SLOWMIST

Smart Contract
Security Audit Report

Table Of Contents

1 Executive Summary

On 2025.10.19, the SlowMist security team received the Sigma Money team's security audit application for Sigma DAO round 2, 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.
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
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

Serial Number	Audit Class	Audit Subclass
7	Security Design 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

Sigma DAO protocol is forked from Shadow Protocol.

3.2 Vulnerability Information

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

NO	Title	Category	Level	Status
N1	Missing zero address check	Others	Suggestion	Acknowledged
N2	Missing event records	Others	Suggestion	Fixed

NO	Title	Category	Level	Status
N3	Improper variable declaration	Others	Suggestion	Acknowledged
N4	Risk of excessive authority	Design Logic Audit	Medium	Acknowledged

4 Code Overview

4.1 Contracts Description

https://github.com/SigmaMoney/dao/tree/feat/bsc

Initial audit version: 28acd3f9dd82125b67b643cb2e2c11ce5eaf5a4b

Final audit version: 21d8dd099f6523869c19ed17696052a29b074ff7

Audit Scope

```
- contracts/AccessHub.sol
- contracts/Minter.sol
- contracts/interfaces/IAccessHub.sol
- contracts/interfaces/IMinter.sol
- contracts/interfaces/IXShadow.sol
- contracts/xShadow/XShadow.sol
```

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:

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

AccessHub			
initialize	External	Can Modify State	initializer
reinit	External	Can Modify State	timelocked
initializeVoter	External	Can Modify State	timelocked
addVestingSchedule	External	Can Modify State	onlyRole
removeVestingSchedule	External	Can Modify State	onlyRole
startRebase	External	Can Modify State	onlyRole
setNewGovernorInVoter	External	Can Modify State	onlyRole
createSigmaGauge	External	Can Modify State	onlyRole
createVeFunderGauge	External	Can Modify State	onlyRole
governanceWhitelist	External	Can Modify State	onlyRole
killGauge	External	Can Modify State	onlyRole
reviveGauge	External	Can Modify State	onlyRole
setEmissionsRatioInVoter	External	Can Modify State	onlyRole
retrieveStuckEmissionsToGovernance	External	Can Modify State	onlyRole
setSigmaGaugePreallocation	External	Can Modify State	onlyRole
createCLGaugeOveridden	External	Can Modify State	onlyRole
transferWhitelistInXShadow	External	Can Modify State	onlyRole
toggleXShadowGovernance	External	Can Modify State	onlyRole
operatorRedeemXShadow	External	Can Modify State	onlyRole
migrateOperator	External	Can Modify State	onlyRole
rescueTrappedTokens	External	Can Modify State	onlyRole
setExemptionToInXShadow	External	Can Modify State	onlyRole

AccessHub			
setEmissionsMultiplierInMinter	External	Can Modify State	onlyRole
setGaugeActiveInMinter	External	Can Modify State	onlyRole
augmentGaugeRewardsForPair	External	Can Modify State	onlyRole
removeFeeDistributorRewards	External	Can Modify State	onlyRole
setCooldownExemption	External	Can Modify State	timelocked
setNewRebaseStreamingDuration	External	Can Modify State	timelocked
setNewVoteModuleCooldown	External	Can Modify State	timelocked
kickInactive	External	Can Modify State	onlyRole
execute	External	Can Modify State	timelocked
setNewTimelock	External	Can Modify State	timelocked

Minter			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Public	Can Modify State	-
kickoff	External	Can Modify State	-
setGaugeActive	External	Can Modify State	onlyGovernance
updatePeriod	External	Can Modify State	-
startEmissions	External	Can Modify State	-
updateEmissionsMultiplier	External	Can Modify State	onlyGovernance
calculateWeeklyEmissions	Public	-	-
releaseSigmaVesting	External	Can Modify State	-
getPeriod	Public	-	-
getEpoch	Public	-	-

