



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 2023.07.10, the SlowMist security team received the UniPassID team's security audit application for UniPass ID 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	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

UniPass Snap is an innovative product designed to enhance your MetaMask experience by providing it with the power of a smart contract wallet.

Audit Version:

<https://github.com/UniPassID/UniPass-Snap>

commit: d74b81a05f317b568c808053b4f62fefac9668ff

Fixed Version:

<https://github.com/UniPassID/UniPass-Snap>

commit: b9fe1ee98e4b84bf6b0f2e8245bd7b9f7b228e99

3.2 Vulnerability Information

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

NO	Title	Category	Level	Status
N1	Bypass the restrictions on 3 gas-free payments	Web API security audit	High	Fixed
N2	originTransaction arbitrary forgery issue	Snap application security audit	High	Fixed
N3	Lack of origin domain display	Snap application security audit	Suggestion	Fixed
N4	Missing address confirmation	Snap application security audit	Suggestion	Acknowledged

NO	Title	Category	Level	Status
N5	Lack of DNSSEC protection	DNSSEC security audit	Suggestion	Fixed
N6	Incorrect storage of sensitive information	Web front-end storage security audit	Suggestion	Fixed

3.3 Vulnerability Summary

[N1] [High] Bypass the restrictions on 3 gas-free payments

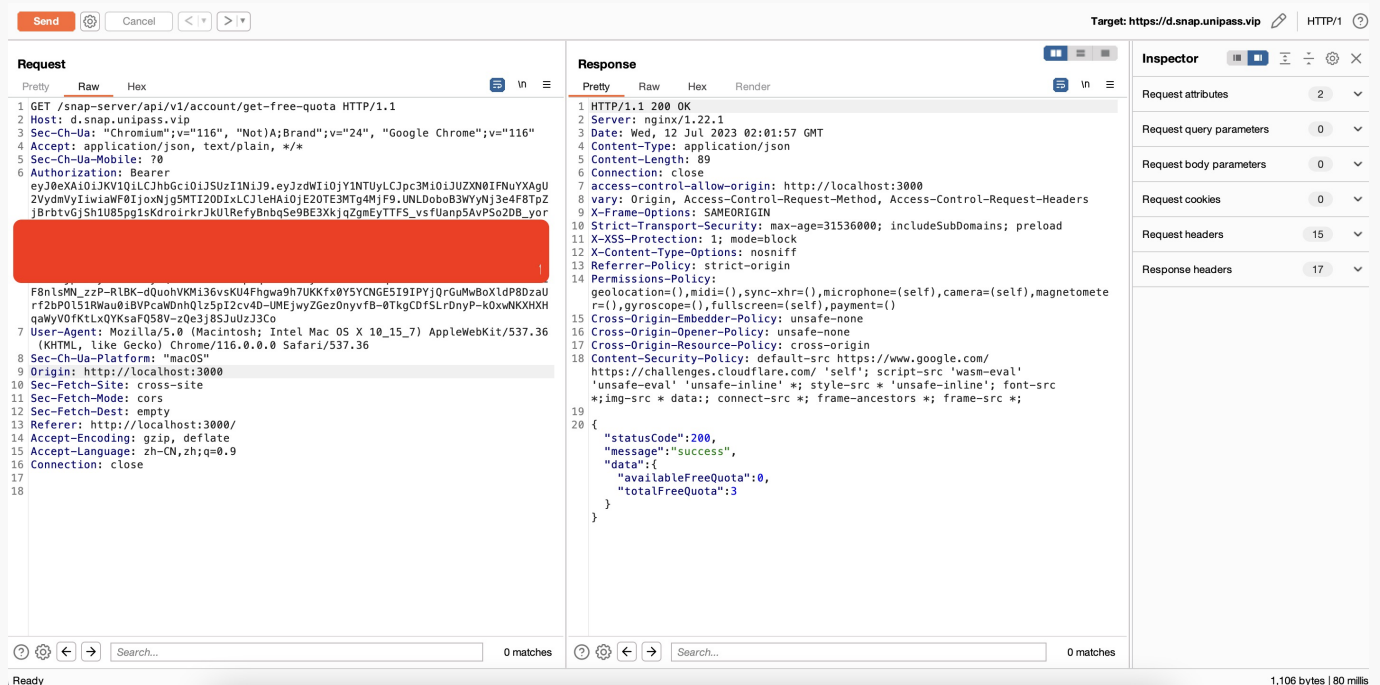
Category: Web API security audit

Content

UniPass Snap allows 3 gas-free payments per address, but additional time can be added in the following ways.

