Web Front-end Security Audit Report

Table Of Contents

1 Executive Summary

On 2023.08.03, the SlowMist security team received the SenderWallet team's security audit application for DappAuto, 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 box 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:

NO.	Audit Items	Result
1	HSTS security audit	Passed
2	X-Content-Type-Options security audit	Passed
3	X-XSS-Protection security audit	Passed
4	CSP security audit	Passed
5	HTTP cookies security audit	Passed
6	Web front-end storage security audit	Passed
7	Clickjacking protection security audit	Passed
8	XSS defense security audit	Passed
9	CSRF defense security audit	Passed
10	Third-party resource security audit	Passed
11	CORS security audit	Passed
12	postMessage security audit	Passed
13	Web API security audit	Passed
14	DNSSEC security audit	Passed
15	SSL/TLS security audit	Passed

NO.	Audit Items	Result
16	Others	Passed

3 Project Overview

3.1 Project Introduction

Audit Version:

https://github.com/DappAuto/frontend/tree/slowmist

Audited commit: 9aa5e7f557367d87a138897c16a5c1d2862334ab

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	CSP policy is not enabled	CSP security audit	Suggestion	Fixed
N5	Clickjacking Security Risk	Clickjacking protection security audit	Low	Fixed
N6	Risks of third-party JS	Third-party resource security audit	Suggestion	Fixed

NO	Title	Category	Level	Status
N7	Missing DNS security policy	DNSSEC security audit	Suggestion	Acknowledged

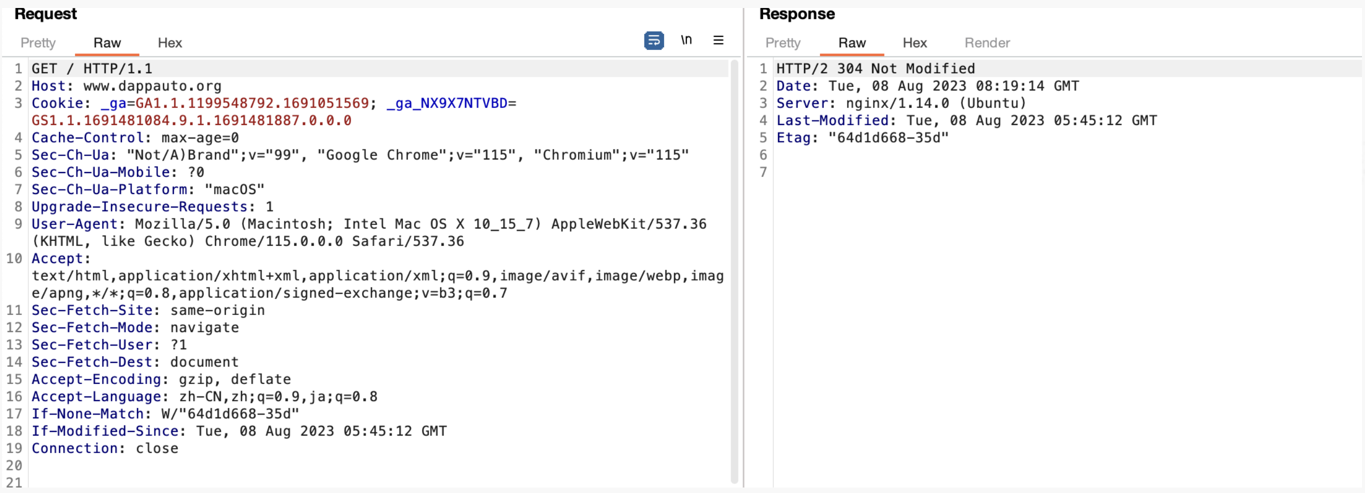
3.3 Vulnerability Summary

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

Category: HSTS security audit

Content

When visiting https://www.dappauto.org, it is found that the HSTS security policy is not configured on the server.



```
GET / HTTP/1.1
Host: www.dappauto.org
Cookie: _ga=GA1.1.1199548792.1691051569; _ga_NX9X7NTVBD=
GS1.1.1691481084.9.1.1691481887.0.0.0
Cache-Control: max-age=0
Sec-Ch-Ua: "Not/A)Brand";v="99", "Google Chrome";v="115", "Chromium";v="115"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "macOS"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/115.0.0.0 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,imag
e/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,ja;q=0.8
If-None-Match: W/"64d1d668-35d"
If-Modified-Since: Tue, 08 Aug 2023 05:45:12 GMT
Connection: close
```

```
HTTP/2 304 Not Modified
Date: Tue, 08 Aug 2023 08:19:14 GMT
Server: nginx/1.14.0 (Ubuntu)
Last-Modified: Tue, 08 Aug 2023 05:45:12 GMT
Etag: "64d1d668-35d"
```