Minter			
_safeTransfer	Internal	Can Modify State	-

XShadow			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Public	Can Modify State	ERC20
startRebase	External	Can Modify State	onlyGovernance
pause	External	Can Modify State	onlyGovernance
unpause	External	Can Modify State	onlyGovernance
_update	Internal	Can Modify State	-
_isExempted	Internal	-	-
convertEmissionsToken	External	Can Modify State	whenNotPaused
rebase	External	Can Modify State	whenNotPaused
exit	External	Can Modify State	whenNotPaused
createVest	External	Can Modify State	whenNotPaused
exitVest	External	Can Modify State	whenNotPaused
operatorRedeem	External	Can Modify State	onlyGovernance
rescueTrappedTokens	External	Can Modify State	onlyGovernance
migrateOperator	External	Can Modify State	onlyGovernance
setExemption	External	Can Modify State	onlyGovernance
setExemptionTo	External	Can Modify State	onlyGovernance
getBalanceResiding	Public	-	-
usersTotalVests	Public	-	-
getVestInfo	Public	-	-

XShadow			
isExempt	External	-	-
shadow	External	-	-

4.3 Vulnerability Summary

[N1] [Suggestion] Missing zero address check

Category: Others

Content

1.In the Minter contract, the `constructor` function lacks zero address checks for _accessHub and _operator.

- dao/contracts/Minter.sol#L53-L56

```
constructor(address _accessHub, address _operator) {
    //...
}
```

2.In the AccessHub contract, the `initialize` function lacks a zero address check for the address type parameter.

- dao/contracts/AccessHub.sol#L62-L85

```
function initialize(InitParams calldata params) external initializer {
    //...
}
```

3.In the AccessHub contract, the `reinit` function lacks a zero address check for the address type parameter.

- dao/contracts/AccessHub.sol#L87-L95

```
function reinit(InitParams calldata params) external timelocked {
    //...
}
```

Solution

It is recommended to add zero address check.

Status

Acknowledged

[N2] [Suggestion] Missing event records

Category: Others

Content

In the AccessHub contract, the `setGaugeActiveInMinter` function lacks event logging when setting key variables.

- dao/contracts/AccessHub.sol#L306-L310

```
    function setGaugeActiveInMinter(
        bool _isGaugeActive
    ) external onlyRole(PROTOCOL_OPERATOR) {
        minter.setGaugeActive(_isGaugeActive);
    }
```

Solution

It is recommended to add time records.

Status

Fixed

[N3] [Suggestion] Improper variable declaration

Category: Others

Content

In the Minter contract, `operator` and `accessHub` should be immutable.

- dao/contracts/Minter.sol#L36, L38

```
    address public operator;
    address public accessHub;
```

Solution

It is recommended to add appropriate modifiers to variables.

Status

Acknowledged

[N4] [Medium] Risk of excessive authority

Category: Design Logic Audit

Content

1.In the Minter contract, the `operator` role can call the `kickoff` function to set the key contract address and initial issuance parameters at one time, and can call the `startEmissions` function to start the entire token issuance process.

- dao/contracts/Minter.sol#L59-L88, L133-L144

```
function kickoff(
    address _shadow,
    address _voter,
    uint256 _initialWeeklyEmissions,
    uint256 _initialMultiplier,
    uint256 _multiplierUpdatePeriod,
    address _xShadow,
    address _sigmaVesting
 ) external {}

function startEmissions() external {}
```

2.In the AccessHub contract, the `DEFAULT_ADMIN_ROLE` role can grant and revoke all other roles and has the `kickInactive` function permission to kick out inactive users who have not voted and reset their voting status.

- dao/contracts/AccessHub.sol#L380-L402

```
function kickInactive(
    address[] calldata _nonparticipants
 ) external onlyRole(DEFAULT_ADMIN_ROLE) {}
```

3.In the AccessHub contract, the PROTOCOL_OPERATOR role has operational management permissions, including: managing the addition and removal of SigmaVesting, setting the governor of the Voter contract, creating and managing SigmaGauge and VeFunderGauge, whitelist management (token whitelist, reward whitelist), pause/unpause Gauge, setting emission ratios and pre-allocation, controlling xShadow's transfer whitelist and pause/unpause functions, operator redemption and migration, adjusting Minter's emission multiples, and switching Gauge token emission status, among other core functions.

