



# Web Front-end Security Audit Report



# Table Of Contents

<b>1 Executive Summary</b>	_____
<b>2 Audit Methodology</b>	_____
<b>3 Project Overview</b>	_____
3.1 Project Introduction	_____
3.2 Vulnerability Information	_____
3.3 Vulnerability Summary	_____
<b>4 Audit Result</b>	_____
<b>5 Statement</b>	_____

# 1 Executive Summary

On 2023.12.27, the SlowMist security team received the Xstro team's security audit application for Xstro.io, 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
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

NO.	Audit Items
16	Wallet Connect security audit
17	Others

## 3 Project Overview

### 3.1 Project Introduction

#### Audit scope

website :

<https://www.xstro.io/> ( including [xstro.io/staking](https://www.xstro.io/staking) and [xstro.io/portal](https://www.xstro.io/portal) )

API :

[api.xgamefi.com](https://api.xgamefi.com)

Functional Modules :

- X-Point Module
- Staking Module
- Referral Module
- Leaderboard Module
- Wallet Connect Module

### 3.2 Vulnerability Information

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

NO	Title	Category	Level	Status
N1	Third-party resource security issue	Third-party resource security audit	Suggestion	Fixed

NO	Title	Category	Level	Status
N2	CORS security audit issue	CORS security audit	Suggestion	Fixed
N3	CSP security audit issue	CSP security audit	Suggestion	Acknowledged
N4	Incomplete Logout Issue	Web API security audit	Low	Fixed
N5	Unverified Signatures for 'Stake' and 'Unstake'	Web API security audit	Low	Fixed

### 3.3 Vulnerability Summary

#### [N1] [Suggestion] Third-party resource security issue

**Category:** Third-party resource security audit

##### Content

Be cautious of any third-party JavaScript/CSS/image, etc., link resources introduced in the web frontend, especially JavaScript. This may lead to blacklisting due to the third-party being compromised, resulting in the implantation of malicious code in JavaScript, leading to frontend attacks targeting users, such as wallet address hijacking. If it's necessary to use third-party JavaScript resources, enabling a good security mechanism in HTML5 is advisable: the 'integrity' attribute inside the

We consider Google's third-party script to be secure.

However, the website also references other third-party resources, which are not particularly trustworthy.

##### Solution

[SRI Hash Generator](#) is an online tool used for generating SRI (Subresource Integrity) hash values.

For example:

```
<script src="https://cdn.jsdelivr.net/npm/@walletconnect/web3-provider@1.6.5/dist/umd/index.min.js" integrity="sha384-UgppGJjGfByOmlkLz4URiTiJFj7FuWXsVyro2OWEMNBnIiUXu0sbjXMSPfez+EW" crossorigin="anonymous"></script>
```

## Status

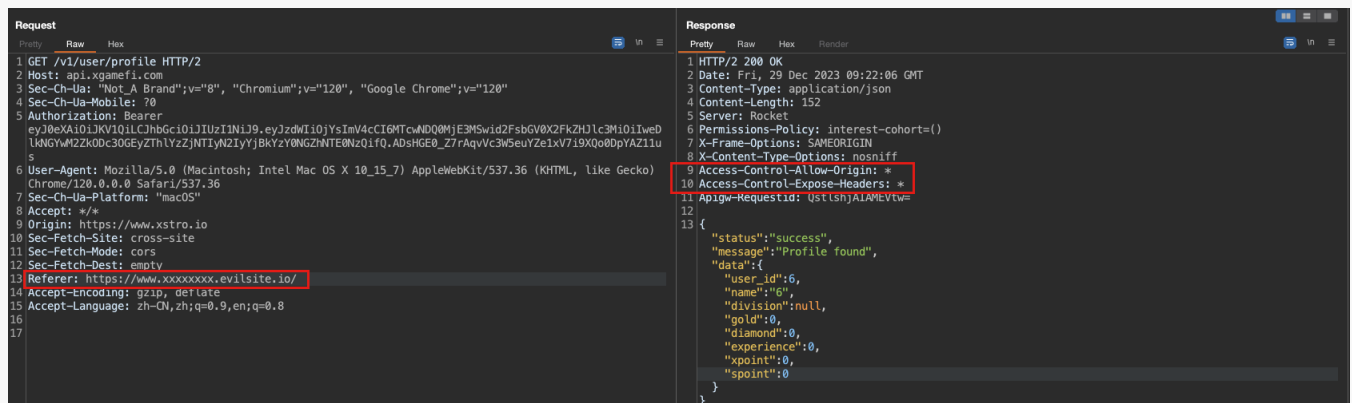
Fixed

### [N2] [Suggestion] CORS security audit issue

Category: CORS security audit

#### Content

Add an unexpected **Origin** header with the **Access-Control-Allow-Origin** header set to **\***, allowing access from all origins by default.