Solution

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

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

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

Reference: https://hstspreload.org/

Status

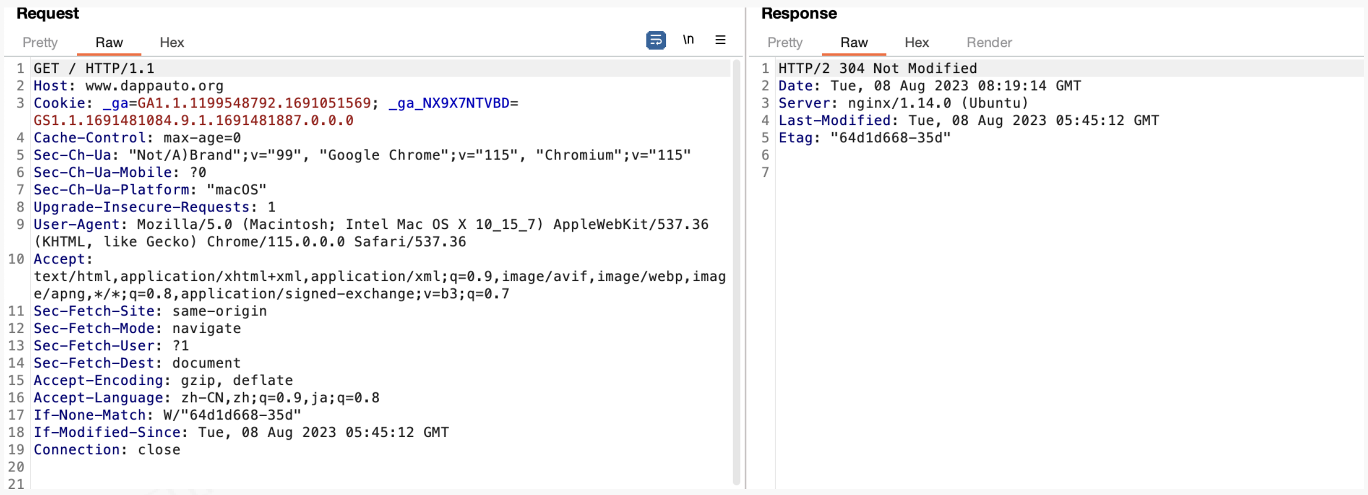
Fixed

[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.



```
GET / HTTP/1.1
Host: www.dappauto.org
Cookie: _ga=GA1.1.1199548792.1691051569; _ga_NX9X7NTVBD=
GS1.1.1691481084.9.1.1691481887.0.0.0
Cache-Control: max-age=0
Sec-Ch-Ua: "Not/A)Brand";v="99", "Google Chrome";v="115", "Chromium";v="115"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "macOS"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/115.0.0.0 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,imag
e/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,ja;q=0.8
If-None-Match: W/"64d1d668-35d"
If-Modified-Since: Tue, 08 Aug 2023 05:45:12 GMT
Connection: close
```

```
HTTP/2 304 Not Modified
Date: Tue, 08 Aug 2023 08:19:14 GMT
Server: nginx/1.14.0 (Ubuntu)
Last-Modified: Tue, 08 Aug 2023 05:45:12 GMT
Etag: "64d1d668-35d"
```

