Smart Contract Security Audit Report

[2021]

Table Of Contents

1 Executive Summary

On 2021.05.26, the SlowMist security team received the Cook Finance team's security audit application for Cook Distribution and Reward, 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 party 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 Aduit
- Design Logic Audit
- Scoping and Declarations Audit

3 Project Overview

3.1 Project Introduction

Audit Information:

Project address:

https://github.com/CookFinance/cook-distribution-and-reward/tree/main/contracts/farm

commit: 4291bd2fe8ee18577a5e824a5763cbfe10bc8444

Fix address:

https://github.com/CookFinance/cook-distribution-and-reward/tree/main/contracts/farm

commit:1bf5704c2d6d7cb679b12899b85c047f33fc6b33

3.2 Vulnerability Information

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

NO	Title	Category	Level	Status
N1	Re-initialize issue	Others	Suggestion	Fixed
N2	Missing authority check	Others	Suggestion	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:

StakingPools			
Function Name	**Visibility**	**Mutability**	**Modifiers**
	Public	can modify state	-
setPendingGovernance	External	can modify state	onlyGovernance
acceptGovernance	External	can modify state	-
setRewardRate	External	can modify state	onlyGovernance
createPool	External	can modify state	onlyGovernance
setRewardWeights	External	can modify state	onlyGovernance
deposit	External	can modify state	nonReentrant checkIfNewReferral

StakingPools			
withdraw	External	can modify state	nonReentrant
claim	External	can modify state	nonReentrant
exit	External	can modify state	nonReentrant
rewardRate	External	-	-
totalRewardWeight	External	-	-
poolCount	External	-	-
getPoolToken	External	-	-
getPoolTotalDeposited	External	-	-
getPoolTotalReferralAmount	External		-
getPoolRewardWeight	External	-	-
getPoolRewardRate	External	-	-
getStakeTotalDeposited	External	-	-
getStakeTotalUnclaimed	External	-	-
getAccumulatedReferralPower	External	-	-
getPoolReferral	External	-	-
getPoolreferee	External	-	-
_updatePools	Internal	can modify state	-
_deposit	Internal	can modify state	-
_withdraw	Internal	can modify state	-
_claim	Internal	can modify state	-

StakingPools			
setRewardVesting	External	can modify state	-
setSentinel	External	can modify state	onlyGovernance
setPause	External	can modify state	-
startReferralBonus	External	can modify state	-
stoptReferralBonus	External	can modify state	-

4.3 Vulnerability Summary

[N1] [Suggestion] Re-initialize issue

Category: Others

Content

In RewardVesting contract, the Governance role can re-initialize the the contract through initialize function

Location：

```
function initialize(IERC20 _cookReward) external onlyGovernance {

    cookReward = _cookReward;
}
```

Solution

It is recommended to prohibit repeated initialization operations and use a separate interface to set related parameters.

Status

Fixed

[N2] [Suggestion] Missing authority check

Category: Others

Content

The addEarning function exists in the RewardVesting contract. When the claim operation is performed in the StakingPool contract, if the corresponding reward needs to be time locked, the addEarning function of the RewardVesting contract will be called to perform the locking operation. However, the visibility of this function is external, which will cause any user to perform the addEarning operation.

代码位置:

```
    function addEarning(address user, uint256 amount, uint256 durationInSecs)
external {
        _addPendingEarning(user, amount, durationInSecs);
        cookReward.safeTransferFrom(msg.sender, address(this), amount);
    }
```

Solution

It is recommended to restrict the addEarning function to only StakingPool contract calls.

Status

Fixed

5 Audit Result

Audit Number	Audit Team	Audit Date	Audit Result
0X002106020002	SlowMist Security Team	2021.05.26 - 2021.06.02	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 suggestions. 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.

Official Website

www.slowmist.com

E-mail

team@slowmist.com

Twitter

@SlowMist_Team

Github

https://github.com/slowmist