verichains

SECURITY AUDIT OF

FLOWX SWAP SMART CONTRACTS

FlowX.finance

Public Report

Aug 28, 2023

Verichains Lab

info@verichains.io

https://www.verichains.io

Driving Technology > Forward

ABBREVIATIONS

Name	Description
Sui Blockchain	Sui is an innovative, decentralized Layer 1 blockchain that redefines asset ownership. Sui Move feels like a paradigm change in web3 development. Treating objects as 1st class citizens brings composability to a whole new level. Polymedia. We are thrilled to be building on Sui.
Sui Object	The basic unit of storage in Sui is object. In contrast to many other blockchains where storage is centered around accounts and each account contains a key-value store, Sui's storage is centered around objects.
Move	Move is a new programming language that implements all the transactions on the Aptos/Sui blockchain.
Move Module	A Move module defines the rules for updating the global state of the Aptos/Sui blockchain. In the Aptos/Sui protocol, a Move module is a smart contract.

EXECUTIVE SUMMARY

This Security Audit Report was prepared by Verichains Lab on Aug 28, 2023. We would like to thank the Flowx for trusting Verichains Lab in auditing smart contracts. Delivering high-quality audits is always our top priority.

This audit focused on identifying security flaws in code and the design of the Flowx Swap Smart Contracts. The scope of the audit is limited to the source code files provided to Verichains. Verichains Lab completed the assessment using manual, static, and dynamic analysis techniques.

During the audit process, the audit team had identified some vulnerable issues in the smart contracts code.

TABLE OF CONTENTS

1. MANAGEMENT SUMMARY

1.1. About Flowx Swap Smart Contracts

FlowX is the ecosystem-focused decentralized exchange built on the Sui Blockchain.

Flowx Swap Contract revolutionizes decentralized exchanges on the Sui blockchain by drawing inspiration from the proven principles of Uniswap V2.

1.2. Audit scope

This audit focused on identifying security flaws in code and the design of the smart contracts of Flowx Swap Smart Contracts.

It was conducted on commit `c9fccad83741655fd00daf81a92b5c691150a6e1` from git repository link: *https://github.com/FlowX-Finance/move-contracts/*.

The following files were made available in the course of the review:

- exchange/factory.move
- exchange/pair.move
- exchange/router.move
- exchange/treasury.move
- utils/comparator.move
- utils/math.move
- utils/swap_utils.move
- utils/type_helper.move

1.3. Audit methodology

Our security audit process for smart contract includes two steps:

- Smart contract codes are scanned/tested for commonly known and more specific vulnerabilities using public and RK87, our in-house smart contract security analysis tool.
- 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 were considered during the audit of the smart contract:

- Numerical precision errors
- Transaction-Ordering Dependence
- DoS with (Unexpected) revert
- Gas Usage, Gas Limit and Loops

- Logic Flaws

For vulnerabilities, we categorize the findings into categories as listed in table below, depending on their severity level:

SEVERITY LEVEL	DESCRIPTION
CRITICAL	A vulnerability that can disrupt the contract functioning; creates a critical risk to the contract; required to be fixed immediately.
HIGH	A vulnerability that could affect the desired outcome of executing the contract with high impact; needs to be fixed with high priority.
MEDIUM	A vulnerability that could affect the desired outcome of executing the contract with medium impact in a specific scenario; needs to be fixed.
LOW	An issue that does not have a significant impact, can be considered as less important.

Table 1. Severity levels

1.4. Disclaimer

Please note that security auditing cannot uncover all existing vulnerabilities, and even an audit in which no vulnerabilities are found is not a guarantee for a 100% secure smart contract. However, auditing allows discovering vulnerabilities that were unobserved, overlooked during development and areas where additional security measures are necessary.

2. AUDIT RESULT

2.1. Overview

The Flowx Swap Smart Contracts was developed using the Move programming language and deployed on the Sui Blockchain.