- dao/contracts/AccessHub.sol#L130-L137, L140-L142, L147-L152, L154-L159, L161-L167, L170-L187, L190-L200, L203-L211, L214-L218, L221-L226, L229-L234, L236-L242, L247-L254, L257-L262, L265-L269, L272-L276, L279-L284, L287-L294, L299-L303, L305-L309, L314-L338, L340-L351

```
   function addVestingSchedule(
        address _beneficiary,
        address _tokenAddress,
        uint8 _category,
        ISigmaVesting.UnlockEntry[] calldata _entries
    ) external onlyRole(PROTOCOL_OPERATOR) {}

    function removeVestingSchedule(address _beneficiary, address _tokenAddress)
external onlyRole(PROTOCOL_OPERATOR) {}

    function startRebase(address _voteModule, address _voter) external
onlyRole(PROTOCOL_OPERATOR) {}

    function setNewGovernorInVoter(address _newGovernor) external
onlyRole(PROTOCOL_OPERATOR) {}

    function createSigmaGauge(address _pool, uint256 _preallocationBps) external
onlyRole(PROTOCOL_OPERATOR) {}

    function createVeFunderGauge(address _receiver, uint256 _maxEmission, address
_pool) external onlyRole(PROTOCOL_OPERATOR) {}

    function governanceWhitelist(address[] calldata _token, bool[] calldata
_whitelisted) external onlyRole(PROTOCOL_OPERATOR) {}

    function killGauge(address[] calldata _pairs) external
onlyRole(PROTOCOL_OPERATOR) {}

    function reviveGauge(address[] calldata _pairs) external
onlyRole(PROTOCOL_OPERATOR) {}

    function setEmissionsRatioInVoter(uint256 _pct) external
onlyRole(PROTOCOL_OPERATOR) {}

    function retrieveStuckEmissionsToGovernance(address _gauge, uint256 _period)
external onlyRole(PROTOCOL_OPERATOR) {}

    function setSigmaGaugePreallocation(address _gauge, uint256 _preallocationBps)
```

```
external onlyRole(PROTOCOL_OPERATOR) {}

    function createCLGaugeOveridden(address tokenA, address tokenB, int24
tickSpacing) external onlyRole(PROTOCOL_OPERATOR) {}

    function transferWhitelistInXShadow(address[] calldata _who, bool[] calldata
_whitelisted) external onlyRole(PROTOCOL_OPERATOR) {}

    function toggleXShadowGovernance(bool enable) external
onlyRole(PROTOCOL_OPERATOR) {}

    function operatorRedeemXShadow(uint256 _amount) external
onlyRole(PROTOCOL_OPERATOR) {}

    function migrateOperator(address _operator) external onlyRole(PROTOCOL_OPERATOR)
{}

    function rescueTrappedTokens(address[] calldata _tokens, uint256[] calldata
_amounts) external onlyRole(PROTOCOL_OPERATOR) {}

    function setExemptionToInXShadow(address[] calldata _who, bool[] calldata
_whitelisted) external onlyRole(PROTOCOL_OPERATOR) {}

    function setEmissionsMultiplierInMinter(uint256 _multiplier) external
onlyRole(PROTOCOL_OPERATOR) {}

    function setGaugeActiveInMinter(bool _isGaugeActive) external
onlyRole(PROTOCOL_OPERATOR) {}

    function augmentGaugeRewardsForPair(address[] calldata _pools, address[] calldata
_rewards, bool[] calldata _addReward) external onlyRole(PROTOCOL_OPERATOR) {}

    function removeFeeDistributorRewards(address[] calldata _pools, address[]
calldata _rewards) external onlyRole(PROTOCOL_OPERATOR) {}
```

Solution

In the short term, transferring owner ownership to multisig contracts is an effective solution to avoid single-point risk. But in the long run, it is a more reasonable solution to implement a privilege separation strategy and set up multiple privileged roles to manage each privileged function separately. And the authority involving user funds should be managed by the community, and the EOA address can manage the authority involving emergency contract suspension. This ensures both a quick response to threats and the safety of user funds.

Status

Acknowledged

5 Audit Result

Audit Number	Audit Team	Audit Date	Audit Result
0X002510190001	SlowMist Security Team	2025.10.19 - 2025.10.19	Medium Risk

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 medium risk, 2 suggestion.

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