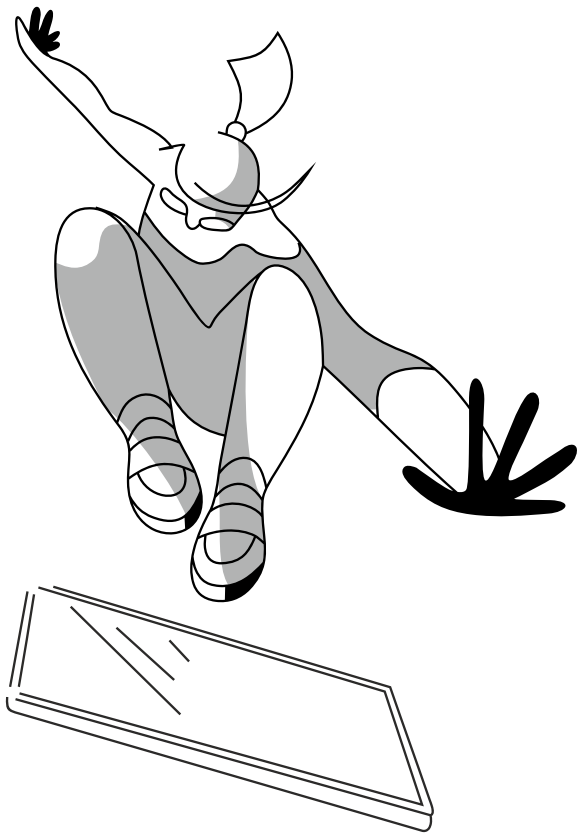
Smart Contract Security Audit Report



By

Igor Gulamov

1inch cross-chain swap

Table of contents

1. Disclaimer

Important to remember:

1. This audit was performed based on the current state of the code at the time of evaluation. Any subsequent changes or modifications to the codebase could render auditors' findings obsolete. Re-audit is recommended post any alterations.
2. While we strive for accuracy, auditors cannot guarantee that all potential vulnerabilities or bugs have been identified. The auditor is not responsible for any overlooked issues.
3. It's always recommended to have multiple layers of checks and balances, including but not limited to, regular code reviews and updated audits.

2. Summary

2.1. Project description

Project: Cross-chain atomic swap

Scope:

- contracts/Escrow.sol
- contracts/EscrowDst.sol
- contracts/EscrowSrc.sol
- contracts/EscrowFactory.sol
- contracts/interfaces/IEscrow.sol
- contracts/interfaces/IEscrowSrc.sol
- contracts/interfaces/IEscrowFactory.sol
- contracts/libraries/Clones.sol
- contracts/libraries/ImmutablesLib.sol
- contracts/libraries/TimelocksLib.sol

Commit: 2e0fafdddb59ba40d8d346b51765910abd3474ed

2.2. Scope

- 1inch/cross-chain-swap/tree/2e0fafdddb59ba40d8d346b51765910abd3474ed

2.3. Conclusion

We consider commit 8dbf51a19dc0b403cd739156941186ebf1e58e14 as a safe version from the informational security point of view.

3. Issue statistics

	Critical	Major	Medium	Minor
Fixed	0	1	1	0
Not Fixed	0	0	0	1

4. Issues

4.1. Critical

No critical issues found

4.2. Major

4.2.1. Write to mem[64:72] in memory-safe assembly block

- 1inch/cross-chain-swap/2e0fa.../contracts/libraries/Clones.sol#L43

Severity: Major
Status: Fixed

If the compiler does not mangle the memory, it does not lead error, because mem[64:96] is a free memory pointer and upper bytes are always zero.

Anyway, this logic is out of memory safety specs and potentially could lead to UB.

We propose using temporary memory according to memory safety specs.

Feedback:

Fixed https://github.com/1inch/cross-chain-swap/pull/59

4.3. Medium

4.3.1. Problem 2038 year vulnerability

- 1inch/cross-chain-swap/2e0fa.../contracts/libraries/TimelocksLib.sol#L32

Severity: Medium
Status: Fixed

TimelockLib is vulnerable to the year 2038 problem. We propose adding comments to the code to inform about this issue or using 32-bit format only for offsets and 64-bit format for timestamps.

Feedback:

Fixed https://github.com/1inch/cross-chain-swap/pull/59

4.4. Minor

4.4.1. Readability

- 1inch/cross-chain-swap/2e0fa.../contracts/libraries/Clones.sol#L33

Severity: Minor
Status: Will not fix

We propose replacing the assembly code with a more readable version, like

```
33 let t := mload(0x40)
34 mstore(add(t, 0x23), 0x5af43d82803e903d91602b57fd5bf3)
35 mstore(add(t, 0x14), implementation)
36 mstore(t, 0x3d602d80600a3d3981f3363d3d373d3d3d363d73)
```

Feedback:

Won't fix. We will use the cloneDeterministic function of the OpenZeppelin library as soon as they release it.