet 1. extension-wallet-main/api/index.ts

If the variable network has some property which is a function, the above handler will cause source code leakage by sending the property name as networkId. For example, let's say networks is defined as:

```
network = {
    toString: function() {
        return `[System network container, possible values: ${Object.keys…
  (this).join(",")}]`
    }
}
```

Snippet 2. Example setup with exposable function

Requesting */networks/toString* will give us the source:

```
function() {\n        return `[System network container, possible values:
${Object.keys(this).join(",")}]`\n    }
```

The current audited version of **Ronin Wallet** does not assign any function to network, but it is possible that it may be changed in future, or an attacker can somehow trigger prototype pollution into global variables and read the result using this.

RECOMMENDATION

Validate the parameter:

```
app.use(`/networks/:networkId`, (req, res) => {
  const networkId = req.params['networkId'];
  const maybeNetwork = networks.hasOwnProperty(networkId) && typeof n…
```

```
etworks[networkId] === 'object';

  if (maybeNetwork) {
    res.send(networks[networkId]);
```

Snippet 3. Possible fix for /networks/:networkId handler

UPDATES

- *Mar 18, 2022*: This issue has been acknowledged and fixed by the Sky Mavis team.

2.3.2. Double calling of error handler INFORMATIVE

The implementaton of ConfirmSubprovider put the next and end callback into try block and then again use it in catch block, as highlighted on line *37*, *40*, *57* and *60* in the following snippet:

```
22 async handleRequest(
23     payload: JSONRPCRequestPayload,
24     next: Callback,
25     end: ErrorCallback,
26   ): Promise<void> {
27     switch (payload.method) {
28       case 'eth_sendTransaction': {
29         try {
30           const txParams = payload.params[0];
31           const userConfirmation = await this.askForTransactionConfir…
m(txParams);
32
33           if (userConfirmation) {
34             next();
35           } else {
36             const err = new Error('User cancel transaction');
37             end(err, null);
38           }
39         } catch (err) {
40           end(err, null);
41         } finally {
42           return;
43         }
44       }
45       case 'eth_sign':
46       case 'personal_sign': {
47         const data = payload.method === 'eth_sign' ? payload.params[1…
] : payload.params[0];
```

```
    const address = payload.method === 'eth_sign' ? payload.params[0] : payload.params[1];

    try {
     const userConfirmation = await this.askForPersonalMessageConfirm(data, address);

     if (userConfirmation) {
      next();
     } else {
      const err = new Error('User cancel sign message');
      end(err, null);
     }
    } catch (err) {
     end(err, null);
    } finally {
     return;
    }
   }
```

Snippet 4. background/subprovider/ConfirmSubprovider.ts

This can cause the end handler can be called after next or end already called and raised exception. Error of end handler should not be handled in this case.

RECOMMENDATION

Rewrite the handler:

```
let userConfirmation = false;
try {
    const txParams = payload.params[0];
    userConfirmation = await this.askForTransactionConfirm(txParams);
} catch (err) {
    end(err, null);
    return;
}
if (userConfirmation) {
    next();
} else {
    const err = new Error('User cancel transaction');
    end(err, null);
}
```

Snippet 5. Possible fix for ConfirmSubprovider handler

UPDATES

- *Mar 18, 2022*: This issue has been acknowledged and fixed by the Sky Mavis team.

2.3.3. Wrong early-return condition INFORMATIVE

The if condition on line *22* is wrong, correct operator should be <=.

```
26  // find tab by popup pattern and switch to it
27    chrome.tabs.query(
28      {
29        url: url + 'popup.html',
30      },
31      tabs => {
32        if (tabs.length < 0) return;
33        chrome.tabs.update(tabs[0].id, { active: true });
34      },
35    );
```

Snippet 6. extension-wallet-main/public/trezor-usb-permissions.js

RECOMMENDATION

Update the operator from < to <=.

UPDATES

- *Mar 18, 2022*: This issue has been acknowledged and fixed by the Sky Mavis team.

2.4. Possible enhancements

2.4.1. Weak authentication in private npm registry

Consider using a strong password (combination of numbers, letters, and special characters) for your npm registry password.

UPDATES

- *Mar 18, 2022*: This issue has been acknowledged and fixed by the Sky Mavis team.

2.4.2. Use noopener in window.open

Consider adding noopener to all window.open functions to mitigate tab-nabbing when open third-party link in new tab.

2.4.3. Consider using string search instead of Regex

Consider using string search instead of Regex to avoid confusing user while their search term contains special characters like ?, '.', '*', etc...

```
const pattern = new RegExp(`.*${searchValue}.*`, 'i');
```

Snippet 7. AccountSelection.tsx:28, AssetSelection.tsx:29

For example, an user needs to search for a name like a. (*a and dot*) then it will show all names include letter a appearing before the last position instead of names exactly contains a. (*a and dot*).

UPDATES

- *Mar 18, 2022*: This issue has been acknowledged and fixed by the Sky Mavis team.

2.4.4. Allow listeners to be notified when then are removed

When network controller changing to a new provider, all listener will be silently removed:

```
setProvider(config: NetworkConfig, subProviders: Subprovider[]) {
    this.removeAllListeners();
```

Snippet 8. background/network/NetworkController.ts

This may cause any ongoing transaction listener to misbehave, consider adding a mechanic so that listeners can acknowledge this event.

UPDATES

- *Mar 18, 2022*: This issue has been acknowledged and fixed by the Sky Mavis team.

3. VERSION HISTORY

Version	Date	Status/Change	Created by
1.0	*Sep 28, 2021*	Private Report	Verichains Lab
1.1	*Mar 18, 2022*	Public Report	Verichains Lab

Table 3. Report versions history