The Flowx swap contracts derive inspiration from the established principles of Uniswap V2. Similarly, the Flowx smart contracts adopt a comparable structure to those found in the Uniswap V2 contracts.

2.1.1. Factory module

The Factory module is responsible for creating new instances of `PairMetadata`. Each `PairMetadata` represents a token pair. Factory maintains a record of all token pairs available on the platform.

2.1.2. Pair module

The module contains essential functions for executing token swaps and updating the data based on `PairMetadata`.

2.1.3. Router module

The Router module handles the execution of trades and routes transactions to the appropriate `PairMetadata` and `Pair module`. It provides functions for swapping tokens, adding liquidity to pairs, and removing liquidity.

2.1.4. Treasury module

The module implements the object-to-point profit address during token pairs working.

2.1.5. Util modules

2.1.5.1.Comparator

Implement comparison logic which allows users to compare two values of any type.

2.1.5.2.Math

Implement some primary math functions.

2.1.5.3.Swap_utils

Implement some useful functions to calculate amount during swap process.

2.1.5.4.Type_helpers

Help to get the name of generic type.

2.2. Findings

During the audit process, we have audited the Flowx Swap Smart Contracts for commonly known and specific vulnerabilities. Here are some item passed:

Item	Status
Numerical precision errors	**Passed**
Transaction-Ordering Dependence	**Passed**
DoS with (Unexpected) revert	**Passed**
Number Overflow and Underflow	**Passed**
Access Control & Authorization	**Passed**
Assert Violation	**Passed**
Race Conditions	**Passed**
Data Consistency	**Passed**
Insecure Fee Calculation	**Passed**
Unintended Function Invocation	**Passed**

The audit team found some vulnerability issues in the given version of Flowx Swap Smart Contracts.

2.2.1. Comparator - Incorrect number comparison MEDIUM

Within the `comparator` module, the `compare` function facilitates the comparison of two values, irrespective of their data types. However, it is worth noting that when the data type of the inputs is numeric, the `compare` logic does not yield accurate results.

Provided below is a Proof of Concept (PoC) highlighting this issue:

```
#[test]
    public fun incorrect_number_comparision() {
        let value2: u128 = 767;  // 0x2ff
        let value4: u128 = 8190; // 0x1ffe
        //expect true, but it's false when running.
        assert!(is_smaller_than(&compare(&value2, &value4)), 0);
    }
```

RECOMMENDATION

Update the `compare` logic when the input type is the numeric.

UPDATES

- *Aug 28, 2023*: This issue has been acknowledged by the Flowx team.

2.2.2. Router - Mistake in description of some functions INFORMATIVE

In the `router` module, the `swap` and `mint` functions are required to pass `Coin<X>` and `Coin<Y>` like the inputs when calling these functions. But descriptions of these functions are currently based on `Uniswap V2` knowledge in EVM environment, which mandates the prior deposition of `Coin` before the function calls. This could potentially lead to user misconceptions when employing these functions.

```
/// Swaps X coins to Y coins or Y coins to X coins based on the "constant product formula".
    /// The coins must be deposited into the liquidity pool before calling this function.
    public fun swap<X, Y>(metadata: &mut PairMetadata<X, Y>, coin_x: Coin<X>, amount_x_out:
u64, coin_y: Coin<Y>, amount_y_out: u64, ctx: &mut TxContext): (Coin<X>, Coin<Y>) {
        ...
        deposit_x<X,Y>(metadata, coin_x);
        deposit_y<X,Y>(metadata, coin_y);
        ...
```

RECOMMENDATION

Remove the second description.

UPDATES

- *Aug 28, 2023*: This issue has been acknowledged by the Flowx team.

3. VERSION HISTORY

Version	Date	Status/Change	Created by
1.0	*Aug 11,2023*	Private Report	Verichains Lab
1.1	*Aug 28,2023*	Public Report	Verichains Lab

Table 2. Report versions history