



Snap Application 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 2022.11.02, the SlowMist security team received the BTCSnap team's security audit application for BTCSnap, 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	HSTS security audit
2	X-Content-Type-Options security audit
3	X-XSS-Protection security audit
4	CSP security audit
5	HTTP cookies security audit
6	Web front-end storage security audit
7	Clickjacking protection security audit
8	XSS defense security audit
9	CSRF defense security audit
10	Third-party resource security audit
11	CORS security audit
12	postMessage security audit
13	Web API security audit
14	DNSSEC security audit
15	SSL/TLS security audit
16	Signature security audit
17	Deposit/Transfer security audit

Serial Number.	Audit Items
18	Transaction malleability security audit
19	Snap application security audit

3 Project Overview

3.1 Project Introduction

Fixed Version

Snap application:

<https://github.com/snapdao/btcsnap/tree/master>

commit: 66d6136523c5ca91760d5839dc060798c31086c6

Website:

<https://btc.justsnap.io>

3.2 Vulnerability Information

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

NO	Title	Category	Level	Status
N1	Missing Strict-Transport-Security security configuration	HSTS security audit	Low	Fixed
N2	Missing the X-Content-Type-Options security configuration	X-Content-Type-Options security audit	Suggestion	Fixed
N3	Missing the X-XSS-Protection security configuration	X-XSS-Protection security audit	Suggestion	Fixed
N4	Missing CSP security configuration	CSP security audit	Suggestion	Fixed

NO	Title	Category	Level	Status
N5	Clickjacking Security Risk	Clickjacking protection security audit	Low	Fixed
N6	Missing DNS security policy	DNSSEC security audit	Suggestion	Fixed
N7	Missing origin domain name when the transfer signature	Signature security audit	Low	Fixed
N8	npm project resource leak	Others	Low	Fixed
N9	Missing blockchain network display when signature	Signature security audit	Low	Fixed
N10	Missing authorization to switch networks in Snap	Snap application security audit	Low	Fixed
N11	Unfriendly transaction page display	Signature security audit	Suggestion	Fixed

3.3 Vulnerability Summary

[N1] [Low] Missing Strict-Transport-Security security configuration

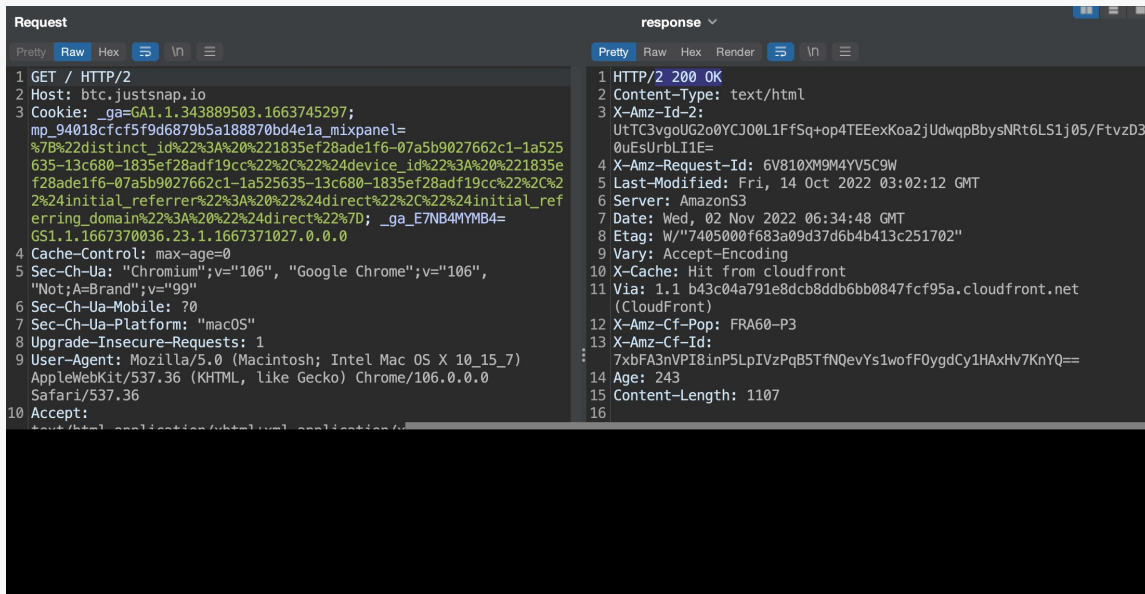
Category: HSTS security audit

Content

When accessing using <http://btc.justsnap.io>, it will force a jump to <https://btc.justsnap.io>.

