Blockchain Security Audit Report

[2021]

Table Of Contents

1 Executive Summary

On 2021.11.01, the SlowMist security team received the bit-country team's security audit application for Metaverse Network, 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" 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.

In black box testing and gray box testing, we use methods such as fuzz testing and script testing to test the robustness of the interface or the stability of the components by feeding random data or constructing data with a specific structure, and to mine some boundaries Abnormal performance of the system under conditions such as bugs or abnormal performance. In white box testing, we use methods such as code review, combined with the relevant experience accumulated by the security team on known blockchain security vulnerabilities, to analyze the object definition and logic implementation of the code to ensure that the code has the key components of the key logic. Realize no known vulnerabilities; at the same time, enter the vulnerability mining mode for new scenarios and new technologies, and find possible 0day errors.

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:

NO.	Audit Items	Result
1	Others	Some Risks
2	State Consistency Audit	Passed

NO.	Audit Items	Result
3	Failure Rollback Audit	Passed
4	Unit Test Audit	Passed
5	Value Overflow Audit	Passed
6	Parameter Verification Audit	Passed
7	Error Unhandle Audit	Passed
8	Boundary Check Audit	Passed

3 Project Overview

3.1 Project Introduction

Offical description: Metaverse Network is an EVM-enabled blockchain network for user-created metaverses and games.

3.2 Coverage

Target Code and Revision:

https://github.com/bit-country/Metaverse-Network/tree/master/pallets

commit: cb31e5938de100fe6b682db41ffd5bc9dcc5a888

review commit:2a74e9559c6e02a59a8babb0698cdae4b08e104a

NFT Pallet - NFT creation, basic functionality.

nft/src/lib.rs 435 lines

Auction Pallet - NFT marketplace

auction/src/lib.rs 742 lines

Metaverse Pallet - metaverse creation

metaverse/src/lib.rs 215 lines

Tokenization Pallet - multi-currency system

tokenization/src/lib.rs 387 lines

Estate Pallet - Land registry system

estate/src/lib.rs 300 lines

2079 lines in total.

3.3 Vulnerability Information

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

NO	Title	Category	Level	Status
N1	The returned result is not handled	Others	Low	Fixed
N2	Redundant conditions	Others	Suggestion	Fixed
N3	The size of the list is not limited	Others	Medium	Fixed
N4	No length check for parameters	Others	Low	Fixed
N5	Gas optimization	Others	Suggestion	Fixed

4 Findings

4.1 Visibility Description

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

auction			
Function Name	Parameter verification	State consistency	Modifiers
bid	3/3	ok	ensure_signed
bid_local	4/4	ok	ensure_signed
buy_now	3/3	ok	ensure_signed
buy_now_local	4/4	ok	ensure_signed
create_new_auction	4/5	ok	ensure_signed
create_new_buy_now	4/5	ok	ensure_signed

estate			
Function Name	Parameter verification	State consistency	Modifiers
set_max_bounds	1/3	ok	ensure_root
mint_land	4/4	ok	ensure_root
mint_lands	4/4	ok	ensure_root
transfer_land	4/4	ok	ensure_signed
mint_estate	3/4	ok	ensure_root
create_estate	3/4	ok	ensure_root
transfer_estate	2/3	ok	ensure_signed
deploy_land_block	4/4	ok	ensure_signed
issue_undeployed_land_blocks	1/5	ok	ensure_root

estate

freeze_undeployed_land_blocks	1/2	ok	ensure_root
transfer_undeployed_land_blocks	2/3	ok	ensure_signed
burn_undeployed_land_blocks	1/2	ok	ensure_root
approve_undeployed_land_blocks	2/3	ok	ensure_signed
unapprove_undeployed_land_blocks	2/2	ok	ensure_signed

nft

Function Name	Parameter verification	State consistency	Modifiers
create_group	1/3	ok	ensure_root
create_class	3/5	ok	ensure_signed
mint	6/6	ok	ensure_signed
transfer	3/3	ok	ensure_signed
transfer_batch	2/2	ok	ensure_signed
sign_asset	1/2	ok	ensure_signed

metaverse

Function Name	Parameter verification	State consistency	Modifiers
create_metaverse	2/2	ok	ensure_signed
transfer_metaverse	3/3	ok	ensure_signed
freeze_metaverse	2/2	ok	ensure_origin
unfreeze_metaverse	2/2	ok	ensure_origin

metaverse			
destroy_metaverse	2/2	ok	ensure_origin

tokenization			
Function Name	Parameter verification	State consistency	Modifiers
mint_token	2/6	ok	ensure_signed
transfer	3/4	ok	ensure_signed
claim	2/2	ok	ensure_signed
vested_transfer	2/3	ok	ensure_signed
update_vesting_schedules	2/4	ok	ensure_root

