1INCH FIXED RATE SWAP SMART CONTRACT AUDIT

November 29, 2021

MixBytes()

CONTENTS

1. INTRODUCTION

1.1 DISCLAIMER

The audit makes no statements or warranties about utility of the code, safety of the code, suitability of the business model, investment advice, endorsement of the platform or its products, regulatory regime for the business model, or any other statements about fitness of the contracts to purpose, or their bug free status. The audit documentation is for discussion purposes only. The information presented in this report is confidential and privileged. If you are reading this report, you agree to keep it confidential, not to copy, disclose or disseminate without the agreement of 1inch. If you are not the intended recipient(s) of this document, please note that any disclosure, copying or dissemination of its content is strictly forbidden.

1.2 SECURITY ASSESSMENT METHODOLOGY

A group of auditors are involved in the work on the audit who check the provided source code independently of each other in accordance with the methodology described below:

01 Project architecture review:
> Reviewing project documentation
> General code review
> Reverse research and study of the architecture of the code based on the source code only
> Mockup prototyping

Stage goal:
Building an independent view of the project's architecture and identifying logical flaws in the code.

02 Checking the code against the checklist of known vulnerabilities:
> Manual code check for vulnerabilities from the company's internal checklist
> The company's checklist is constantly updated based on the analysis of hacks, research and audit of the clients' code
> Checking with static analyzers (i.e Slither, Mythril, etc.)

Stage goal:
Eliminate typical vulnerabilities (e.g. reentrancy, gas limit, flashloan attacks, etc.)

03 Checking the code for compliance with the desired security model:
> Detailed study of the project documentation
> Examining contracts tests
> Examining comments in code
> Comparison of the desired model obtained during the study with the reversed view obtained during the blind audit
> Exploits PoC development using Brownie

Stage goal:
Detection of inconsistencies with the desired model

04 Consolidation of interim auditor reports into a general one:
> Cross-check: each auditor reviews the reports of the others
> Discussion of the found issues by the auditors
> Formation of a general (merged) report

Stage goal:
Re-check all the problems for relevance and correctness of the threat level and provide the client with an interim report.

05 Bug fixing & re-check:
> Client fixes or comments on every issue
> Upon completion of the bug fixing, the auditors double-check each fix and set the statuses with a link to the fix

Stage goal:
Preparation of the final code version with all the fixes

06 Preparation of the final audit report and delivery to the customer.

Findings discovered during the audit are classified as follows:

FINDINGS SEVERITY BREAKDOWN

Level	Description	Required action
Critical	Bugs leading to assets theft, fund access locking, or any other loss funds to be transferred to any party	Immediate action to fix issue
Major	Bugs that can trigger a contract failure. Further recovery is possible only by manual modification of the contract state or replacement.	Implement fix as soon as possible
Warning	Bugs that can break the intended contract logic or expose it to DoS attacks	Take into consideration and implement fix in certain period
Comment	Other issues and recommendations reported to/acknowledged by the team	Take into consideration

Based on the feedback received from the Customer's team regarding the list of findings discovered by the Contractor, they are assigned the following statuses:

Status	Description
Fixed	Recommended fixes have been made to the project code and no longer affect its security.
Acknowledged	The project team is aware of this finding. Recommendations for this finding are planned to be resolved in the future. This finding does not affect the overall safety of the project.
No issue	Finding does not affect the overall safety of the project and does not violate the logic of its work.

1.3 PROJECT OVERVIEW

1inch is a DeFi aggregator and a decentralized exchange with smart routing. The core protocol connects a large number of decentralized and centralized platforms in order to minimize price slippage and find the optimal trade for the users.In this scope, there is only one smart contract `FixedRateSwap` - this is AMM, which is intended for assets with a stable price for each other, for example, USDC and USDT.
A price curve with a constant sum x + y = const is used. The commission may vary, depending on the balance of tokens.
In most cases, the commission is 1 bip. But when the balance reaches its extreme limits, it either drops to 0 or increases to 20 bip.The following external functions are used:

