Smart Contract
Security Audit Report

Table Of Contents

1 Executive Summary

On 2022.08.26, the SlowMist security team received the StepEx team's security audit application for StepEx, 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 team 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.

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:

Serial Number	Audit Class	Audit Subclass
1	Overflow Audit	-
2	Reentrancy Attack Audit	-
3	Replay Attack Audit	-
4	Flashloan Attack Audit	-
5	Race Conditions Audit	Reordering Attack Audit
6	Permission Vulnerability Audit	Access Control Audit
		Excessive Authority Audit

Serial Number	Audit Class	Audit Subclass
7	Security Design Audit	External Module Safe Use Audit
		Compiler Version Security Audit
		Hard-coded Address Security Audit
		Fallback Function Safe Use Audit
		Show Coding Security Audit
		Function Return Value Security Audit
		External Call Function Security Audit
		Block data Dependence Security Audit
		tx.origin Authentication Security Audit
8	Denial of Service Audit	-
9	Gas Optimization Audit	-
10	Design Logic Audit	-
11	Variable Coverage Vulnerability Audit	-
12	"False Top-up" Vulnerability Audit	-
13	Scoping and Declarations Audit	-
14	Malicious Event Log Audit	-
15	Arithmetic Accuracy Deviation Audit	-
16	Uninitialized Storage Pointer Audit	-

3 Project Overview

3.1 Project Introduction

Audit Version:

Project address:

https://github.com/step-app-fi/stepex-lockdrop-core-contracts/blob/master/contracts/StepExLockdropVault.sol

commit: b3f6632e3598d38ee4c2d63ddd46bad534923de7

Project address:

https://github.com/step-app-fi/stepex-periphery

commit: a79e97b347bf4eabbc72d2601ab95a7a4d1b9333

Project address:

https://github.com/step-app-fi/stepex-core

commit: 1bb1d1ba631f393706721d72051454e3ecd66405

3.2 Vulnerability Information

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

NO	Title	Category	Level	Status
N1	Swap operation issue	Design Logic Audit	Low	Ignored
N2	Missing event records	Others	Suggestion	Ignored

4 Code Overview

4.1 Contracts Description

The main network address of the contract is as follows:

StepExFactory:

https://stepscan.io/address/0xf62b74E4a7aE8D27Cd983A54a9d24A89345413a5/contracts#address-tabs

StepExRouter:

https://stepscan.io/address/0xA4196322aA900ACc92cD5Cd978aB47e77EfA07eb/contracts#address-tabs

4.2 Visibility Description

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

StepExLockdropVault			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Public	Can Modify State	-
deposit	External	Can Modify State	nonReentrant whenNotPaused
depositFor	External	Can Modify State	nonReentrant whenNotPaused
withdraw	External	Can Modify State	nonReentrant
migrate	External	Can Modify State	nonReentrant
setMigrator	External	Can Modify State	onlyOwner
pause	External	Can Modify State	onlyOwner
unpause	External	Can Modify State	onlyOwner
_deposit	Internal	Can Modify State	-
_checkLock	Internal	-	-

StepExMigrator			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Public	Can Modify State	-
<Receive Ether>	External	Payable	-
migrate	External	Can Modify State	-

StepExRouter01			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Public	Can Modify State	-
<Receive Ether>	External	Payable	-
_addLiquidity	Private	Can Modify State	-
addLiquidity	External	Can Modify State	ensure
addLiquidityFITFI	External	Payable	ensure
removeLiquidity	Public	Can Modify State	ensure
removeLiquidityFITFI	Public	Can Modify State	ensure
removeLiquidityWithPermit	External	Can Modify State	-
removeLiquidityFITFIWithPermit	External	Can Modify State	-
_swap	Private	Can Modify State	-
swapExactTokensForTokens	External	Can Modify State	ensure
swapTokensForExactTokens	External	Can Modify State	ensure
swapExactFITFIForTokens	External	Payable	ensure

StepExRouter01			
swapTokensForExactFITFI	External	Can Modify State	ensure
swapExactTokensForFITFI	External	Can Modify State	ensure
swapFITFIForExactTokens	External	Payable	ensure
quote	Public	-	-
getAmountOut	Public	-	-
getAmountIn	Public	-	-
getAmountsOut	Public	-	-
getAmountsIn	Public	-	-

StepExRouter			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Public	Can Modify State	-
<Receive Ether>	External	Payable	-
_addLiquidity	Internal	Can Modify State	-
addLiquidity	External	Can Modify State	ensure
addLiquidityFITFI	External	Payable	ensure
removeLiquidity	Public	Can Modify State	ensure
removeLiquidityFITFI	Public	Can Modify State	ensure
removeLiquidityWithPermit	External	Can Modify State	-

