Smart Contract Security Audit Report

[2021]

Table Of Contents

1 Executive Summary

On 2021.12.01, the SlowMist security team received the Otter Finance team's security audit application for Otter Finance, 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
- Unsafe External Call Audit
- Design Logic Audit
- Scoping and Declarations Audit

3 Project Overview

3.1 Project Introduction

Project official website:

https://app.otterclam.finance/

3.2 Vulnerability Information

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

NO	Title	Category	Level	Status
N1	Risk of excessive authority	Authority Control Vulnerability	Suggestion	Confirmed
N2	Event log missing	Others	Suggestion	Ignored

4 Code Overview

4.1 Contracts Description

The main network address of the contract is as follows:

Bond Calculator: 0x47655e27667E5B4EC9EB70799f281524d031381c

MAI bond: 0x28077992bFA9609Ae27458A766470b03D43dEe8A

MAI/CLAM LP Bond v1: 0x79B47c03B02019Af78Ee0de9B0b3Ac0786338a0d

MAI/CLAM LP Bond v2: 0x64c766f9A4936c3a4b51C55Ea5C4854E19766035

CLAM: 0xC250e9987A032ACAC293d838726C511E6E1C029d

sCLAM: 0xAAc144Dc08cE39Ed92182dd85ded60E5000C9e67

Treasury: 0x8ce47D56EAa1299d3e06FF3E04637449fFb01C9C

Staking: 0xC8B0243F350AA5F8B979b228fAe522DAFC61221a

Staking Helper: 0x76B38319483b570B4BCFeD2D35d191d3c9E01691

Staking Warm Up: 0x8b2943667957ec2ce851fd449b7a870f253ca1e7

Staking Distributor: 0x0Dd015889df6F50d39e9D7A52711D0B86E43FC62

Audit version:

https://github.com/OtterClam/otter-contracts

f0eaa7433898d4aa8f2265de298b7aa8cb50c045

4.2 Visibility Description

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

OtterBondDepository			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Public	Can Modify State	-
initializeBondTerms	External	Can Modify State	onlyOwner
setBondTerms	External	Can Modify State	onlyOwner
setAdjustment	External	Can Modify State	onlyOwner
setStaking	External	Can Modify State	onlyOwner
deposit	External	Can Modify State	-
redeem	External	Can Modify State	-
stakeOrSend	Internal	Can Modify State	-
adjust	Internal	Can Modify State	-

OtterBondDepository			
decayDebt	Internal	Can Modify State	-
maxPayout	Public	-	-
payoutFor	Public	-	-
bondPrice	Public	-	-
_bondPrice	Internal	Can Modify State	-
bondPriceInUSD	Public	-	-
debtRatio	Public	-	-
standardizedDebtRatio	External	-	-
currentDebt	Public	-	-
debtDecay	Public	-	-
percentVestedFor	Public	-	-
pendingPayoutFor	External	-	-
recoverLostToken	External	Can Modify State	-

OtterBondingCalculator			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Public	Can Modify State	-
getKValue	Public	-	-
getTotalValue	Public	-	-
valuation	External	-	-

OtterBondingCalculator			
markdown	External	-	-

OtterBondStakeDepository			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Public	Can Modify State	-
initializeBondTerms	External	Can Modify State	onlyOwner
setBondTerms	External	Can Modify State	onlyOwner
setAdjustment	External	Can Modify State	onlyOwner
setStaking	External	Can Modify State	onlyOwner
deposit	External	Can Modify State	-
redeem	External	Can Modify State	-
adjust	Internal	Can Modify State	-
decayDebt	Internal	Can Modify State	-
maxPayout	Public	-	-
payoutFor	Public	-	-
bondPrice	Public	-	-
_bondPrice	Internal	Can Modify State	-
bondPriceInUSD	Public	-	-
debtRatio	Public	-	-
standardizedDebtRatio	External	-	-

OtterBondStakeDepository			
currentDebt	Public	-	-
debtDecay	Public	-	-
percentVestedFor	Public	-	-
pendingPayoutFor	External	-	-
recoverLostToken	External	Can Modify State	-

OtterClamERC20			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Public	Can Modify State	ERC20
mint	External	Can Modify State	onlyVault
burn	Public	Can Modify State	-
burnFrom	Public	Can Modify State	-
_burnFrom	Public	Can Modify State	-

OtterClamERC20V2			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Public	Can Modify State	ERC20
completeMigration	External	Can Modify State	onlyOwner
mint	External	Can Modify State	onlyVault
burn	Public	Can Modify State	-
burnFrom	Public	Can Modify State	-

OtterClamERC20V2			
_burnFrom	Public	Can Modify State	-

OtterStaking			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Public	Can Modify State	-
stake	External	Can Modify State	-
claim	External	Can Modify State	-
forfeit	External	Can Modify State	-
toggleDepositLock	External	Can Modify State	-
unstake	External	Can Modify State	-
index	Public	-	-
rebase	Public	Can Modify State	-
contractBalance	Public	-	-
giveLockBonus	External	Can Modify State	-
returnLockBonus	External	Can Modify State	-
setContract	External	Can Modify State	onlyOwner
setWarmup	External	Can Modify State	onlyOwner

OtterStakingDistributor			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Public	Can Modify State	-

OtterStakingDistributor			
distribute	External	Can Modify State	-
adjust	Internal	Can Modify State	-
nextRewardAt	Public	-	-
nextRewardFor	Public	-	-
addRecipient	External	Can Modify State	onlyOwner
removeRecipient	External	Can Modify State	onlyOwner
setAdjustment	External	Can Modify State	onlyOwner

OtterStakingHelper			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Public	Can Modify State	-
stake	External	Can Modify State	-