#### Solution

It is recommended to configure a whitelist for the "Origin" header.

## Status

Fixed

### [N3] [Suggestion] CSP security audit issue

Category: CSP security audit

#### Content

The website lacks proper CSP (Content Security Policy) security configuration.

CSP is an effective defense in depth technique to mitigate the risk of vulnerabilities such as Cross Site Scripting (XSS) and Clickjacking.

#### Solution

It is recommended to implement CSP (Content Security Policy) for enhanced security.

## Status

Acknowledged; We have communicated with the project team and will not perform a Content Security Policy check.

The project team is aware that "Content Security Policy" has not been configured.

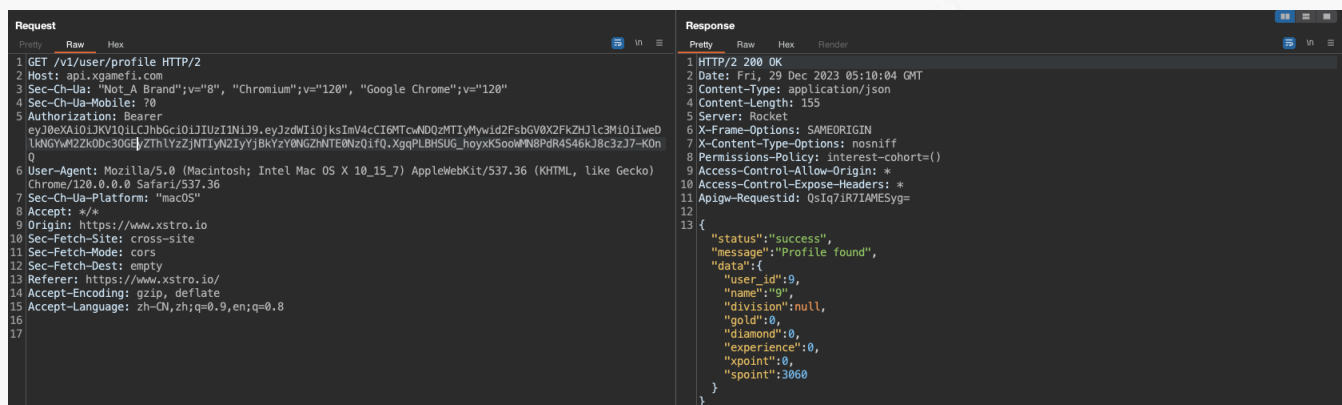
### [N4] [Low] Incomplete Logout Issue

**Category: Web API security audit**

## Content

When users click 'LOGOUT' on the frontend page to log out, the actual token remains unrevoked.

As shown in the image below, the token remains valid after logout, allowing for query requests and other operations.



### Solution

After logout, the server should effectively invalidate the original token information.