PoC:

1. Use the `/snap-server/api/v1/account/get-free-quota` interface to query availableFreeQuota and get 0.



The screenshot displays a web browser interface with a target URL of `https://d.snap.unipass.vip`. The browser shows a successful HTTP request and response for the endpoint `/snap-server/api/v1/account/get-free-quota`. The response status is 200 OK, and the body contains a JSON object with the following data:

```

{
  "statusCode": 200,
  "message": "success",
  "data": {
    "availableFreeQuota": 0,
    "totalFreeQuota": 3
  }
}

```

2. Use the current nonce request `/snap-server/api/v1/transaction/authorize-transaction-fees`

interface.

[illegible]

3. Use the `/snap-server/api/v1/account/get-free-quota` interface to query availableFreeQuota and get 4294967295, There is an underflow in availableFreeQuota.

Send

Cancel

< >

< >

Target: https://d.snap.unipass.jp HTTP/1

Ready

Request

PrettyRawHex

In

```
1 GET /snap-server/api/v1/account/get-free-quota HTTP/1.1
2 Host: d.snap.unipass.vip
3 Sec-Ch-Ua: "Chromium";v="116", "NotA]Brand";v="24", "Google Chrome";v="116"
4 Accept: application/json, text/plain, */*
5 Sec-Ch-Ua-Mobile: ?0|
6 Authorization: Bearer
eyJ0eXA1OjKV1Q1LCjhbc6I0JSUzIiwia3RlLnV1YXN0YyY1NTYuYVVCjpc3M1OjZUN0IFNuYXAgU
[REDACTED]
[REDACTED]
[REDACTED]
moMuA-KNlpZ7XR0KZVB8VDPSeJqg-hhgUwqfJqtBtPmF1Y6PsY98XglbQ8EK8ecQCz6vnS86W-
[REDACTED]
F8nlsmw_Zp-RLBK-dQuoVKM136vsKU4Fghwa9H7UKKfx0YSYCNGE5I9IPYJ0rGuMwBoXLdPB8Dza
rfzbP0L5IRWau0IBVPcaMDnhqlzp5I2cv40-UMEfyyZGezOnyvFB-0tkgcDFSLrDnyP-k0xwNKXHXH
qaWyV0fKLxQYKsaF05W-zQe3j85JuZz3Cco
7 User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/116.0.0.0 Safari/537.36
8 Sec-Ch-UA-Platform: "macOS"
9 Origin: http://localhost:3000
10 Sec-Fetch-Site: cross-site
11 Sec-Fetch-Mode: cors
12 Sec-Fetch-Dest: empty
13 Referer: http://localhost:3000/
14 Accept-Encoding: gzip, deflate
15 Accept-Language: zh-CN,zh;q=0.9
16 Connection: close
17
18
```

Response

PrettyRawHexRender

In

```
1 HTTP/1.1 200 OK
2 Server: nginx/1.22.1
3 Date: Wed, 12 Jul 2023 02:02:44 GMT
4 Content-Type: application/json
5 Content-Length: 98
6 Connection: close
7 access-control-allow-origin: http://localhost:3000
8 vary: Origin, Access-Control-Request-Method, Access-Control-Request-Headers
9 X-Frame-Options: SAMEORIGIN
10 Strict-Transport-Security: max-age=31536000; includeSubDomains; preload
11 X-XSS-Protection: 1; mode=block
12 X-Content-Type-Options: nosniff
13 Referrer-Policy: strict-origin
14 Permissions-Policy:
geolocation=(),midi=(),sync-xhr=(),microphone=(self),camera=(self),magnetomet
r=(),gyroscope=(),fullscreen=(self),payment()
15 Cross-Origin-Embedder-Policy: unsafe-none
16 Cross-Origin-Opener-Policy: unsafe-none
17 Cross-Origin-Resource-Policy: cross-origin
18 Content-Security-Policy: default-src https://www.google.com/
https://challenges.cloudflare.com/ 'self'; script-src 'unsafe-eval'
'unsafe-eval' 'unsafe-inline' *; style-src * 'unsafe-inline'; font-src
*;img-src * data; connect-src *; frame-ancestors *; frame-src *;
19
20 {
  "statusCode":200,
  "message":"success",
  "data":{"
    "availableFreeQuota":4294967295,
    "totalFreeQuota":3
  }
}
```

Inspector

Request attributes

2

Request query parameters

0

Request body parameters

0

Request cookies

0

Request headers

15

Response headers

17

Search...