StepExRouter			
removeLiquidityFITFIWithPermit	External	Can Modify State	-
removeLiquidityFITFISupportingFeeOnTransferTokens	Public	Can Modify State	ensure
removeLiquidityFITFIWithPermitSupportingFeeOnTransferTokens	External	Can Modify State	-
_swap	Internal	Can Modify State	-
swapExactTokensForTokens	External	Can Modify State	ensure
swapTokensForExactTokens	External	Can Modify State	ensure
swapExactFITFIForTokens	External	Payable	ensure
swapTokensForExactFITFI	External	Can Modify State	ensure
swapExactTokensForFITFI	External	Can Modify State	ensure
swapFITFIForExactTokens	External	Payable	ensure
_swapSupportingFeeOnTransferTokens	Internal	Can Modify State	-
swapExactTokensForTokensSupportingFeeOnTransferTokens	External	Can Modify State	ensure
swapExactFITFIForTokensSupportingFeeOnTransferTokens	External	Payable	ensure
swapExactTokensForFITFISupportingFeeOnTransferTokens	External	Can Modify State	ensure
quote	Public	-	-
getAmountOut	Public	-	-
getAmountIn	Public	-	-

StepExRouter			
getAmountsOut	Public	-	-
getAmountsIn	Public	-	-

StepExERC20			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Public	Can Modify State	-
_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	-

StepExFactory			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Public	Can Modify State	-
pairCodeHash	External	-	-
allPairsLength	External	-	-
createPair	External	Can Modify State	-

StepExFactory			
setFeeTo	External	Can Modify State	-
setFeeToSetter	External	Can Modify State	-

StepExPair			
Function Name	Visibility	Mutability	Modifiers
getReserves	Public	-	-
_safeTransfer	Private	Can Modify State	-
<Constructor>	Public	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

4.3 Vulnerability Summary

[N1] [Low] Swap operation issue

Category: Design Logic Audit

Content

In the StepExRouter contract, users can call swapExactTokensForFITFISupportingFeeOnTransferTokens function to exchange other tokens for WFITFI token. But the amountOut here is equal to the balance of all WFITFI in the contract. This will cause users to get more WFITFI if there is excess WFITFI in the contract.

Code location: stepex-periphery/contracts/StepExRouter.sol#L379-400

```
    function swapExactTokensForFITFISupportingFeeOnTransferTokens(
        uint amountIn,
        uint amountOutMin,
        address[] calldata path,
        address to,
        uint deadline
    )
        external
        virtual
        override
        ensure(deadline)
    {
        require(path[path.length - 1] == WFITFI, 'StepExRouter: INVALID_PATH');
        TransferHelper.safeTransferFrom(
            path[0], msg.sender, StepExLibrary.pairFor(factory, path[0], path[1]),
amountIn
        );
        _swapSupportingFeeOnTransferTokens(path, address(this));
        uint amountOut = IERC20(WFITFI).balanceOf(address(this));
        require(amountOut >= amountOutMin, 'StepExRouter:
INSUFFICIENT_OUTPUT_AMOUNT');
        IWFITFI(WFITFI).withdraw(amountOut);
        TransferHelper.safeTransferETH(to, amountOut);
    }
```

Solution

We recommend to calculate the amountOut by subtracting the balance of WFITFI after the swap operation from the balance before the swap operation.

Status

Ignored; The project team response: We decided to leave it exactly like in the Uniswap V2 and not change anything from the logic. The Uniswap V2 code has been battle-tested by time and billions of TVL.

[N2] [Suggestion] Missing event records

Category: Others

Content

In the StepExFactory contract, The feeToSetter role can set the feeTo and feeToSetter parameters, but there are missing an event records.

Code location: stepex-core/contracts/StepExFactory.sol#L44-52

```
    function setFeeTo(address _feeTo) external {
        require(msg.sender == feeToSetter, 'StepEx: FORBIDDEN');
        feeTo = _feeTo;
    }

    function setFeeToSetter(address _feeToSetter) external {
        require(msg.sender == feeToSetter, 'StepEx: FORBIDDEN');
        feeToSetter = _feeToSetter;
    }
```

Solution

It is recommended to record events when sensitive parameters are modified for subsequent self-inspection or community review.

Status

Ignored; The project team response: We decided to leave it exactly like in the Uniswap V2 and not change anything from the logic. The Uniswap V2 code has been battle-tested by time and billions of TVL.

5 Audit Result

Audit Number	Audit Team	Audit Date	Audit Result
0X002208310004	SlowMist Security Team	2022.08.26 - 2022.08.31	Passed

Summary conclusion: The SlowMist security team uses a manual and SlowMist team's analysis tool to audit the project, during the audit work we found 1 low risk and 1 suggestion vulnerabilities. All findings were ignored.

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