SLOWMIST

Smart Contract Security Audit Report

[2021]

Table Of Contents

1 Executive Summary

On 2021.05.06, the SlowMist security team received the PancakeSwap team's security audit application for PancakeSwap, 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 contracts

PancakeSwap Factory:

https://bscscan.com/address/0xcA143Ce32Fe78f1f7019d7d551a6402fC5350c73#code

PancakeSwap Router:

https://bscscan.com/address/0x10ED43C718714eb63d5aA57B78B54704E256024E#code

3.2 Vulnerability Information

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

NO	Title	Category	Level	Status
N1	Missing Checking	Others	Suggestion	Confirmed

4 Code Overview

4.1 Contracts Description

The main network address of the contract is as follows:

PancakeSwap Factory:

https://bscscan.com/address/0xcA143Ce32Fe78f1f7019d7d551a6402fC5350c73#code

PancakeSwap Router:

https://bscscan.com/address/0x10ED43C718714eb63d5aA57B78B54704E256024E#code

4.2 Visibility Description

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

PancakeRouter			
Function Name	Visibility	Mutability	Modifiers
removeLiquidityETHSupportingFeeOnTransferTokens	external	can modify state	-
removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	external	can modify state	-
swapExactTokensForTokensSupportingFeeOnTransferTokens	external	can modify state	-
swapExactETHForTokensSupportingFeeOnTransferTokens	external	payable	-
swapExactTokensForETHSupportingFeeOnTransferTokens	external	can modify state	-
factory	external	-	-
WETH	external	-	-
addLiquidity	external	can modify state	-

PancakeRouter			
addLiquidityETH	external	payable	-
removeLiquidity	external	can modify state	-
removeLiquidityETH	external	can modify state	-
removeLiquidityWithPermit	external	can modify state	-
removeLiquidityETHWithPermit	external	can modify state	-
swapExactTokensForTokens	external	can modify state	-
swapTokensForExactTokens	external	can modify state	-
swapExactETHForTokens	external	payable	-
swapTokensForExactETH	external	can modify state	-
swapExactTokensForETH	external	can modify state	-
swapETHForExactTokens	external	payable	-
quote	external	-	-
getAmountOut	external	-	-
getAmountIn	external	-	-
getAmountsOut	external	-	-
getAmountsIn	external	-	-
constructor	public	can modify state	-
receive	external	payable	-
_addLiquidity	internal	can modify state	-
addLiquidity	external	can modify state	ensure

PancakeRouter			
addLiquidityETH	external	payable	ensure
removeLiquidity	public	can modify state	ensure
removeLiquidityETH	public	can modify state	ensure
removeLiquidityWithPermit	external	can modify state	-
removeLiquidityETHWithPermit	external	can modify state	-
removeLiquidityETHSupportingFeeOnTransferTokens	public	can modify state	ensure
removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	external	can modify state	-
_swap	internal	can modify state	-
swapExactTokensForTokens	external	can modify state	ensure
swapTokensForExactTokens	external	can modify state	ensure
swapExactETHForTokens	external	payable	ensure
swapTokensForExactETH	external	can modify state	ensure
swapExactTokensForETH	external	can modify state	ensure
swapETHForExactTokens	external	payable	ensure
_swapSupportingFeeOnTransferTokens	internal	can modify state	-
swapExactTokensForTokensSupportingFeeOnTransferTokens	external	can modify state	ensure
swapExactETHForTokensSupportingFeeOnTransferTokens	external	payable	ensure
swapExactTokensForETHSupportingFeeOnTransferTokens	external	can modify state	ensure
quote	public	-	-

PancakeRouter			
getAmountOut	public	-	-
getAmountIn	public	-	-
getAmountsOut	public	-	-
getAmountsIn	public	-	-

PancakeERC20			
Function Name	Visibility	Mutability	Modifiers
name	external	-	-
symbol	external	-	-
decimals	external	-	-
totalSupply	external	-	-
balanceOf	external	-	-
allowance	external	-	-
approve	external	can modify state	-
transfer	external	can modify state	-
transferFrom	external	can modify state	-
DOMAIN_SEPARATOR	external	-	-
PERMIT_TYPEHASH	external	-	-
nonces	external	-	-
permit	external	can modify state	-

