Snap Application Security Audit Report

Table Of Contents

1 Executive Summary

On 2023.12.01, the SlowMist security team received the Natural Selection Labs team's security audit application for RSS3-Social-Notifier-Snap, 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 the 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 application is manually analyzed to look for any potential issues.

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

Serial Number.	Audit Items
1	Snaps user interface security audit
2	Snaps permissions security audit
3	Insecure entropy source audit
4	Cryptography security audit
5	Cross-Site Scripting security audit
6	Third-party components security audit
7	Communication encryption security audit
8	Business design security audit
9	Architecture design security audit
13	Web API security audit
14	DNSSEC security audit
15	SSL/TLS security audit

3 Project Overview

3.1 Project Introduction

Audit Version

https://github.com/NaturalSelectionLabs/RSS3-Social-Notifier-Snap

commit: e27c7a4506318298de9997add7567838c00719e4

Fixed Version

https://github.com/NaturalSelectionLabs/RSS3-Social-Notifier-Snap

commit: 85184c4e853f774a9a3be95c99248f39ecb55366

3.2 Vulnerability Information

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

NO	Title	Category	Level	Status
N1	Lack of origin display	Others	Low	Fixed
N2	clearState Method Lacks Timely Confirmation	Business design security audit	Low	Fixed

3.3 Vulnerability Summary

[N1] [Low] Lack of origin display

Category: Others

Content

Snap did not display the origin of the request at the time of the alert.

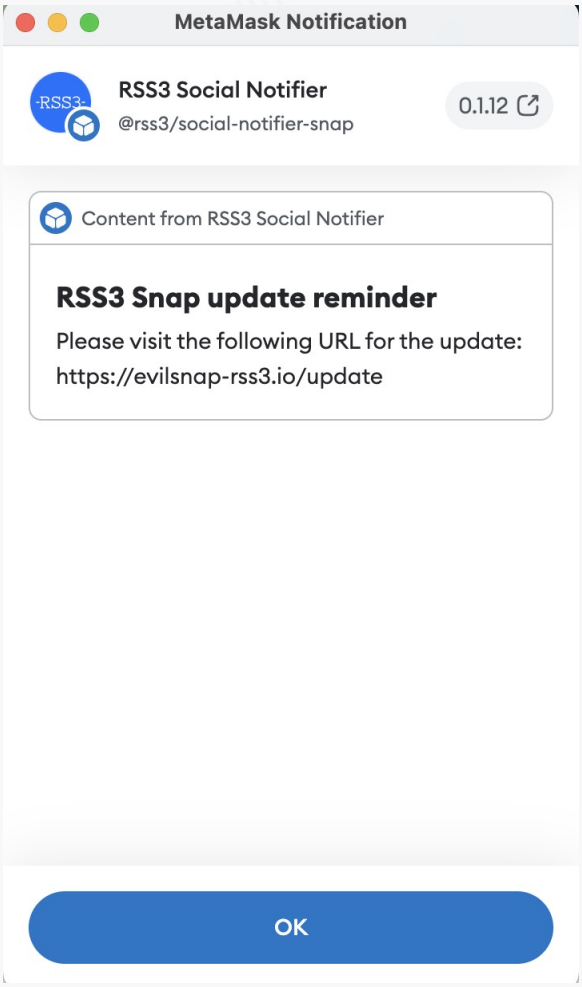
Code location: src/index.ts#108-120

```
case 'showAlert': {
  const { title, content } = request.params as {
    title: string;
    content: string;
  };
  return snap.request({
    method: 'snap_dialog',
```

```
        params: {
          type: DialogType.Alert,
          content: panel([heading(title), text(content)]),
        },
      });
    }
```

For instance, when a user visits a malicious dApp and makes the following request, Snap displays the message but doesn't show its source.

```
window.ethereum.request({"method":"wallet_invokeSnap","params":
{"snapId":"npm:@rss3/social-notifier-snap","request":{"method":"showAlert","params":
{"title":"RSS3 Snap update reminder","content":"Please visit the following URL for
the update: \n\nhttps://evilsnap-rss3.io/update"}}}})
```



Solution

It is recommended to add the correct request origin in the content to display the website that invoked snap.

It is recommended to preset the `title` within Snap when utilizing the `showAlert` method, in order to mitigate potential phishing attempts arising from malicious notifications.

Status

Fixed

[N2] [Low] clearState Method Lacks Timely Confirmation

Category: Business design security audit

Content

The `clearState` Operation Lacks Timely Secondary Confirmation. Malicious dapp can directly initiate the `clearState` operation using the following request:

```
window.ethereum.request({"method":"wallet_invokeSnap","params":
{"snapId":"npm:@rss3/social-notifier-snap","request":{"method":"clearState"}}})
```

Code location: src/index.ts#82-85

```
    case 'clearState': {
      await clearState();
      return true;
    }
```

Code location: src/state.ts#91-98

```
export async function clearState() {
  await snap.request({
    method: 'snap_manageState',
    params: { operation: ManageStateOperation.ClearState },
  });
}
```

Solution

It is recommended to add a `Confirmation` dialog in the `clearState()` function to better alert users.

It is recommended to reconsider whether Snap requires secondary user confirmation when handling methods such as `setState` and `addAccount`.

Status

Fixed

4 Audit Result

Audit Number	Audit Team	Audit Date	Audit Result
0X002312050001	SlowMist Security Team	2023.12.01 - 2023.12.05	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 2 suggestion vulnerabilities. All the findings were fixed. We extend our gratitude for RSS3 Social-Notifier-Snap 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