- `getReturn()` - estimates the return value of the swap
- `deposit()` - makes a deposit of both tokens to the AMM
- `depositFor()` - makes a deposit of both tokens to the AMM and transfers LP tokens to the specified address
- `withdraw()` - makes a proportional withdrawal of both tokens
- `withdrawFor()` - makes a proportional withdrawal of both tokens and transfers them to the specified address
- `withdrawWithRatio()` - makes a withdrawal with custom ratio
- `withdrawForWithRatio()` - makes a withdrawal with custom ratio and transfers tokens to the specified address
- `swap0To1()` - swaps token0 for token1
- `swap1To0()` - swaps token1 for token0
- `swap0To1For()` - swaps token0 for token1 and transfers them to specified receiver address
- `swap1To0For()` - swaps token1 for token0 and transfers them to specified receiver address

1.4 PROJECT DASHBOARD

Client	1inch
Audit name	Fixed Rate Swap
Initial version	0b5a75e9f56e7d21c290dd28c59dc140dcbcc1d5
Final version	52552c22f88ba3e628479bcb34bca541e9812ab6
Date	November 08, 2021 - November 29, 2021
Auditors engaged	3 auditors

FILES LISTING

FixedRateSwap.sol	https://github.com/1inch/fixed-rate-swap/blob/0b5a75e9f56e7d21c290dd28c59dc140dcbcc1d5/contracts/FixedRateSwap.sol

FINDINGS SUMMARY

Level	Amount
Critical	0
Major	1
Warning	1
Comment	1

CONCLUSION

Smart contract has been audited and several suspicious places have been detected. During the audit, no critical problems were found, one issue was marked major, several warnings and comments were identified. After working on the reported results, they all were fixed or confirmed by the client.Final commit identifier with all fixes: `52552c22f88ba3e628479bcb34bca541e9812ab6`

2. FINDINGS REPORT

2.1 CRITICAL

Not Found

2.2 MAJOR

MJR-1	Withdraw "with ratio" without fee via reentrancy
File	FixedRateSwap.sol
Severity	Major
Status	Acknowledged

DESCRIPTION

Tokens with callback e.g. ERC-777 allow easy reentrancy.
Users can exploit reentrancy to withdraw with non-equal proportions without fees when call `withdraw`
after the first transfer of FixedRateSwap.sol#L136 or FixedRateSwap.sol#L321

RECOMMENDATION

We recommend adding nonReentrant modifier to all external functions.

CLIENT'S COMMENTARY

We'll take extra care to not support ERC-777 tokens.

2.3 WARNING

WRN-1	Token exchange blocking
File	FixedRateSwap.sol
Severity	Warning
Status	Acknowledged

DESCRIPTION

There are two tokens on the balance of the contract.
The exchange of token 0 for token 1 is possible when there is token 1 on the contract balance.
You can block the exchange functionality if you constantly make a small balance of one of the tokens.
This check is on the line:
FixedRateSwap.sol#L92

RECOMMENDATION

We recommend adding an event so that you can record and correct such situations from the outside.
Or you can automatically transfer tokens from another contract.

CLIENT'S COMMENTARY

Won't fix.

2.4 COMMENT

CMT-1	Duplicate code
File	FixedRateSwap.sol
Severity	Comment
Status	Fixed at a35978f7

DESCRIPTION

Duplicate code, i.e. using the same code structures in several places. Combining these structures will improve your code. The use of duplicate code structures impairs the perception of the program logic and can easily lead to errors in subsequent code edits. Duplicate code violates SOLID (single responsibility, open-closed, Liskov substitution, interface segregation и dependency inversion) software development principles.
The duplicate code is on the following lines:

- FixedRateSwap.sol#L171-L178
- FixedRateSwap.sol#L212-L220
- FixedRateSwap.sol#L346-L358
- FixedRateSwap.sol#L388-L400

RECOMMENDATION

We recommend grouping duplicate parts of the source code in a private function.

CLIENT'S COMMENTARY

Binsearch deduplication is a bit harder, we'll leave it as is.

3. ABOUT MIXBYTES

MixBytes is a team of blockchain developers, auditors and analysts keen on decentralized systems. We build open-source solutions, smart contracts and blockchain protocols, perform security audits, work on benchmarking and software testing solutions, do research and tech consultancy.

BLOCKCHAINS

- Ethereum
- Cosmos
- EOS
- Substrate

TECH STACK

- Python
- Solidity
- Rust
- C++

CONTACTS

https://github.com/mixbytes/audits_public

https://mixbytes.io/

hello@mixbytes.io

https://t.me/MixBytes

https://twitter.com/mixbytes