0 matches

Search...

0 matches

1,115 bytes | 67 mins

Solution

It is recommended to check the server code to check the calculation of availableFreeQuota to avoid underflow issue.

Status

Fixed

[N2] [High] originTransaction arbitrary forgery issue

Category: Snap application security audit

Content

UnipassID Snap uses signTransactionMessage to sign messages when sending transactions, but does not check whether originTransaction's HASH and signTxMessage.message are consistent.

Malicious Dapp can use this issue to forge originTransaction as a transaction with a small amount of transfers. The actual message is a transaction with a large amount of transfers, thus deceiving the user into stealing the user's assets.

Code location: unipass-snap/packages/snap/src/rpc.ts

```
export async function signTransactionMessage(signTxMessage: SignTxMessageInput) {
    let panelContent: { value: string; type: NodeType.Text }[]
    const originTransaction = JSON.parse(signTxMessage.originTransaction) as
    originTransaction

    if (originTransaction.transactions.length > 1) {
        let payContent = originTransaction.transactions.map((tx, index) => {
            return [
                text(`**Payment ${index + 1}**`),
                text(`Pay ${tx.amount}
                ${getTokenSymbolByAddress(tx.token)}`),
                text(`To: ${tx.to}`)
            ]
        })
        panelContent = [
            ...payContent.flat(),
            text(
                `**Gasfee: ${
                    originTransaction.fee ?
                `${originTransaction.fee.amount} ${originTransaction.fee.symbol}` : 'Free'
                }**`
            ),
            text(`**Chain: ${originTransaction.chain}**`)
        ]
    } else {
        panelContent = [
            text(
                `**Pay ${originTransaction.transactions[0].amount}
                ${getTokenSymbolByAddress(
                    originTransaction.transactions[0].token
                )} on ${originTransaction.chain}**`
            ),
            text(`**To** ${originTransaction.transactions[0].to}`),

```



```

        text(
            `**Gasfee: ${
                originTransaction.fee ?
`${originTransaction.fee.amount} ${originTransaction.fee.symbol}` : 'Free'
            }**`
        )
    ]
}

let result = await snap.request({
    method: 'snap_dialog',
    params: {
        type: 'confirmation',
        content: panel(panelContent)
    }
})

if (result) return signMessage(arrayify(signTxMessage.message))
throw new Error('User reject to sign transaction')
}

```

Solution

It is recommended to check the HASH of originTransaction in snap to ensure consistency with the message.

