# Smart Contract Security Audit Report

## DeFi Swap

# 1. Executive Summary

The SlowMist security team assessed the Crypto.com Defi Swap project’s smart contracts and finally issued an independent security audit report.

The SlowMist security team adopts the strategy of primarily focusing on white-box testing but also performing black- and grey-box testing to conduct a complete security assessment on the project to imitate real attacks.

SlowMist Smart Contract DeFi project test method are shown in the table below.

| | |
|---|---|
| Black box testing | Conduct security tests from an attacker's perspective externally. |
| Grey box testing | Conduct security testing on code module 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. |

SlowMist Smart Contract DeFi project risk level:

| | |
|---|---|
| Critical vulnerabilities | Critical vulnerabilities will have a significant impact on the security of the DeFi project, and it is strongly recommended to fix the critical vulnerabilities. |
| High-risk vulnerabilities | High-risk vulnerabilities will affect the normal operation of DeFi project. It is strongly recommended to fix high-risk vulnerabilities. |
| Medium-risk | Medium vulnerability will affect the operation of DeFi project. It is recommended |

| | |
|---|---|
| vulnerablities | to fix medium-risk vulnerabilities. |
| Low-risk vulnerabilities | Low-risk vulnerabilities may affect the operation of DeFi project in certain scenarios. It is suggested that the project party should evaluate and consider whether these vulnerabilities need to be fixed. |
| Weaknesses | There are safety risks theoretically, but it is extremely difficult to reproduce in engineering. |
| Enhancement Suggestions | There are better practices for coding or architecture. |

# 2. 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 in-house 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 attack and other Race Conditions
- Replay attack
- Reordering attack
- Short address attack
- Denial of service attack
- Transaction Ordering Dependence attack
- Conditional Completion attack
- Authority Control attack

- Integer Overflow and Underflow attack
- TimeStamp Dependence attack
- Gas Usage, Gas Limit and Loops
- Redundant fallback function
- Unsafe type Inference
- Explicit visibility of functions state variables
- Logic Flaws
- Uninitialized Storage Pointers
- Floating Points and Numerical Precision
- tx.origin Authentication
- "False top-up" Vulnerability
- Scoping and Declarations

# 3. Project Background

## 3.1 Project Introduction

DeFi Swap was designed to be the best place to swap and farm DeFi coins at the best available rate, leveraging proven and audited protocols, while offering an outstanding incentive program powered by CRO.

**Project website:**

https://crypto.com/defi

**Audit version code:**

File name: cro-defi-core.zip

SHA256: 98b9c8a238d65a6832fd99dc996b8c85f1ca1795ce47df3cf5cb49cfdaba2bca

File name: cro-defi-periphery.zip

SHA256: ec53055f10c3d33ef8fa0e0a52554a49a3beca266e82a0ff2e317ae057a480f1

**Fix version code:**

File name: cro-defi-core.zip

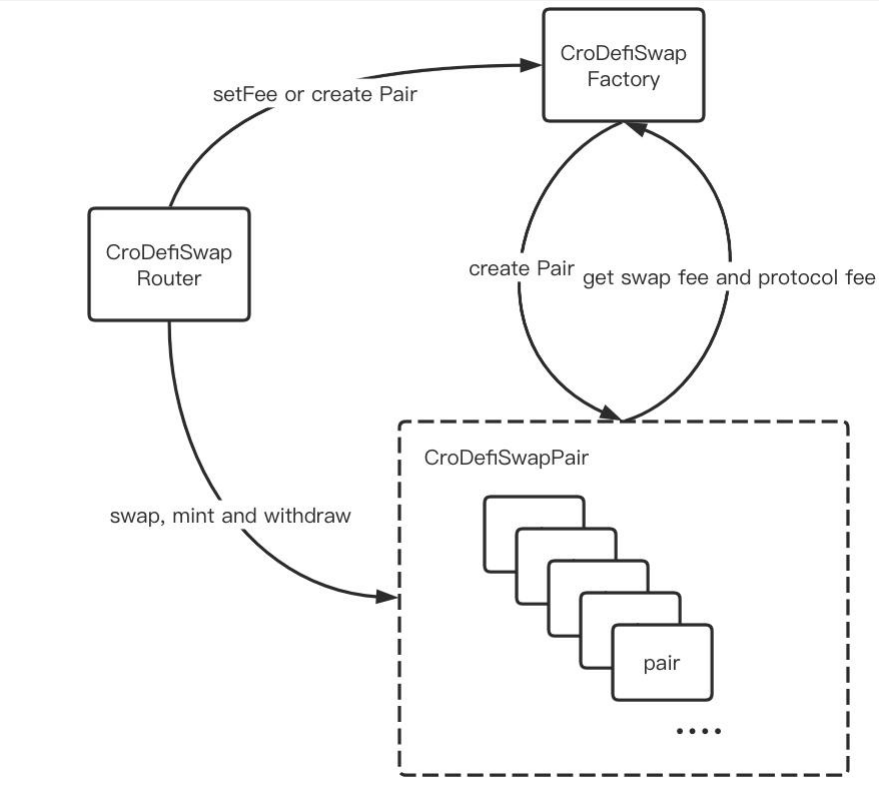
SHA256: 98b9c8a238d65a6832fd99dc996b8c85f1ca1795ce47df3cf5cb49cfdaba2bca

File name: cro-defi-periphery.zip

SHA256: e6295a39c014893bbc26559605a65084972ec6c854db03df5d1549237b0151e0

## 3.2 Project structure

DeFi Swap is a swap protocol that is divided into three parts: CroDefiSwapRouter, CroDefiSwapFactory and CroDefiSwapPair. CroDefiSwapRouter is a router for CroDefiSwapPair to help users make trades. CroDefiSwapFactory is used to create trading pairs and set protocol fees and trading fees.



