Smart Contract
Security Audit Report

Table Of Contents

1 Executive Summary

On 2022.08.22, the SlowMist security team received the PancakeSwap team's security audit application for PancakeSwap Stable Swap, 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:

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

Serial Number	Audit Class	Audit Subclass
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
		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

Audit Version:

Project address:

https://github.com/ChefSnoopy/pancake-contracts/tree/stable-swap/projects/stable-swap/contracts

commit: 9e313eae7eb6baf416b82cd938cad7c8069743c6

Fixed Version:

Project address:

https://github.com/ChefSnoopy/pancake-contracts/tree/stable-swap/projects/stable-swap/contracts

commit: e256ee5761a5452382f976089ad9411709ebe1bc

3.2 Vulnerability Information

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

NO	Title	Category	Level	Status
N1	Missing event records	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:

PancakeStableSwap			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Public	Can Modify State	-
initialize	External	Can Modify State	-
get_A	Internal	-	-
A	External	-	-
_xp	Internal	-	-
_xp_mem	Internal	-	-
get_D	Internal	-	-
get_D_mem	Internal	-	-
get_virtual_price	External	-	-
calc_token_amount	External	-	-
add_liquidity	External	Can Modify State	nonReentrant
get_y	Internal	-	-
get_dy	External	-	-
get_dy_underlying	External	-	-
exchange	External	Can Modify State	nonReentrant
remove_liquidity	External	Can Modify State	nonReentrant

PancakeStableSwap			
remove_liquidity_imbalance	External	Can Modify State	nonReentrant
get_y_D	Internal	-	-
_calc_withdraw_one_coin	Internal	-	-
calc_withdraw_one_coin	External	-	-
remove_liquidity_one_coin	External	Can Modify State	nonReentrant
ramp_A	External	Can Modify State	onlyOwner
stop_rampget_A	External	Can Modify State	onlyOwner
commit_new_fee	External	Can Modify State	onlyOwner
apply_new_fee	External	Can Modify State	onlyOwner
revert_new_parameters	External	Can Modify State	onlyOwner
admin_balances	External	-	-
withdraw_admin_fees	External	Can Modify State	onlyOwner
donate_admin_fees	External	Can Modify State	onlyOwner
kill_me	External	Can Modify State	onlyOwner
unkill_me	External	Can Modify State	onlyOwner

PancakeStableSwapLP			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Public	Can Modify State	ERC20
mint	External	Can Modify State	onlyMinter

PancakeStableSwapLP			
burnFrom	External	Can Modify State	onlyMinter

PancakeStableSwapFactory			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Public	Can Modify State	-
sortTokens	Internal	-	-
createSwapPair	External	Can Modify State	onlyOwner
getPairInfo	External	-	-

4.3 Vulnerability Summary

[N1] [Suggestion] Missing event records

Category: Others

Content

In the PancakeStableSwap contract, the owner can set the is_killed, balances and admin_actions_deadline parameters respectively through the kill_me, unkill_me, donate_admin_fees and revert_new_parameters functions, but no event recording is performed.

Code location: contracts/PancakeStableSwap.sol

```
function revert_new_parameters() external onlyOwner {
    admin_actions_deadline = 0;
}
...
function donate_admin_fees() external onlyOwner {
    for (uint256 i = 0; i < N_COINS; i++) {
        balances[i] = IERC20(coins[i]).balanceOf(address(this));
    }
```

```
}

function kill_me() external onlyOwner {
    require(kill_deadline > block.timestamp, "Exceeded deadline");
    is_killed = true;
}

function unkill_me() external onlyOwner {
    is_killed = false;
}
```

Solution

It is recommended to record events when sensitive parameters are modified for subsequent self-inspection or community review.

Status

Fixed

5 Audit Result

Audit Number	Audit Team	Audit Date	Audit Result
0X002208250001	SlowMist Security Team	2022.08.22 - 2022.08.25	Passed

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 suggestion vulnerability. All other 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