



# Smart Contract Security Audit Report



# Table Of Contents

<b>1 Executive Summary</b>	_____
<b>2 Audit Methodology</b>	_____
<b>3 Project Overview</b>	_____
3.1 Project Introduction	_____
3.2 Vulnerability Information	_____
<b>4 Code Overview</b>	_____
4.1 Contracts Description	_____
4.2 Visibility Description	_____
4.3 Vulnerability Summary	_____
<b>5 Audit Result</b>	_____
<b>6 Statement</b>	_____

# 1 Executive Summary

On 2024.05.21, the SlowMist security team received the BitKeep team's security audit application for Bwb Claim Solana Contract, 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.
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
- 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
- Account substitution attack Audit
- Malicious Event Log Audit

## 3 Project Overview

### 3.1 Project Introduction

solana contract for bwb collection and pledges.

## 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	Malicious Event Log Audit	Suggestion	Acknowledged
N2	Risk of excessive authority	Authority Control Vulnerability Audit	Medium	Acknowledged
N3	Preemptive Initialization	Race Conditions Vulnerability	Suggestion	Fixed

## 4 Code Overview

### 4.1 Contracts Description

<https://github.com/bitkeepwallet/bwb-claim>

commit:

6dc547623ba1dca6e93c7d01bc21fa87fd7e5192

review commit:

aff7ddf42b86abbbcc466154d587b0dbba44f190

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:

cosigner_distributor			
Function Name	Account check coverage	Auth Signer	Params Check
new_distributor	6/6	-	0/5

cosigner_distributor			
claim	7/9	cosigner	1/2
update_receiver	2/2	admin_auth	0/1
update_cosigner	2/2	admin_auth	0/1
update_operator	2/2	admin_auth	0/1
set_is_paused	2/2	operator	0/1
withdraw_bwb_token	5/5	operator	0/1
withdraw_other_token	7/7	operator	0/1

merkle_distributor			
Function Name	Account check coverage	Auth Signer	Params Check
new_distributor	6/6	-	0/8
init_merkle_roots	4/4	operator	0/2
claim	8/10	cosigner	9/9
update_receiver	2/2	admin_auth	0/1
update_cosigner	2/2	admin_auth	0/1
update_operator	2/2	admin_auth	0/1
set_is_paused	2/2	operator	0/1
withdraw_bwb_token	5/5	operator	0/1
withdraw_other_token	7/7	operator	0/1

## 4.3 Vulnerability Summary

[N1] [Suggestion] Missing event record

## Category: Malicious Event Log Audit

### Content

The following parameters are updated without event logging.

- cosigner-distributor/programs/cosigner-distributor/src/lib.rs

```
update_receiver  
update_cosigner  
update_operator  
set_is_paused
```

The following parameters are updated without event logging.

- merkle-distributor/programs/merkle-distributor/src/lib.rs

```
init_merkle_roots  
update_receiver  
update_cosigner  
update_operator  
set_is_paused
```

### Solution

Record the corresponding event.

### Status

Acknowledged

## [N2] [Medium] Risk of excessive authority

### Category: Authority Control Vulnerability Audit

### Content

The role of `admin` can be set `operator` role and `receiver`, if the private key of admin role is leaked, it will cause the loss of contract funds.

- cosigner-distributor/programs/cosigner-distributor/src/lib.rs

```
admin_auth can update_receiver  
admin_auth can update_cosigner  
admin_auth can update_operator
```

```
operator can set_is_paused
operator can withdraw_bwb_token
operator can withdraw_other_token
```

The role of `admin` can be set `operator` role and `receiver`, if the private key of admin role is leaked, it will cause the loss of contract funds.

- merkle-distributor/programs/merkle-distributor/src/lib.rs

```
operator can init_merkle_roots
admin_auth can update_receiver
admin_auth can update_cosigner
admin_auth can update_operator
operator can set_is_paused
operator can withdraw_bwb_token
operator can withdraw_other_token
```

## 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. The authority involving user funds should be managed by the community, and the authority involving emergency contract suspension can be managed by the EOA address. This ensures both a quick response to threats and the safety of user funds.

## Status

Acknowledged; The owner will be a multi-signature address to manage.operator can only withdraw funds to the specified receiver address

## [N3] [Suggestion] Preemptive Initialization

### Category: Race Conditions Vulnerability

### Content

The new\_distributor function does not restrict the caller and is subject to a preemptive initialization attack.

- cosigner-distributor/programs/cosigner-distributor/src/lib.rs

The new\_distributor function does not restrict the caller and is subject to a preemptive initialization attack.



- merkle-distributor/programs/merkle-distributor/src/lib.rs

**Solution**

Deployment contracts and settings can be executed in a deployment transaction.

**Status**

Fixed

## 5 Audit Result

Audit Number	Audit Team	Audit Date	Audit Result
0X002405210001	SlowMist Security Team	2024.05.21 - 2024.05.21	Low 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 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](http://www.slowmist.com)



**E-mail**  
[team@slowmist.com](mailto:team@slowmist.com)



**Twitter**  
[@SlowMist\\_Team](https://twitter.com/SlowMist_Team)



**Github**  
<https://github.com/slowmist>