Request	Response
<pre> 1 GET / HTTP/1.1 2 Host: btc.justsnap.io 3 Upgrade-Insecure-Requests: 1 4 User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/106.0.0.0 Safari/537.36 5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9 6 Accept-Encoding: gzip, deflate 7 Accept-Language: en 8 Connection: close 9 10 </pre>	<pre> 1 HTTP/1.1 301 Moved Permanently 2 Server: CloudFront 3 Date: Wed, 02 Nov 2022 06:32:13 GMT 4 Content-Type: text/html 5 Content-Length: 167 6 Connection: close 7 Location: https://btc.justsnap.io/ 8 X-Cache: Redirect from cloudfront 9 Via: 1.1 ab21b6436bc1d51d57b228ad39b1fa54.cloudfront.net (CloudFront) 10 X-Amz-Cf-Pop: FRA60-P3 11 X-Amz-Cf-Id: U_ly2iJEyGNRKMcu0B--v2z6pdNpHPF81SrRtVcmxfNh-BD8JSjMTQ== 12 X-XSS-Protection: 1 13 X-Frame-Options: DENY 14 X-Content-Type-Options: nosniff </pre>

When visiting <https://btc.justsnap.io>, it is found that the HSTS security policy is not configured on the server.



Solution

It is recommended to configure HSTS security policy, HTTP response header configuration such as:

Strict-Transport-Security: max-age=63072000; includeSubDomains; preload

Reference: <https://hstspreload.org/>

Status

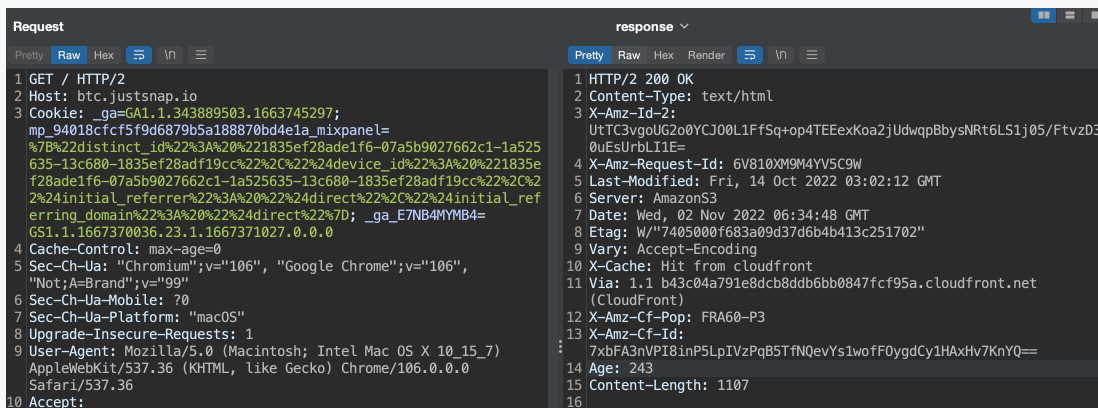
Fixed

[N2] [Suggestion] Missing the X-Content-Type-Options security configuration

Category: X-Content-Type-Options security audit

Content

The HTTP response header is missing the X-Content-Type-Options security configuration.



