Smart Contract Security Audit Report

Table Of Contents

1 Executive Summary

On 2025.03.04, the SlowMist security team received the AWE Network team's security audit application for AWE Token Bridge, 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 "white box lead, black, grey 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 DeFi project, and it is strongly recommended to fix the critical vulnerabilities.
High	High severity vulnerabilities will affect the normal operation of the DeFi project. It is strongly recommended to fix high-risk vulnerabilities.
Medium	Medium severity vulnerability will affect the operation of the DeFi project. It is recommended to fix medium-risk vulnerabilities.
Low	Low severity vulnerabilities may affect the operation of the DeFi 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 smart contract includes two steps:

- Smart contract codes are scanned/tested for commonly known and more specific vulnerabilities using automated analysis tools.
- Manual audit of the codes for security issues. The contracts are manually analyzed to look for any potential problems.

Following is the list of commonly known vulnerabilities that was considered during the audit of the smart contract:

Serial Number	Audit Class	Audit Subclass
1	Overflow Audit	-
2	Reentrancy Attack Audit	-
3	Replay Attack Audit	-
4	Flashloan Attack Audit	-
5	Race Conditions Audit	Reordering Attack Audit
6	Permission Vulnerability Audit	Access Control Audit
		Excessive Authority Audit
7	Security Design Audit	External Module Safe Use Audit
		Compiler Version Security Audit
		Hard-coded Address Security Audit
		Fallback Function Safe Use Audit
		Show Coding Security Audit
		Function Return Value Security Audit
		External Call Function Security Audit

Serial Number	Audit Class	Audit Subclass
7	Security Design Audit	Block data Dependence Security Audit
		tx.origin Authentication Security Audit
8	Denial of Service Audit	-
9	Gas Optimization Audit	-
10	Design Logic Audit	-
11	Variable Coverage Vulnerability Audit	-
12	"False Top-up" Vulnerability Audit	-
13	Scoping and Declarations Audit	-
14	Malicious Event Log Audit	-
15	Arithmetic Accuracy Deviation Audit	-
16	Uninitialized Storage Pointer Audit	-

3 Project Overview

3.1 Project Introduction

This is the AWE Token Bridge project that enables cross-chain circulation of STPT and AWE between the Ethereum and Base networks. Users can deposit STPT on the Ethereum chain and claim an equivalent amount on the Base chain after the owner batches and adds claimers, ensuring seamless asset movement across multiple blockchains. This mechanism is supported by the STPTDeposit and AWEClaim contracts.

3.2 Vulnerability Information

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

NO	Title	Category	Level	Status
N1	Centralized Cross-Chain	Authority Control Vulnerability Audit	Information	Acknowledged
N2	Low-level call reminder	Unsafe External Call Audit	Suggestion	Acknowledged

4 Code Overview

4.1 Contracts Description

The main network address of the contract is as follows:

https://etherscan.io/address/0x80bf0c6ba9a4f32fea71d1848bdc0b6e6b34fd06

https://basescan.org/address/0x80bf0c6ba9a4f32fea71d1848bdc0b6e6b34fd06

4.2 Visibility Description

The SlowMist Security team analyzed the visibility of major contracts during the audit, the result as follows:

STPTDeposit			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Public	Can Modify State	Ownable
deposit	External	Can Modify State	-
withdraw	External	Can Modify State	onlyOwner
<Receive Ether>	External	Payable	-

AWEClaim			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Public	Can Modify State	Ownable
addClaim	External	Can Modify State	onlyOwner

AWEClaim			
claim	External	Can Modify State	-
getTotalUnclaimed	External	-	-
withdraw	External	Can Modify State	onlyOwner
<Receive Ether>	External	Payable	-

4.3 Vulnerability Summary

[N1] [Information] Centralized Cross-Chain

Category: Authority Control Vulnerability Audit

Content

1.In the STPTDeposit contract, users can deposit STPT but cannot withdraw. Only the owner has the authority to call the withdraw function, allowing them to transfer all deposited STPT tokens and Ether from the contract to their own address.

2.In the AWEClaim contract, users must wait for the owner to distribute tokens before claiming their AWE. Additionally, only the owner has the authority to call the withdraw function at any time, enabling them to transfer all AWE tokens and Ether from the contract to their own address.

Solution

It is recommended that in the early stages of the project, the core role like the owner should use multi-signatures and the time-lock contract to avoid single-point risks. After the project is running stably, the authority of the core role should be handed over to community governance for management.

Status

Acknowledged

[N2] [Suggestion] Low-level call reminder

Category: Unsafe External Call Audit

Content

In the STPTDeposit and AWEClaim contract, the contract uses low-level calls and does not limit the amount of gas used to transfer native tokens.

Code location:

STPTDeposit/contracts/STPTDeposit.sol#L52-55

```
        if (ethAmount > 0) {
            (bool success, ) = owner().call{value: ethAmount}("");
            require(success, "Ether transfer failed");
        }
```

AWEClaim/contracts/AWEClaim.sol#L82-85

```
        if (ethAmount > 0) {
            (bool success, ) = owner().call{value: ethAmount}("");
            require(success, "Ether transfer failed");
        }
```

Solution

When using low-level calls, it is recommended to limit the amount of gas used.

Status

Acknowledged

5 Audit Result

Audit Number	Audit Team	Audit Date	Audit Result
0X002503050001	SlowMist Security Team	2025.03.04 - 2025.03.05	Passed

Summary conclusion: The SlowMist security team uses a manual and SlowMist team's analysis tool to audit the project, during the audit work, we found 1 suggestion, and 1 information. All the findings were acknowledged. The code has deployed to the mainnet.

6 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.

SLOWMIST

Official Website

www.slowmist.com

✉

E-mail

team@slowmist.com

Twitter

@SlowMist_Team

Github

https://github.com/slowmist