# 4. Code Overview

## 4.1 Contracts Description

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

| CroDefiSwapPair | | | |
|---|---|---|---|
| Function Name | Visibility | Mutability | Modifiers |
| getReserves | Public | - | - |
| _safeTransfer | Private | Can modify state | - |
| initialize | External | Can modify state | - |
| _update | Private | Can modify state | - |
| _mintFee | Private | Can modify state | - |
| mint | External | Can modify state | lock |
| burn | External | Can modify state | lock |
| swap | External | Can modify state | lock |
| skim | External | Can modify state | lock |
| sync | External | Can modify state | lock |

| CroDefiSwapERC20 | | | |
|---|---|---|---|
| Function Name | Visibility | Mutability | Modifiers |
| _mint | Internal | Can modify state | - |
| _burn | Internal | Can modify state | - |
| _approve | Private | Can modify state | - |
| _transfer | Private | Can modify state | - |
| approve | External | Can modify state | - |
| transfer | External | Can modify state | - |
| transferFrom | External | Can modify state | - |
| permit | External | Can modify state | - |

| SafeMath | | | |
|---|---|---|---|
| Function Name | Visibility | Mutability | Modifiers |
| add | Internal | - | - |
| sub | Internal | - | - |
| mul | Internal | - | - |

| Math | | | |
|---|---|---|---|
| Function Name | Visibility | Mutability | Modifiers |
| min | Internal | - | - |
| sqrt | Internal | - | - |

| UQ112x112 | | | |
|---|---|---|---|
| Function Name | Visibility | Mutability | Modifiers |
| mul | Internal | - | - |
| min | Internal | - | - |

| CroDefiSwapFactory | | | |
|---|---|---|---|
| Function Name | Visibility | Mutability | Modifiers |
| allPairsLength | External | - | - |
| createPair | External | Can modify state | - |
| setFeeTo | External | Can modify state | - |
| setFeeSetter | External | Can modify state | - |
| setFeeToBasisPoint | External | Can modify state | - |
| setTotalFeeBasisPoint | External | Can modify state | - |

## 4.2 Code Audit

### 4.2.1 Critical Vulnerabilities

Critical severity issues can have a major impact on the security of smart contracts, and it is highly recommended to fix critical severity vulnerability.

**The audit has shown no critical severity vulnerability.**

### 4.2.2 High Risk Vulnerabilities

High severity issues can affect the normal operation of smart contracts, and it is highly recommended to fix high severity vulnerability.

**The audit has shown no high severity vulnerability.**

### 4.2.3 Medium Risk Vulnerabilities

Medium severity issues can affect the operation of a smart contract, and it is recommended to fix medium severity vulnerability.

**The audit has shown no medium severity vulnerability.**

### 4.2.4 Low Risk Vulnerabilities

Low severity issues can affect smart contracts operation in future versions of code. We recommend the project party to evaluate and consider whether these problems need to be fixed.

(1) Missing event and the upper limit to fee

The setFeeToBasisPoint and setTotalFeeBasisPoint function does not emit an event after changing the fee, which leads to user can not trace the change of the fee from blockchain. and the fee does not have a upper limit, which may make liquidity provider make no profit.

Location: CroDefiSwapFactory.sol Line:55 Line:62

```
function setFeeToBasisPoint(uint _feeToBasisPoint) external {
        require(msg.sender == feeSetter, 'CroDefiSwap: FORBIDDEN - only current feeSetter can update feeToBasisPoint');
        require(_feeToBasisPoint >= 0, 'CroDefiSwap: FORBIDDEN - _feeToBasisPoint need to be bigger than or equal to 0');
        require(_feeToBasisPoint <= totalFeeBasisPoint, 'CroDefiSwap: FORBIDDEN - _feeToBasisPoint need to be smaller than or equal to totalFeeBasisPoint');
        feeToBasisPoint = _feeToBasisPoint;
    }

    function setTotalFeeBasisPoint(uint _totalFeeBasisPoint) external {
        require(msg.sender == feeSetter, 'CroDefiSwap: FORBIDDEN - only current feeSetter can update feeToBasisPoint');
        require(_totalFeeBasisPoint >= feeToBasisPoint, 'CroDefiSwap: FORBIDDEN - _totalFeeBasisPoint need to be bigger than or equal to feeToBasisPoint');
        totalFeeBasisPoint = _totalFeeBasisPoint;
    }
```

Fix status: fixed.

## 4.2.5 Enhancement suggestion

Enhancement suggestion are suggestions for improvement of the project code, which do not cause security risks to the project, and the project party can decide whether to optimize the project code according to the improvement suggestions

**The audit has shown no medium enhancement suggestions.**

# 5. Audit Result

## 5.1 Conclusion

Audit Result : Passed

Audit Number : 0X002009100002

Audit Date : September 10, 2020

Audit Team : SlowMist Security Team

Summary conclusion: The SlowMist security team used manual reviews and in-house analysis tools to audit the provided code for security issues. No critical, high-risk and medium-risk vulnerabilities were identified but one low-risk vulnerability was found during the audit. After feedback the project team fixed the mentioned issue.

# 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 base 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 this 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