Solution

Browser sniff behavior may cause risks such as: a picture resource, the content is not a picture resource but a string of strings, such as:

```
<script>alert(1);</script>
```

This can lead to XSS attacks. Of course, the appearance of this kind of attack still needs to meet certain scenarios and modern browsers have different coping strategies. But the best security practice recommends to completely eliminate this risk and configure the HTTP response header:

```
X-Content-Type-Options: nosniff
```

Status

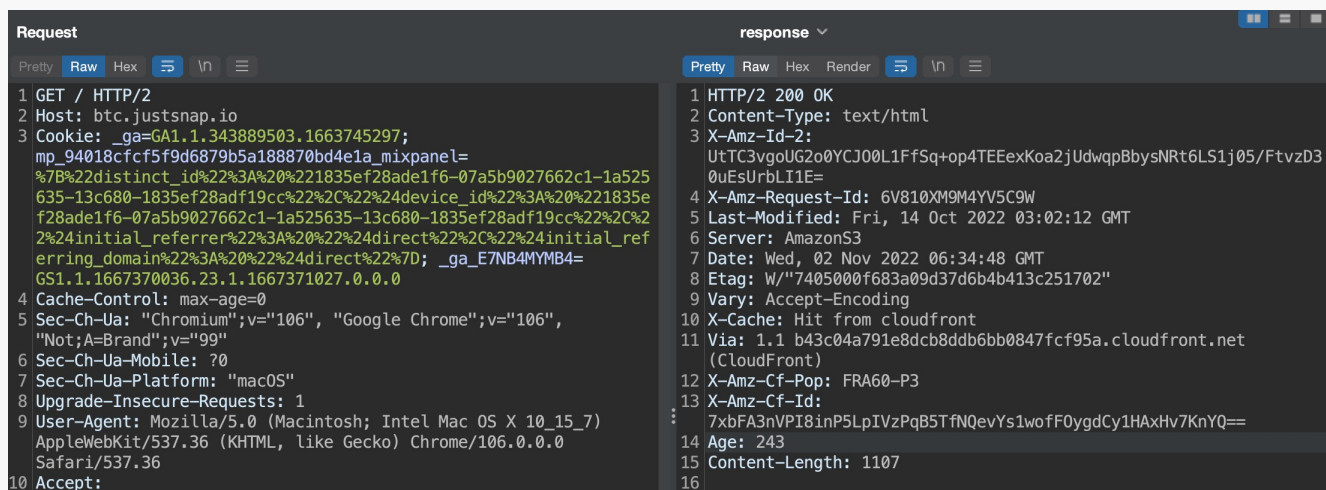
Fixed

[N3] [Suggestion] Missing the X-XSS-Protection security configuration

Category: X-XSS-Protection security audit

Content

The HTTP return package is missing the X-XSS-Protection security configuration policy.



Solution

It is recommended to add the X-XSS-Protection security configuration policy.

Status

Fixed

[N4] [Suggestion] Missing CSP security configuration

Category: CSP security audit

Content

The HTTP return packet did not find the CSP security configuration policy.

Solution

It is recommended to add a CSP security configuration policy.

Reference: https://cheatsheetseries.owasp.org/cheatsheets/Content_Security_Policy_Cheat_Sheet.html