Status

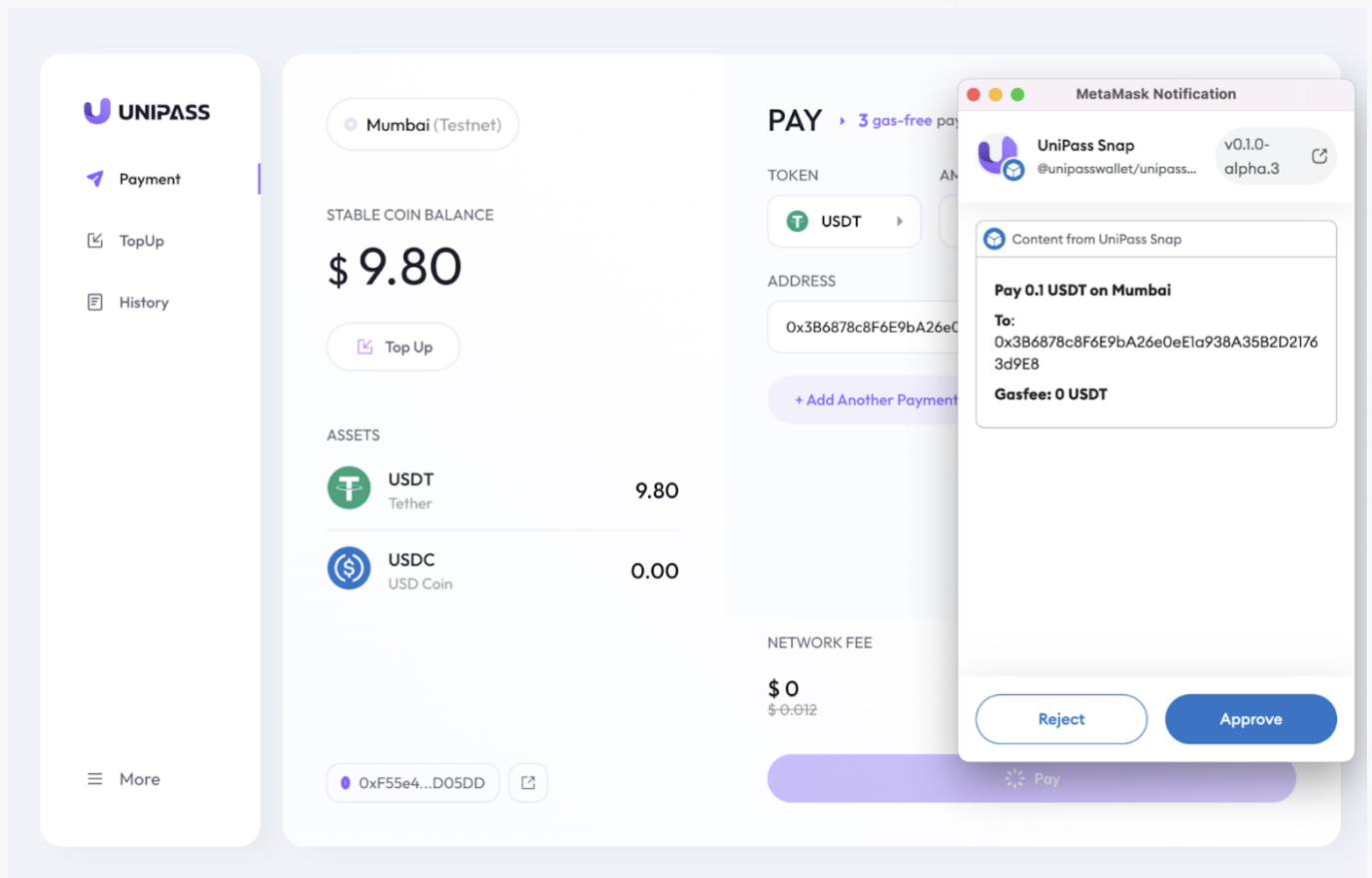
Fixed

[N3] [Suggestion] Lack of origin domain display

Category: Snap application security audit

Content

UniPass Snap does not display the origin station information that initiated the request.



Solution

It is recommended that the origin domain of the request be displayed in the Notification so that the user can confirm the source of the signed request.

