Smart Contract Security Audit Report

Table Of Contents

1 Executive Summary

On 2025.01.13, the SlowMist security team received the StakeStone team's security audit application for Story Pre Deposit Vault, 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 Story Pre Deposit Vault module of the StakeStone protocol, which will be deployed on the Ethereum mainnet. Users can deposit USDC or USDT tokens into the Story Pre Deposit Vault contract to receive earlyReceipt tokens.

3.2 Vulnerability Information

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

NO	Title	Category	Level	Status
N1	Description of deposit features	Others	Information	Acknowledged
N2	Risks of excessive privilege	Authority Control Vulnerability Audit	Medium	Acknowledged

4 Code Overview

4.1 Contracts Description

Audit Version:

https://github.com/stakestone/story-pre-deposit-vault

commit: 5ab2c94f22b66de7fbf212b90c8b8b2dfcf746f2

Audit Scope:

```
./src
├── Errors.sol
├── StoryPreDepositVault.sol
└── Token.sol
```

The main network address of the contract is as follows:

The code was not deployed to the mainnet.

4.2 Visibility Description

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

StoryPreDepositVault			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Public	Can Modify State	-
deposit	External	Can Modify State	-

StoryPreDepositVault			
setCap	External	Can Modify State	onlyRole
setMinDeposit	External	Can Modify State	onlyRole
setMaxDeposit	External	Can Modify State	onlyRole
setDepositPause	External	Can Modify State	onlyRole
withdrawTokens	External	Can Modify State	onlyRole
getSupportedTokens	External	-	-
getRate	Public	-	-

Token			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Public	Can Modify State	ERC20
mint	External	Can Modify State	onlyRole
burn	External	Can Modify State	onlyRole

4.3 Vulnerability Summary

[N1] [Information] Description of deposit features

Category: Others

Content

In the StoryPreDepositVault contract, users can convert USDT or USDC tokens into earlyReceipt tokens using the deposit function. The StoryPreDepositVault contract will only be deployed on the ETH mainnet, which will prevent deposit errors caused by token decimal conflicts.

Code location: src/StoryPreDepositVault.sol#L120-L149

```
    function deposit(
        address _token,
        uint256 _amount,
```

```
        address _receiver
    ) external {
        uint256 afterDeposit = _amount + depositedAmount[msg.sender];
        uint256 afterDepositTotal = totalDeposit + _amount;

        ...
    }
```

Solution

N/A

Status

Acknowledged

[N2] [Medium] Risks of excessive privilege

Category: Authority Control Vulnerability Audit

Content

In the StoryPreDepositVault's earlyReceipt contract, the MINTER_ROLE can arbitrarily mint earlyReceipt tokens to any specified address, and the BURNER_ROLE can burn earlyReceipt tokens from any address. The DEFAULT_ADMIN_ROLE can grant MINTER_ROLE or BURNER_ROLE to any address, which poses a risk of excessive privileges for privileged roles.

Code location: src/Token.sol#L23-L32

```
    function mint(address _to, uint256 _amount) external onlyRole(MINTER_ROLE) {
        _mint(_to, _amount);
    }

    function burn(
        address _from,
        uint256 _amount
    ) external onlyRole(BURNER_ROLE) {
        _burn(_from, _amount);
    }
```

Solution

In the short term, managing privileged roles through multi-signature wallets can effectively mitigate a single point of risk. In the long term, transferring privileged roles to DAO governance can effectively address the risk of excessive

privileges. During the transition period, management through multi-signature wallets combined with timelock-delayed transaction execution can effectively mitigate the risk of excessive privileges.

Status

Acknowledged

5 Audit Result

Audit Number	Audit Team	Audit Date	Audit Result
0X002501130001	SlowMist Security Team	2025.01.13 - 2025.01.13	Medium Risk

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 medium risk vulnerability, and 1 information. All the findings were acknowledged. The code was not deployed to the mainnet. Since the risk of excessive privileges has not yet been resolved, the final conclusion remains as medium risk.

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.

Official Website

www.slowmist.com

E-mail

team@slowmist.com

Twitter

@SlowMist_Team

Github

https://github.com/slowmist