<https://content-security-policy.com>

Status

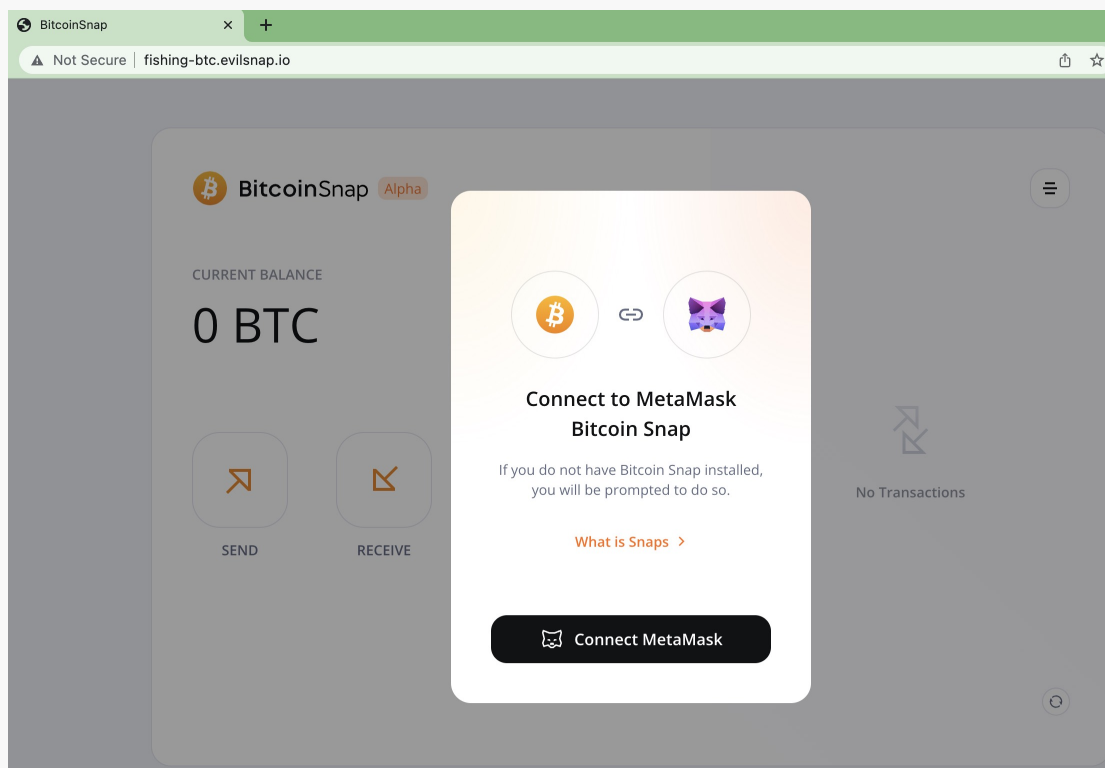
Fixed

[N5] [Low] Clickjacking Security Risk

Category: Clickjacking protection security audit

Content

The site lacks the X-FRAME-OPTIONS security configuration, which allows the page to be embedded in an iframe tag, which can be used for phishing.



The source code of the phishing website is as follows:

```
<!DOCTYPE html>
<!-- This is a test fishing site !!-->
<html>
  <head>...</head>
  <body style="margin:0px;padding:0px;overflow:hidden">
    <iframe src="https://btc.justsnap.io/" frameborder="0" style="overflow:hidden;height:100%;width:100%" height="100%" width="100%">
      <#document
        <!DOCTYPE html>
        <html lang="en">
          <head>...</head>
          <body class="dimnable dimmed">...</body> == $0
        </html>
      </iframe>
    </body>
  </html>
```

Solution

The essence of countering Clickjacking is to counteract that your own service is embedded in the iframe/frame method of pages of other domains. The HTTP response header configuration:

```
X-FRAME-OPTIONS: SAMEORIGIN
```

or

```
X-FRAME-OPTIONS: DENY
```

Status

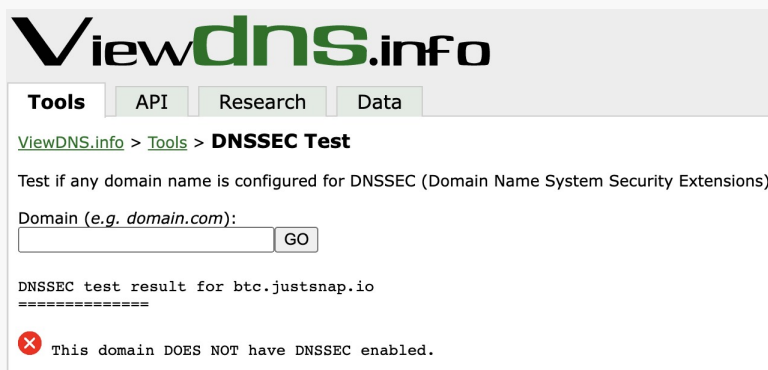
Fixed

[N6] [Suggestion] Missing DNS security policy

Category: DNSSEC security audit

Content

It is not detected that the DNS security policy is enabled by the domain name resolution service provider.