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

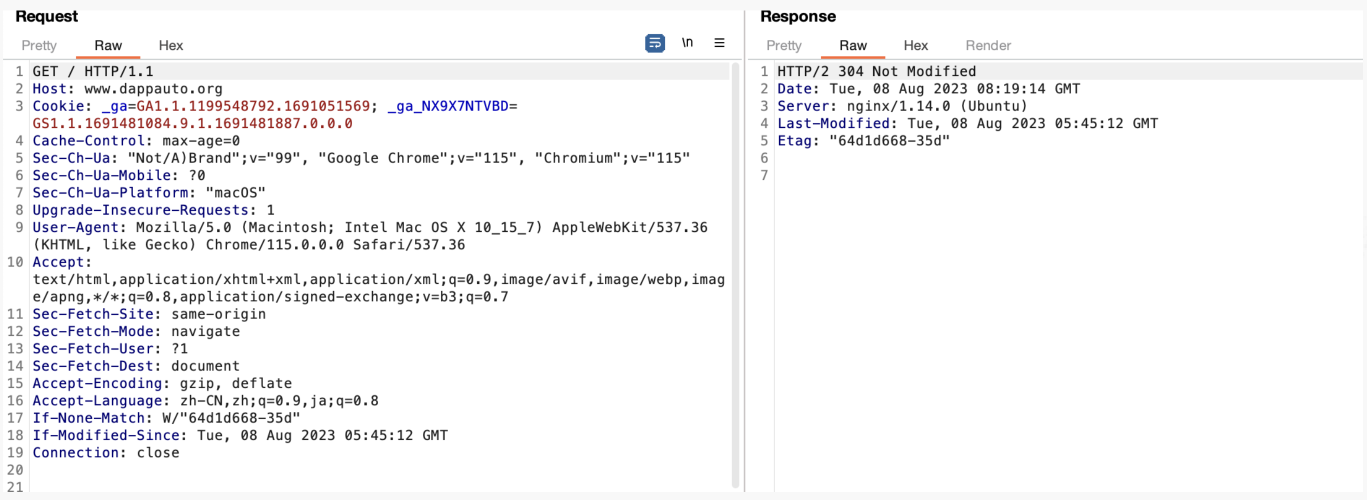
Fixed

[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.


Request

Pretty Raw Hex

```
GET / HTTP/1.1
Host: www.dappauto.org
Cookie: _ga=GA1.1.1199548792.1691051569; _ga_NX9X7NTVBD=
GS1.1.1691481084.9.1.1691481887.0.0.0
Cache-Control: max-age=0
Sec-Ch-Ua: "Not/A)Brand";v="99", "Google Chrome";v="115", "Chromium";v="115"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "macOS"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/115.0.0.0 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,imag
e/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,ja;q=0.8
If-None-Match: W/"64d1d668-35d"
If-Modified-Since: Tue, 08 Aug 2023 05:45:12 GMT
Connection: close
```

Response

Pretty Raw Hex Render

```
HTTP/2 304 Not Modified
Date: Tue, 08 Aug 2023 08:19:14 GMT
Server: nginx/1.14.0 (Ubuntu)
Last-Modified: Tue, 08 Aug 2023 05:45:12 GMT
Etag: "64d1d668-35d"
```

Solution

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

Status

Fixed

[N4] [Suggestion] CSP policy is not enabled

Category: CSP security audit

Content

CSP policy is not enabled on the front end.

Solution

It is recommended that CSP policies be enable.

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

https://content-security-policy.com

Status

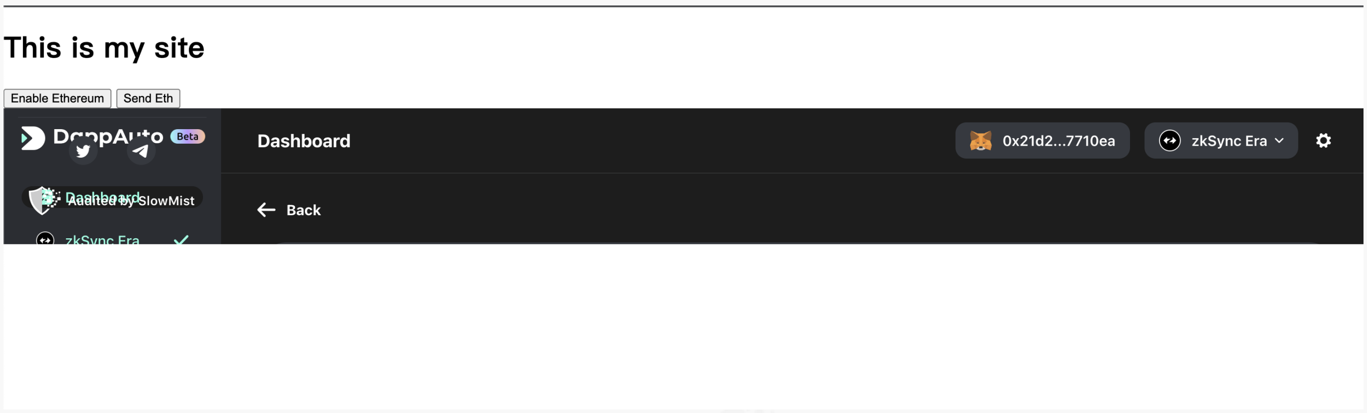
Fixed

[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 iframetag, which can be used for phishing.



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] Risks of third-party JS

Category: Third-party resource security audit

Content

Be wary of any third-party JavaScript/CSS/graphics links introduced in the web front-end, especially JavaScript, which could lead to the third-party being blacked out and the JavaScript being planted with malicious code that could lead to front-end attacks against users, such as hijacking a user's wallet address.

Solution

One of the nice security features of HTML5 can be utilised: the integrity attribute in tags (the SRI mechanism).