PancakeERC20			
constructor	public	can modify state	-
_mint	internal	can modify state	-
_burn	internal	can modify state	-
_approve	private	can modify state	-
_transfer	private	can modify state	-
approve	external	can modify state	-
transfer	external	can modify state	-
transferFrom	external	can modify state	-
permit	external	can modify state	-

PancakePair			
Function Name	Visibility	Mutability	Modifiers
constructor	public	can modify state	-
_mint	internal	can modify state	-
_burn	internal	can modify state	-
_approve	private	can modify state	-
_transfer	private	can modify state	-
approve	external	can modify state	-
transfer	external	can modify state	-
transferFrom	external	can modify state	-

PancakePair			
permit	external	can modify state	-
name	external	-	-
symbol	external	-	-
decimals	external	-	-
totalSupply	external	-	-
balanceOf	external	-	-
allowance	external	-	-
approve	external	can modify state	-
transfer	external	can modify state	-
transferFrom	external	can modify state	-
DOMAIN_SEPARATOR	external	-	-
PERMIT_TYPEHASH	external	-	-
nonces	external	-	-
permit	external	can modify state	-
name	external	-	-
symbol	external	-	-
decimals	external	-	-
totalSupply	external	-	-
balanceOf	external	-	-
allowance	external	-	-

PancakePair			
approve	external	can modify state	-
transfer	external	can modify state	-
transferFrom	external	can modify state	-
DOMAIN_SEPARATOR	external	-	-
PERMIT_TYPEHASH	external	-	-
nonces	external	-	-
permit	external	can modify state	-
MINIMUM_LIQUIDITY	external	-	-
factory	external	-	-
token0	external	-	-
token1	external	-	-
getReserves	external	-	-
price0CumulativeLast	external	-	-
price1CumulativeLast	external	-	-
kLast	external	-	-
mint	external	can modify state	-
burn	external	can modify state	-
swap	external	can modify state	-
skim	external	can modify state	-
sync	external	can modify state	-

PancakePair			
initialize	external	can modify state	-
getReserves	public	-	-
_safeTransfer	private	can modify state	-
constructor	public	can modify state	-
initialize	external	can modify state	-
_update	private	can modify state	-
_mintFee	private	can modify state	-
mint	external	can modify state	lock
burn	external	can modify state	lock
swap	external	can modify state	lock
skim	external	can modify state	lock
sync	external	can modify state	lock

PancakeFactory			
Function Name	Visibility	Mutability	Modifiers
feeTo	external	-	-
feeToSetter	external	-	-
getPair	external	-	-
allPairs	external	-	-
allPairsLength	external	-	-

PancakeFactory			
createPair	external	can modify state	-
setFeeTo	external	can modify state	-
setFeeToSetter	external	can modify state	-
constructor	public	can modify state	-
allPairsLength	external	-	-
createPair	external	can modify state	-
setFeeTo	external	can modify state	-
setFeeToSetter	external	can modify state	-

4.3 Vulnerability Summary

[N1] [Suggestion] Missing Checking

Category: Others

Content

In PancakeSwapRouter contract, the removeLiquidity / removeLiquidityETH / removeLiquidityWithPermit function does not check whether a pair is exist, which will leads to gas wasting when a pair does not exist.

eg. removeLiquidity function

```
function removeLiquidity(
        address tokenA,
        address tokenB,
        uint liquidity,
        uint amountAMin,
        uint amountBMin,
        address to,
        uint deadline
    ) public virtual override ensure(deadline) returns (uint amountA, uint amountB) {
        address pair = PancakeLibrary.pairFor(factory, tokenA, tokenB);
        IPancakePair(pair).transferFrom(msg.sender, pair, liquidity); // send liquidity to pair
        (uint amount0, uint amount1) = IPancakePair(pair).burn(to);
        (address token0,) = PancakeLibrary.sortTokens(tokenA, tokenB);
        (amountA, amountB) = tokenA == token0 ? (amount0, amount1) : (amount1, amount0);
        require(amountA >= amountAMin, 'PancakeRouter: INSUFFICIENT_A_AMOUNT');
        require(amountB >= amountBMin, 'PancakeRouter: INSUFFICIENT_B_AMOUNT');
    }
```

Solution

Check if the pair is exist

Status

Confirmed; The project party confirmed that issue.

5 Audit Result

Audit Number	Audit Team	Audit Date	Audit Result
0x002105120002	SlowMist Security Team	2021.05.06 - 2021.05.12	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 enhancement suggestion. 1 enhancement suggestion were ignored; All other findings were fixed.

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