Smart Contract
Security Audit Report

Table Of Contents

1 Executive Summary

On 2022.05.23, the SlowMist security team received the team's security audit application for Umbrella Network - Gemini, 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:

https://github.com/umbrella-network/gemini

commit: 1a962208a1ab1a2e19d1fc22f5371f81bd902fe8

Fixed version:

https://github.com/umbrella-network/gemini

commit: 67ec62f18f860bc17ec7f8ce897c4d5a8a5470cd

3.2 Vulnerability Information

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

NO	Title	Category	Level	Status
N1	Missing event record	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:

DatumRegistry

DatumRegistry			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Public	Can Modify State	Registrable
sendDeliveries	External	Can Modify State	onlyOperator nonReentrant
create	External	Can Modify State	-
deposit	External	Can Modify State	-
withdraw	External	Can Modify State	-
claimFees	External	Can Modify State	-
addKeys	External	Can Modify State	-
removeKeys	External	Can Modify State	-
setDatumEnabled	External	Can Modify State	-
checkReceivers	External	-	-
getBalance	External	-	-
getName	External	-	-
setCommissionRate	Public	Can Modify State	onlyOwner
checkProofForPallet	Public	-	-
checkProofForPallets	Public	-	-
checkReceiver	Public	-	-
getManyDatums	Public	-	-
resolveId	Public	-	-
_deliverPallet	Internal	Can Modify State	-

DatumRegistry			
_deliverPallets	Internal	Can Modify State	-
_deposit	Internal	Can Modify State	-
_insertKeysToArray	Internal	Can Modify State	-
_computeFee	Internal	-	-
_resolvePrices	Internal	-	-

Registrable			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Public	Can Modify State	-
register	External	Can Modify State	-
unregister	External	Can Modify State	-
getName	External	-	-
tokenContract	Public	-	-

Operatable			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Public	Can Modify State	-
addOperator	Public	Can Modify State	onlyOwner
removeOperator	Public	Can Modify State	onlyOwner
pause	Public	Can Modify State	onlyOwner
unpause	Public	Can Modify State	onlyOwner

Operatable			
isOperator	Public	-	-

StandardDatumReceiver			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Public	Can Modify State	-
setMinTimeBetweenUpdates	External	Can Modify State	onlyOwner
receivePallet	External	Can Modify State	onlyFromDatumRegistry
approvePallet	External	-	-
getStoredRecords	External	-	-

4.3 Vulnerability Summary

[N1] [Suggestion] Missing event record

Category: Others

Content

1.In the DatumRegistry contract, the Owner role can set the commissionRate value and the commissionRate is not checked whether it is 0 from the setCommissionRate function.

Code location:

DatumRegistry.sol#L272-274

```
function setCommissionRate(uint16 newRate) public onlyOwner {
  commissionRate = newRate;
}
```

2.In the StandardDatumReceiver contract, the Owner role can set the minTimeBetweenUpdates value and the minTimeBetweenUpdates is not checked whether it is 0 from the setMinTimeBetweenUpdates function.

Code location:

StandardDatumReceiver.sol#L50-52

```
function setCommissionRate(uint16 newRate) public onlyOwner {
  commissionRate = newRate;
}
```

Solution

It is recommended to record events when sensitive parameters are modified for subsequent self-inspection or community review and check the minTimeBetweenUpdates and the commissionRate value is 0.

Status

Fixed

5 Audit Result

Audit Number	Audit Team	Audit Date	Audit Result
0X002205250003	SlowMist Security Team	2022.05.23 - 2022.05.25	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 suggestion vulnerabilities. All the 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