For example:

```
<script src="https://example.com/example-framework.js" integrity="sha384-
Li9vy3DqF8tnTXuiaAJuML3ky+er10rcgNR/VqsVpcw+ThHmYcwiB1pbOxEbzJr7"
crossorigin="anonymous"></script>
```

integrity supports sha256, sha384, sha512. If a third-party JavaScript resource does not satisfy integrity's hash integrity check, it will not be loaded, which is a good way to keep unintended code from being executed. This is a good way to prevent unintended code execution. However, the use of this mechanism requires that the target resource supports CORS responses.

Reference: https://www.w3.org/TR/SRI/ Reference: https://www.srihash.org/

Status

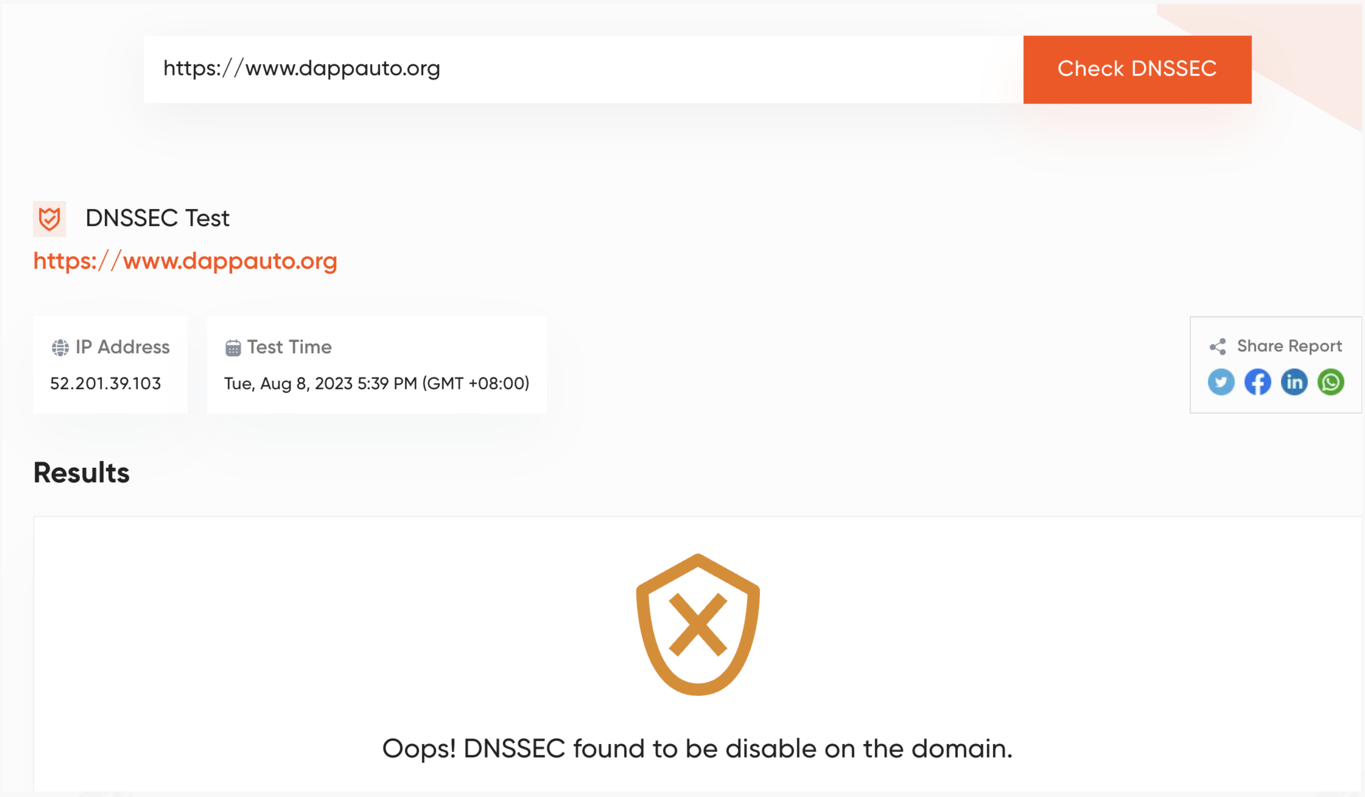
Fixed

[N7] [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.



https://www.dappauto.org

DNSSEC Test
https://www.dappauto.org

IP Address
52.201.39.103

Test Time
Tue, Aug 8, 2023 5:39 PM (GMT +08:00)

Results

Oops! DNSSEC found to be disable on the domain.

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

Acknowledged

4 Audit Result

Audit Number	Audit Team	Audit Date	Audit Result
0X002308090001	SlowMist Security Team	2023.08.03 - 2023.08.09	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 low risk, 5 suggestion vulnerabilities. All bugs have been fixed except N7.

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.