Status

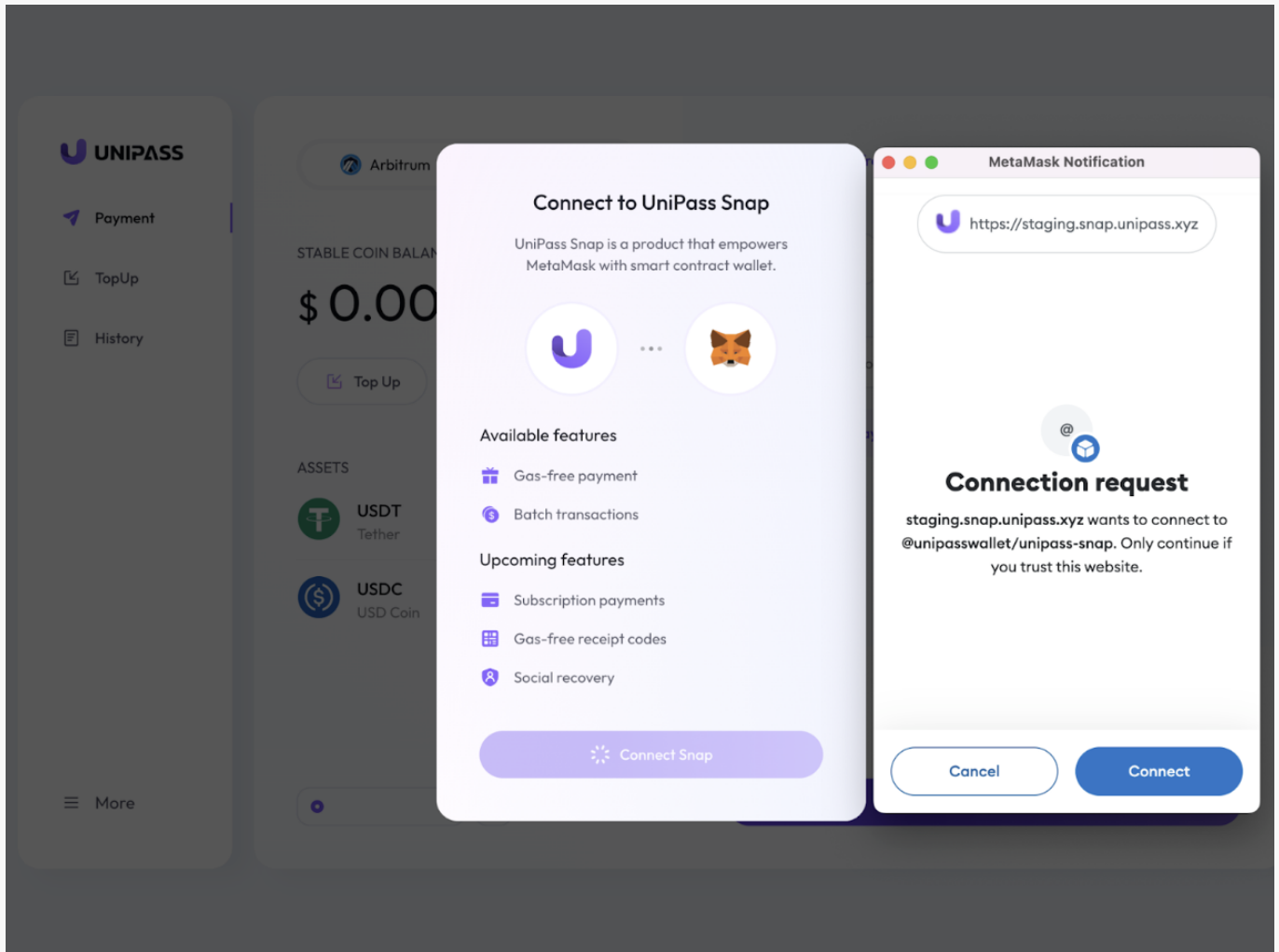
Fixed

[N4] [Suggestion] Missing address confirmation

Category: Snap application security audit

Content

After using Unipass Snap to generate the address, the address is not displayed in Notification so that the user can't confirm with the address that the Dapp website displays.



Solution

It is recommended to display the generated address in the Notification so that users can verify it with the address displayed on the Dapp website.

Status

Acknowledged

[N5] [Suggestion] Lack of DNSSEC protection

Category: DNSSEC security audit

Content

The website lacks DNSSEC protection, so it is easy to be exploited by malicious attackers leading to DNS hijacking.

[ViewDNS.info](#) > [Tools](#) > **DNSSEC Test**

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

Domain (e.g. domain.com):

	GO
--	----

DNSSEC test result for staging.snap.unipass.xyz

■■■■■■■■■■



This domain DOES NOT have DNSSEC enabled.

Solution

It is recommended to enable DNS security at your domain name resolution provider.

Status

Fixed

[N6] [Suggestion] Incorrect storage of sensitive information

Category: Web front-end storage security audit

Content

Snap will store user's sensitive information such as `accessToken` in `localStorage` after login, which can easily lead to user's account privileges being stolen.

Storage	up__currentChainId	80001
	theme	dark
▼ Local Storage	up__accessToken	eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.e
https://staging.snap.unipass.		
▶ Session Storage		
IndexedDB		
Web SQL		
▶ Cookies		
Private State Tokens		
Interest Groups		
▶ Shared Storage	1 eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdWIiOiJYNTM4Ljpc3MiOiJlbnVbmQXNzIFNuYXAiLCJpYXQjE2ODkzMdYwMTAsImV4cC	
Cache Storage		
Background Services		
Back/forward cache		
↕ Background Fetch		
⏸ Background Sync		

Solution

It is recommended to store user's accessToken in sessionStorage instead of localStorage.

Status

Fixed

4 Audit Result

Audit Number	Audit Team	Audit Date	Audit Result
0X002307170002	SlowMist Security Team	2023.07.10 - 2023.07.17	Passed

Summary conclusion: The SlowMist security team used a manual and SlowMist team's analysis tool to audit the project, during the audit work we found 2 high-risk vulnerabilities and 4 suggestions. 2 high-risk vulnerabilities and 3 suggestions 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>