ViewDNS.info

Tools API Research Data

ViewDNS.info > Tools > **DNSSEC Test**

Test if any domain name is configured for DNSSEC (Domain Name System Security Extensions).

Domain (e.g. domain.com):

DNSSEC test result for btc.justsnap.io
=====

✗ This domain DOES NOT have DNSSEC enabled.

Solution

It is recommended to enable DNS security in the domain name resolution service provider.

For example, how AWS enables DNSSEC:

https://docs.aws.amazon.com/en_us/Route53/latest/DeveloperGuide/resolver-dnssec-validation.html

Status

Fixed

[N7] [Low] Missing origin domain name when the transfer signature

Category: Signature security audit

Content

The origin's domain name is not displayed when the transfer is signed, and there are enhancement points, which may be used for phishing.

- packages/snap/src/index.ts

```
export const onRpcRequest = async({origin, request}:rpcRequest) => {
  switch (request.method) {
    case 'btc_getPublicExtendedKey':
      return getExtendedPublicKey(wallet, request.params.scriptType,
        getNetwork(request.params.network))
    case 'btc_signPsbt':
      const psbt = request.params.psbt;
      return signPsbt(wallet, psbt, getNetwork(request.params.network),
        request.params.scriptType)
    case 'btc_masterFingerprint':
      return masterFingerprint(wallet, request.params.action);
    default:
      throw new Error('Method not found.');
```

Solution

The domain name of the origin should be displayed when the transfer is signed.

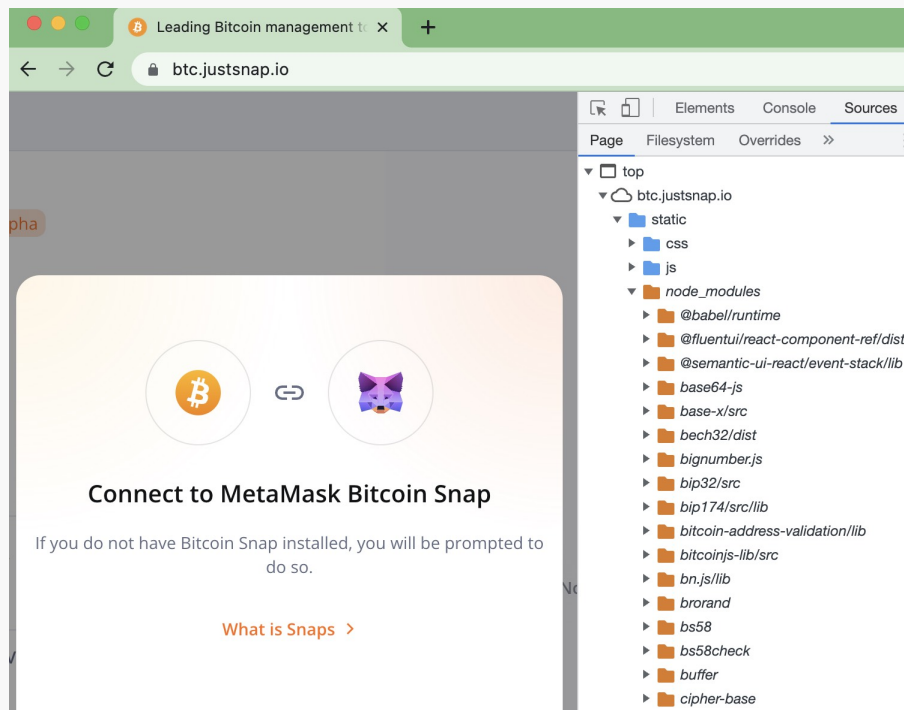
Status

Fixed

[N8] [Low] npm project resource leak

Category: Others**Content**

Compile and run the npm project will leave behind the dependency packages used, which need to be cleaned and hidden to prevent information leakage.

**Solution**

The official operation of the npm project should clear the compilation and production process files to prevent information leakage.