4.2 Vulnerability Summary

[N1] [Low] The returned result is not handled

Category: Others

Content

pallets/nft/src/lib.rs

```
fn sign_asset
```

The returned result is not handled.

```
AssetSupporters: :<T > ::try_mutate(asset_id, |supporters | ->DispatchResult {
    let supporters = supporters.as_mut().ok_or("Empty supporters") ? ;
    supporters.push(sender);
    Ok(())
}); //SlowMist
```

Solution

Status

Fixed

[N2] [Suggestion] Redundant conditions

Category: Others

Content

pallets/tokenization/src/lib.rs

```
fn mint_social_token
```

```
ensure ! (supply_percent > 0u128 && supply_percent >= 20u128, Error: :<T >
::InitialFungibleTokenSupplyIsTooLow); //SlowMist Greater than 20 is greater than 0
```

Solution

Status

Fixed

[N3] [Medium] The size of the list is not limited

Category: Others

Content

pallets/nft/src/lib.rs

```
fn transfer_batch
```

```
for (_i, x) in tos.iter().enumerate() { //SlowMist When the length of tos is too
long, it may cause denial of service
    let item = &x;
    let owner = &sender.clone();
    let asset = Assets: :<T > ::get(item.1).ok_or(Error: :<T > ::AssetIdNotFound) ? ;

    let class_info = NftModule: :<T > ::classes(asset.0).ok_or(Error: :<T >
::ClassIdNotFound) ? ;
    let data = class_info.data;

    match data.token_type {
        TokenType: :Transferable = >{
            let asset_info = NftModule: :<T > ::tokens(asset.0, asset.1).ok_or(Error:
:<T > ::AssetInfoNotFound) ? ;
            ensure ! (owner.clone() == asset_info.owner, Error: :<T >
::NoPermission);
            Self: :handle_asset_ownership_transfer( & owner, &item.0, item.1);
            NftModule: :<T > ::transfer( & owner, &item.0, (asset.0, asset.1)) ? ;
            Self: :deposit_event(Event: :<T > ::TransferedNft(owner.clone(),
item.0.clone(), asset.1.clone(), ));
        }
        _ = >(),
    };
}
```

pallets/nft/src/lib.rs

```
fn mint
```

```
for _ in 0..quantity { //SlowMist When the length of quantity is too long, it may
cause denial of service
    let asset_id = NextAssetId: :<T > ::try_mutate( | id | ->Result < AssetId,
DispatchError > {
        let current_id = *id; * id = id.checked_add(One: :one()).ok_or(Error: :<T >
::NoAvailableAssetId) ? ;
        Ok(current_id)
    }) ? ;
    new_asset_ids.push(asset_id);

    if AssetsByOwner: :<T > ::contains_key( & sender) {
        AssetsByOwner: :<T > ::try_mutate( & sender, |asset_ids | ->DispatchResult {
            /// Check if the asset_id already in the owner
            ensure ! (!asset_ids.iter().any( | i | asset_id == *i), Error: :<T >
::AssetIdAlreadyExist);
            asset_ids.push(asset_id);
            Ok(())
        }) ? ;
    } else {
        let mut assets = Vec: :<AssetId > ::new();
        assets.push(asset_id);
        AssetsByOwner: :<T > ::insert( & sender, assets)
    }

    let token_id = NftModule: :<T > ::mint( & sender, class_id, metadata.clone(),
new_nft_data.clone()) ? ;
    Assets: :<T > ::insert(asset_id, (class_id, token_id));
    last_token_id = token_id;
}
```

Solution

Status

Fixed

[N4] [Low] No length check for parameters

Category: Others

Content

pallets/nft/src/lib.rs

```
fn create_class
```

```
let class_data = NftClassData {
    deposit: class_deposit,
    token_type,
    collection_type,
    metadata: metadata.clone(),//SlowMist No maximum length check is performed on the
incoming metadata
    total_supply: Default::default(),
    initial_supply: Default::default(),
};
```

pallets/nft/src/lib.rs

```
fn mint
```

```
let new_nft_data = NftAssetData {
    deposit,
    name, //SlowMist No maximum length check is performed on the incoming name
    description, //SlowMist No maximum length check is performed on the incoming
description
    properties: metadata.clone() //SlowMist No maximum length check is performed on
the incoming metadata,
};
```

If the user passes in a large amount of data, it will cause a denial of service

Solution

Status

Fixed

[N5] [Suggestion] Gas optimization

Category: Others

Content

pallets/tokenization/src/lib.rs

```
fn transfer_from
```

This is an unexpected result and should not return OK

```
if amount.is_zero() || from == to {
    return Ok(()); //SlowMist For an invalid result, ok is returned and there is no
corresponding event record.
}
```

Solution

Status

Fixed

5 Audit Result

Audit Number	Audit Team	Audit Date	Audit Result

Audit Number	Audit Team	Audit Date	Audit Result
BCA002111120001	SlowMist Security Team	2021.11.01 - 2021.11.12	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 medium risk, 2 low risk, 2 suggestion vulnerabilities. All the 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.