Blockchain Security Audit Report

Table Of Contents

1 Executive Summary

On 2022.04.13, the SlowMist security team received the UINB team's security audit application for Fusotao, 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	Replay Vulnerability	Passed
2	Reordering Vulnerability	Passed

NO.	Audit Items	Result
3	Race Conditions Vulnerability	Passed
4	Authority Control Vulnerability	Passed
5	Block data Dependence Vulnerability	Passed
6	Explicit Visibility of Functions Audit	Passed
7	Arithmetic Accuracy Deviation Vulnerability	Passed
8	Malicious Event Log Audit	Passed
9	Others	Some Risks
10	State Consistency Audit	Passed
11	Failure Rollback Audit	Some Risks
12	Unit Test Audit	Passed
13	Value Overflow Audit	Some Risks
14	Parameter Verification Audit	Some Risks
15	Error Unhandle Audit	Passed
16	Boundary Check Audit	Passed
17	SAST	Passed

3 Project Overview

3.1 Project Introduction

Fusotao is a permissionless blockchain based on Substrate (opens new window)and hosted on Octopus Network (opens new window). It contains a highly customized runtime focused on verifying off-chain matching system. The main purpose of Fusotao is to seperate the role of banks from the centralized exchanges which are shouldn't combined into one, rather than building another general purpose blockchain or a dex on a specific blockchain.

3.2 Coverage

Target Code and Revision:

https://github.com/uinb/fusotao-protocol/tree/vodka-release

commit: 997e3d700257c4c5f31843bf19baced8b6b902ea

Fixed version:

https://github.com/uinb/fusotao-protocol/blob/master

commit: bb865368a0a4e5a43af982b445a62411f2e635bf

3.3 Vulnerability Information

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

NO	Title	Category	Level	Status
N1	Overflow risks	Value Overflow Audit	Medium	Fixed
N2	Calculate inaccurate risk	Value Overflow Audit	Suggestion	Ignored
N3	Incorrect parameters can be submitted	Parameter Verification Audit	High	Fixed
N4	DoS attack risk	Others	Low	Confirmed
N5	Missing macros	Failure Rollback Audit	Low	Fixed
N6	Dexs malicious registration risk	Others	Low	Fixed

4 Findings

4.1 Visibility Description

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

pallet-fuso-foundation				
Function Name	Parameter verification	Overflow	Authority	Weight
on_initialize	1/1	1 issue	-	0 ~ 100_000_000+

pallet-fuso-reward				
Function Name	Parameter verification	Overflow	Authority	Weight
take_reward	1/1	ok	ensure_signed	10_000_000

pallet-fuso-token				
Function Name	Parameter verification	Overflow	Authority	Weight
mark_stable	2/2	ok	ensure_root	0
transfer	4/4	ok	ensure_signed	71_780_000+
issue	5/5	ok	ensure_signed	10_000+

pallet-fuso-verifier				
Function Name	Parameter verification	Overflow	Authority	Weight
on_initialize	1/1	2 issues	-	0+
register	2/2	ok	ensure_signed	31_660_780_000+

pallet-fuso-verifier				
evict	2/2	ok	ensure_root	0
verify	2/2	ok	ensure_signed	31_660_780_000+
stake	2/3	ok	ensure_signed	31_660_780_000+
unstake	3/3	ok	ensure_signed	31_660_780_000+
claim_shares	2/2	ok	ensure_signed	31_660_780_000+
authorize	4/4	ok	ensure_signed	31_660_780_000+
revoke	4/4	ok	ensure_signed	31_660_780_000+

4.2 Vulnerability Summary

[N1] [Medium] Overflow risks

Category: Value Overflow Audit

Content

- fusotao-protocol/pallet-fuso-verifier/src/lib.rs

```
//#L1148
let tq_delta = (tqa0 + tqf0) - (tqa1 + tqf1);
//...ignore other places...
```

- fusotao-protocol/pallet-fuso-reward/src/lib.rs

```
//#L138
Ok(b.0 += confirmed)
//#L168
let a = p * era_reward;
```

There are some risks of value overflow.

Solution

Use `checkd_add/checkd_sub/checkd_mul/checkd_div` to avoid value overflow, instead of using `+-*/, +=, -=`

Status

Fixed; Fixed in latest version.