Status

Fixed

[N9] [Low] Missing blockchain network display when signature**Category: Signature security audit****Content**

The signature process does not show the blockchain network, and users may find it difficult to determine the network of the transaction broadcast because there is no network information

Solution

The Snap should remind the user of the network environment to which the current signature belongs.

Status

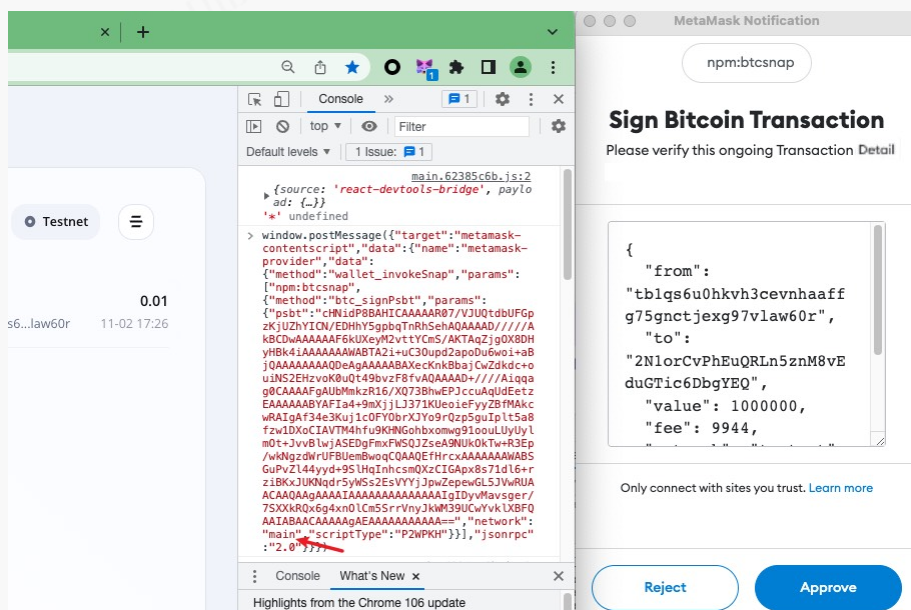
Fixed

[N10] [Low] Missing authorization to switch networks in Snap

Category: Snap application security audit

Content

The current design is that the web page determines the network (testnet/mainnet) for transaction broadcast.



Solution

It is recommended to add authorization to switch networks in Snap to ensure that network switching is determined by the user.

Status

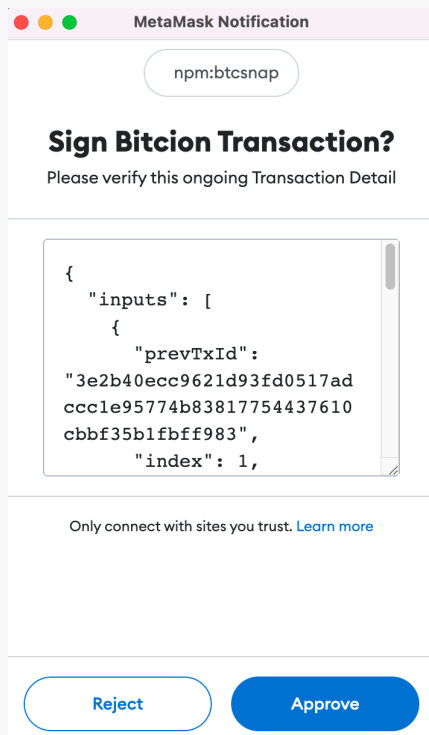
Fixed

[N11] [Suggestion] Unfriendly transaction page display

Category: Signature security audit

Content

At present, Snap does not parse the content of the transaction when displaying the transaction, and only displays the json data, which is not friendly to non-technical personnel.



Solution

It is recommended to optimize the display of the transaction.

Status

Fixed

4 Audit Result

Audit Number	Audit Team	Audit Date	Audit Result
0X002209270001	SlowMist Security Team	2022.11.02 - 2022.11.08	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 6 low risk, 5 suggestion vulnerabilities. And 6 low risk, 5 suggestion vulnerabilities were confirmed and being fixed; All other findings were fixed.

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>