Smart Contract Security Audit Report

Table Of Contents

1 Executive Summary

On 2022.02.22, the SlowMist security team received the team's security audit application for Arowana, 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.

Level	Description
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:

- Reentrancy Vulnerability
- Replay Vulnerability
- Reordering Vulnerability
- Short Address Vulnerability
- Denial of Service Vulnerability
- Transaction Ordering Dependence Vulnerability
- Race Conditions Vulnerability
- Authority Control Vulnerability
- Integer Overflow and Underflow Vulnerability
- TimeStamp Dependence Vulnerability
- Uninitialized Storage Pointers Vulnerability
- Arithmetic Accuracy Deviation Vulnerability
- tx.origin Authentication Vulnerability

- "False top-up" Vulnerability
- Variable Coverage Vulnerability
- Gas Optimization Audit
- Malicious Event Log Audit
- Redundant Fallback Function Audit
- Unsafe External Call Audit
- Explicit Visibility of Functions State Variables Audit
- Design Logic Audit
- Scoping and Declarations Audit

3 Project Overview

3.1 Project Introduction

Audit Version:

582ce12a822921274fe610d69322ab260236ab03

Fixed Version:

d7e9934b5677e7899b5b4f85f134029e9e05579c

3.2 Vulnerability Information

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

NO	Title	Category	Level	Status
N1	Missing event log	Others	Suggestion	Fixed
N2	Missing event log	Others	Suggestion	Fixed

NO	Title	Category	Level	Status
N3	Missing balance change	Design Logic Audit	Low	Fixed
N4	Missing balance check	Design Logic Audit	Medium	Fixed

4 Code Overview

4.1 Contracts Description

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:

ERC20TokenList			
Function Name	Visibility	Mutability	Modifiers
contains	Public	-	-
addToken	Public	Can Modify State	onlyOwner
removeToken	Public	Can Modify State	onlyOwner
getAddressList	Public	-	-

StakePool			
Function Name	Visibility	Mutability	Modifiers

StakePool			
<Constructor>	Public	Can Modify State	-
poolLength	External	-	-
setValut	Public	Can Modify State	onlyOwner
setFeeTo	Public	Can Modify State	onlyOwner
setFeeRate	Public	Can Modify State	onlyOwner
add	Public	Can Modify State	onlyOwner
set	Public	Can Modify State	onlyOwner
deposit	Public	Can Modify State	-
withdraw	Public	Can Modify State	-

4.3 Vulnerability Summary

[N1] [Suggestion] Missing event log

Category: Others

Content

The owner role can call the `addToken` and `removeToken` functions to add and remove the specified token address into the `_addresses` and `_indexes`. If there is no event record, it is not conducive to the review of community users.

Code location: stake-contract/contracts/ERC20TokenList.sol#L36-47

```
    function addToken(address addr) public onlyOwner {
        //console.log("address = %s",addr);
        //console.log("contains = %s",contains(addr));

        require(addr != address(0),"TokenList/address_is_0");
        require(!contains(addr),"TokenList/address_already_exist");
        require(addr.isContract(),"TokenList/address_is_not_contract");

        _addresses.push(addr);
        _indexes[addr] = _addresses.length;
    }
```

Code location: stake-contract/contracts/ERC20TokenList.sol#L60-73

```
    function removeToken(address addr) public  onlyOwner {
        require(contains(addr),"TokenList/address_is_not_exist");
        uint256 idx = _indexes[addr];
        uint256 toDeleteIndex = idx - 1;
        uint256 lastIndex = _addresses.length - 1;

        address lastAddress = _addresses[lastIndex];

        _addresses[toDeleteIndex] = lastAddress;
        _indexes[lastAddress] = toDeleteIndex + 1;

        _addresses.pop();
        delete _indexes[addr];
    }
```

Solution

It is recommended to add related event log.

Status

Fixed

[N2] [Suggestion] Missing event log

Category: Others

Content

The owner role can call the `setValut` function to set the source of reward token distribution. If there is no event record, it is not conducive to the review of community users.

Code location: stake-contract/contracts/StakePool.sol#L68-70

```
function setValut(address _valut) public onlyOwner {
    valut = _valut;
}
```

The owner role can call the `setFeeTo` function to set the address to receive fee. If there is no event record, it is not conducive to the review of community users.