OtterStakingWarmup			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Public	Can Modify State	-
retrieve	External	Can Modify State	-

OtterTreasury			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Public	Can Modify State	-

OtterTreasury			
deposit	External	Can Modify State	-
withdraw	External	Can Modify State	-
incurDebt	External	Can Modify State	-
repayDebtWithReserve	External	Can Modify State	-
repayDebtWithCLAM	External	Can Modify State	-
manage	External	Can Modify State	-
mintRewards	External	Can Modify State	-
excessReserves	Public	-	-
auditReserves	External	Can Modify State	onlyOwner
valueOfToken	Public	-	-
queue	External	Can Modify State	onlyOwner
toggle	External	Can Modify State	onlyOwner
requirements	Internal	-	-
listContains	Internal	-	-

StakedOtterClamERC20			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Public	Can Modify State	ERC20 ERC20Permit
initialize	External	Can Modify State	-
setIndex	External	Can Modify State	onlyOwner

StakedOtterClamERC20			
rebase	Public	Can Modify State	onlyStakingContract
_storeRebase	Internal	Can Modify State	-
balanceOf	Public	-	-
gonsForBalance	Public	-	-
balanceForGons	Public	-	-
circulatingSupply	Public	-	-
index	Public	-	-
transfer	Public	Can Modify State	-
allowance	Public	-	-
transferFrom	Public	Can Modify State	-
approve	Public	Can Modify State	-
_approve	Internal	Can Modify State	-
increaseAllowance	Public	Can Modify State	-
decreaseAllowance	Public	Can Modify State	-

StakedOtterClamERC20V2			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Public	Can Modify State	ERC20 ERC20Permit
completeMigration	External	Can Modify State	onlyOwner
initialize	External	Can Modify State	-

StakedOtterClamERC20V2			
setIndex	External	Can Modify State	onlyOwner
rebase	Public	Can Modify State	onlyStakingContract
_storeRebase	Internal	Can Modify State	-
balanceOf	Public	-	-
gonsForBalance	Public	-	-
balanceForGons	Public	-	-
circulatingSupply	Public	-	-
index	Public	-	-
transfer	Public	Can Modify State	-
allowance	Public	-	-
transferFrom	Public	Can Modify State	-
approve	Public	Can Modify State	-
_approve	Internal	Can Modify State	-
increaseAllowance	Public	Can Modify State	-
decreaseAllowance	Public	Can Modify State	-

OlympusBondDepository			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Public	Can Modify State	-
initializeBondTerms	External	Can Modify State	onlyOwner

OlympusBondDepository			
setBondTerms	External	Can Modify State	onlyOwner
setAdjustment	External	Can Modify State	onlyOwner
setStaking	External	Can Modify State	onlyOwner
deposit	External	Can Modify State	-
redeem	External	Can Modify State	-
stakeOrSend	Internal	Can Modify State	-
adjust	Internal	Can Modify State	-
decayDebt	Internal	Can Modify State	-
maxPayout	Public	-	-
payoutFor	Public	-	-
bondPrice	Public	-	-
_bondPrice	Internal	Can Modify State	-
assetPrice	Public	-	-
bondPriceInUSD	Public	-	-
debtRatio	Public	-	-
standardizedDebtRatio	External	-	-
currentDebt	Public	-	-
debtDecay	Public	-	-
percentVestedFor	Public	-	-
pendingPayoutFor	External	-	-

OlympusBondDepository			
recoverLostToken	External	Can Modify State	-

4.3 Vulnerability Summary

[N1] [Suggestion] Risk of excessive authority

Category: Authority Control Vulnerability

Content

- `contracts/OtterStaking.sol`

`giveLockBonus` If there is a problem with the multi-sign wallet, you can set the locker role. You can remove the `sCLAM` from the pool and replace it with `Clam` through the `unstake` function. `Clam` can go to the `OtterTreasury` contract and use the `withdraw` to remove the above tokens.

Solution

It is recommended to add timelock to the `locker` permission setting like `OtterTreasury`

Status

Confirmed; The locker permission can only be set once. The project party gives up the use of the permission by setting the `locker` address to `0x000000000000000000000000000000000000dead`

[N2] [Suggestion] Event log missing

Category: Others

Content

- `contracts/OtterStaking.sol`

  Missing functions for key event records:

```
function giveLockBonus(uint _amount) external {
    require(msg.sender == locker);
```

```
    totalBonus = totalBonus.add(_amount);
    IERC20(sCLAM).safeTransfer(locker, _amount);
}

function returnLockBonus(uint _amount) external {
    require(msg.sender == locker);
    totalBonus = totalBonus.sub(_amount);
    IERC20(sCLAM).safeTransferFrom(locker, address(this), _amount);
}

function setContract(CONTRACTS _contract, address _address) external onlyOwner() {
    if (_contract == CONTRACTS.DISTRIBUTOR) { // 0
        distributor = _address;
    } else if (_contract == CONTRACTS.WARMUP) { // 1
        require(warmupContract == address(0), "Warmup cannot be set more than once");
        warmupContract = _address;
    } else if (_contract == CONTRACTS.LOCKER) { // 2
        require(locker == address(0), "Locker cannot be set more than once");
        locker = _address;
    }
}

function setWarmup(uint _warmupPeriod) external onlyOwner() {
    warmupPeriod = _warmupPeriod;
}
```

Solution

It is recommended to record incidents when modifying sensitive parameters of the contract for follow-up self-examination or community review.

Status

Ignored

5 Audit Result

Audit Number	Audit Team	Audit Date	Audit Result

Audit Number	Audit Team	Audit Date	Audit Result
0X002112140001	SlowMist Security Team	2021.12.01 - 2021.12.14	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 suggestion vulnerabilities.

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