## Status

Fixed

## [N5] [Low] Unverified Signatures for 'Stake' and 'Unstake'

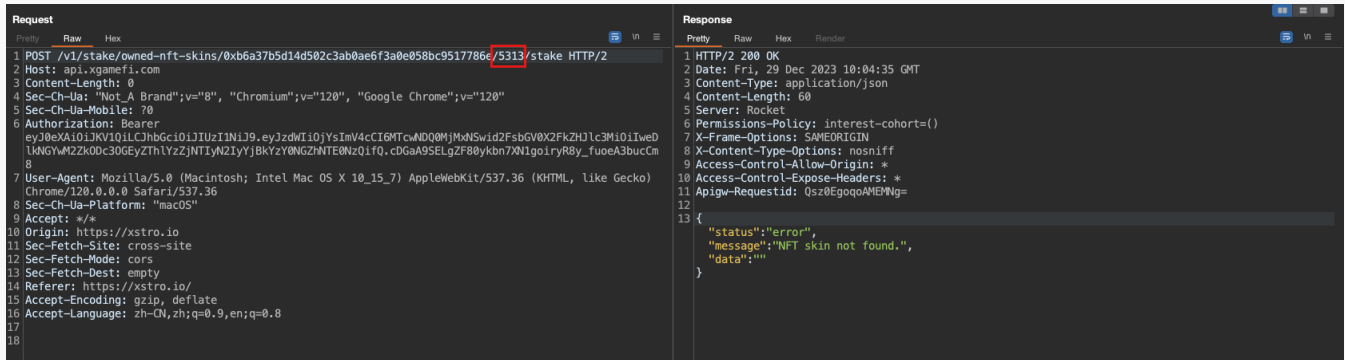
**Category: Web API security audit**

## Content

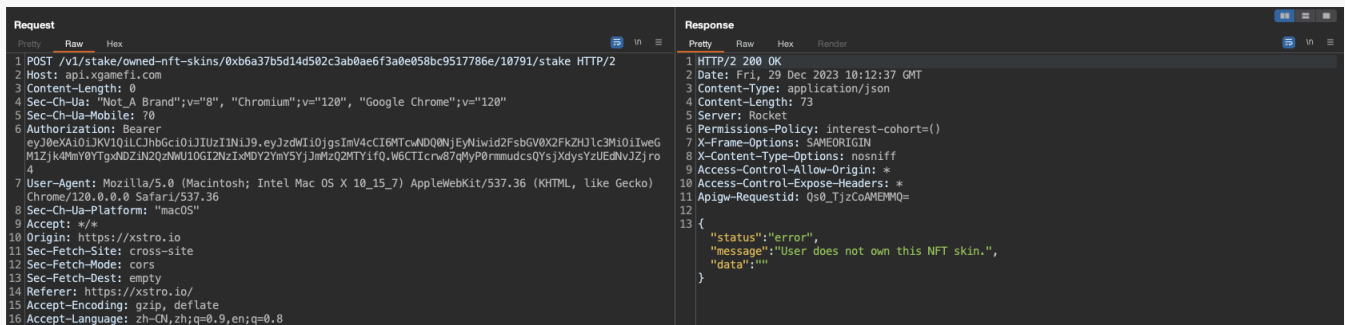
The signature in this case is merely a user confirmation on the frontend, and the server does not perform any signature verification.

Attempting to replace the NFT ID will not allow unauthorized binding.

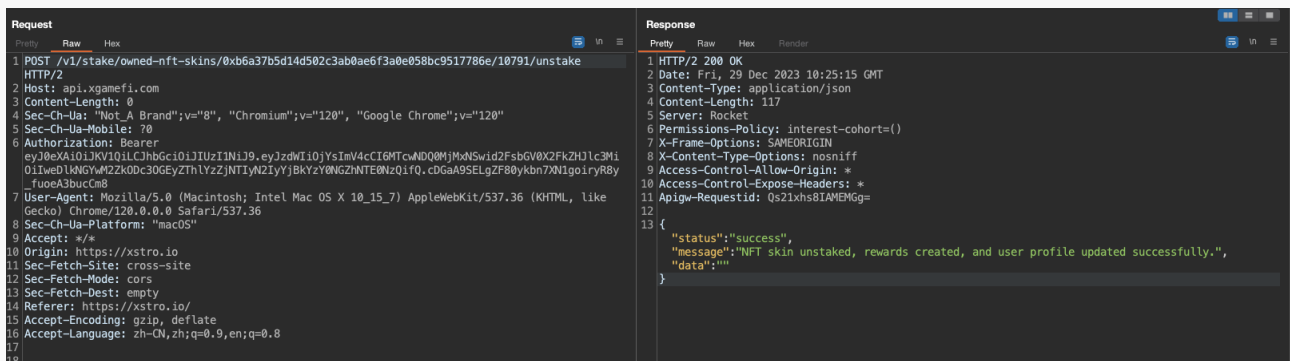




The server will determine whether a wallet address possesses a specific NFT based on the JWT token.



Similarly, the "unstake" operation does not actually verify the signature data on the server side.



## Solution

It is recommended to include the user's actual signature data in the "stake" and "unstake" operations and perform verification on the server side before proceeding.

## Status

Fixed

# 4 Audit Result

Audit Number	Audit Team	Audit Date	Audit Result
0X002312290003	SlowMist Security Team	2023.12.27 - 2023.12.29	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 4 suggestions and 1 low vulnerability. And 1 suggestion was Acknowledged. The other findings were fixed. We extend our gratitude for Xstro 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](http://www.slowmist.com)



**E-mail**  
[team@slowmist.com](mailto:team@slowmist.com)



**Twitter**  
[@SlowMist\\_Team](https://twitter.com/SlowMist_Team)



**Github**  
<https://github.com/slowmist>