Code location: stake-contract/contracts/StakePool.sol#L79-81

```
function setFeeTo(address _feeTo) public onlyOwner {
    feeTo = _feeTo;
}
```

The owner role can call the `setFeeRate` function to set the feeRate. If there is no event record, it is not conducive to the review of community users.

Code location: stake-contract/contracts/StakePool.sol#L91-94

```
function setFeeRate(uint256 _feeRate) public onlyOwner {
    require( _feeRate <= 100,"StakePool/setFeeRate__feeRate <= 100");
    feeRate = _feeRate;
}
```

Solution

It is recommended to add related event log.

Status

Fixed

[N3] [Low] Missing balance change

Category: Design Logic Audit

Content

When the user calls the `withdraw` function for withdrawal (block.timestap >= pool.end), the balance of pool is not changed here.

Code location: stake-contract/contracts/StakePool.sol#L199-222

```
    function withdraw(uint256 _pid, uint256 _amount) public {
        PoolInfo storage pool = poolInfo[_pid];
        UserInfo storage user = userInfo[_pid][msg.sender];
        require(user.amount >= _amount, "StakePool/withdraw_amount_fall_short");
        require(block.timestamp  < pool.start - 1 days || pool.end < block.timestamp
,"StakePool/withdraw_time_error");

        IERC20 erc20 = IERC20(poolInfo[_pid].erc20);

        if(pool.end > block.timestamp){ //종료 전
            erc20.safeTransfer(address(msg.sender), _amount);
            user.amount = user.amount.sub(_amount);
            user.reward = user.amount.mul(pool.apr).div(100);
            pool.amount = pool.amount.sub(_amount);
        } else { //종료 후
            erc20.safeTransfer(address(msg.sender), user.amount);
            uint256 fee = user.reward.mul(feeRate).div(100);
            erc20.safeTransferFrom(address(valut), address(msg.sender),
user.reward.sub(fee));
            erc20.safeTransferFrom(address(valut), address(feeTo), fee);
            user.amount = 0;
            user.reward = 0;
        }

        emit Withdraw(msg.sender, _pid, _amount);
    }
```

Solution

When the user calls the `withdraw` function for withdrawal (block.timestap >= pool.end), the change to the balance of the pool should be added. Like this:

`pool.amount = pool.amount.sub(user.amount);`

Status

Fixed

[N4] [Medium] Missing balance check

Category: Design Logic Audit

Content

When the user calls the `withdraw` function for withdrawal (block.timestap >= pool.end), there is a lack of judgment on the balance of valut. If the token balance of valut is not enough to pay the user's reward, the transaction will be rolled back and the user's principal and reward cannot be withdrawn.

Code location: stake-contract/contracts/StakePool.sol#L199-222

```
    function withdraw(uint256 _pid, uint256 _amount) public {
        PoolInfo storage pool = poolInfo[_pid];
        UserInfo storage user = userInfo[_pid][msg.sender];
        require(user.amount >= _amount, "StakePool/withdraw_amount_fall_short");
        require(block.timestamp  < pool.start - 1 days || pool.end < block.timestamp
,"StakePool/withdraw_time_error");

        IERC20 erc20 = IERC20(poolInfo[_pid].erc20);

        if(pool.end > block.timestamp){ //종료 전
            erc20.safeTransfer(address(msg.sender), _amount);
            user.amount = user.amount.sub(_amount);
            user.reward = user.amount.mul(pool.apr).div(100);
            pool.amount = pool.amount.sub(_amount);
        } else { //종료 후
            erc20.safeTransfer(address(msg.sender), user.amount);
            uint256 fee = user.reward.mul(feeRate).div(100);
            erc20.safeTransferFrom(address(valut), address(msg.sender),
user.reward.sub(fee));
            erc20.safeTransferFrom(address(valut), address(feeTo), fee);
            user.amount = 0;
            user.reward = 0;
        }

        emit Withdraw(msg.sender, _pid, _amount);
    }
```

Solution

Increase the judgment of valut balance. If the balance is enough to pay the user's reward, transfer the principal and reward to the user. But if the balance is not enough to pay the user's reward, only transfer the principal to the user.

Status

Fixed

5 Audit Result

Audit Number	Audit Team	Audit Date	Audit Result
0X002202240001	SlowMist Security Team	2022.02.22 - 2022.02.24	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 1 medium risk, 1 low risk, 2 suggestion vulnerabilities. And all findings were fixed. The code was not 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