[N2] [Suggestion] Calculate inaccurate risk

Category: Value Overflow Audit

Content

- fusotao-protocol/pallet-fuso-foundation/src/lib.rs
- fusotao-protocol/pallet-fuso-token/src/weights.rs
- fusotao-protocol/pallet-fuso-verifier/src/lib.rs
- fusotao-protocol/pallet-fuso-verifier/src/weights.rs

```
weight = weight.saturating_add(RocksDbWeight::get().reads(1 as Weight));
//...ignore other places using saturating_add/saturating_sub
```

Saturating at the numeric bounds instead of overflowing, The returned result is inaccurate.

Solution

Use `checked_add/checked_sub/checked_mul/checked_div` instead of `saturating_add/saturating_sub/saturating_mul/saturating_div`.

Status

Ignored; There is not risk when calculating weights.

[N3] [High] Incorrect parameters can be submitted

Category: Parameter Verification Audit

Content

- fusotao-protocol/pallet-fuso-token/src/lib.rs

```
pub fn issue(
    origin: OriginFor<T>,
    decimals: u8,
    stable: bool,
    symbol: Vec<u8>,
    contract: Vec<u8>,
) -> DispatchResultWithPostInfo {
    let _ = ensure_signed(origin)?;
    ensure!(decimals <= MAX_DECIMALS, Error::<T>::InvalidDecimals);
    let name = AsciiStr::from_ascii(&symbol);
    ensure!(name.is_ok(), Error::<T>::InvalidTokenName);
    let name = name.unwrap();
    ensure!(
        name.len() >= 2 && name.len() <= 8,
        Error::<T>::InvalidTokenName
    );
    ensure!(
        !TokenByName::<T>::contains_key(&contract),
        Error::<T>::InvalidToken
    );
    let id = Self::next_token_id();
    NextTokenId::<T>::mutate(|id| *id += One::one());
    TokenByName::<T>::insert(contract.clone(), id);
    Tokens::<T>::insert(
        id,
        XToken::NEP141(
            symbol.clone(),
            contract.clone(),
            Zero::zero(),
            stable,
            decimals,
        ),
    );
    Self::deposit_event(Event::TokenIssued(id, symbol, contract));
    Ok(().into())
}
```

If someone submit `contract` with error `name/symbol/decimals`, we'll not able to correct it.

Solution

Only administrator can call `issue` function or check the submission.

Status

Fixed; Fixed in latest version.

[N4] [Low] DoS attack risk

Category: Others

Content

If there are too many transactions on the chain, the verification of the dex transaction will time out and the dex will be `EVICTED`.

Solution

Reduce the transaction rate of dex and increase transaction fees when the system is under high pressure.

Status

Confirmed

[N5] [Low] Missing macros

Category: Failure Rollback Audit

Content

- fusotao-protocol/pallet-fuso-verifier/src/lib.rs

```
fn put_profit(
    dominator: &T::AccountId,
    season: Season,
```

```
    currency: TokenId<T>,
    balance: Balance<T>,
) -> DispatchResult {
    if balance == Zero::zero() {
        Ok(())
    } else {
        Bonuses::<T>::try_mutate(dominator, season, |b| {
            b.profit
                .entry(currency)
                .and_modify(|p| *p += balance)
                .or_insert(balance);
            Ok(())
        })
    }
}
```

Missing macros `#[transactional]`

Solution

Add macros `#[transactional]`

Status

Fixed; Fixed in latest version.

[N6] [Low] Dexs malicious registration risk

Category: Others

Content

- fusotao-protocol/pallet-fuso-verifier/src/lib.rs

```
dominator.staked += amount;
dominator.status = if dominator.staked >= T::DominatorOnlineThreshold::get() {
    DOMINATOR_ACTIVE
} else {
    DOMINATOR_INACTIVE
};
```

Only some `TAO` token needed to active a dex, users may maliciously register exchanges.

Solution

Administrators should manually check active permissions for dexs.

Status

Fixed; Fixed in latest version.

5 Audit Result

Audit Number	Audit Team	Audit Date	Audit Result
0X002204260001	SlowMist Security Team	2022.04.13 - 2022.04.26	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 high risk, 1 medium risk, 3 low risk, 1 suggestion vulnerabilities. And 1 low risk vulnerabilities were confirmed and being fixed; 1 suggestion vulnerabilities were ignored; 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