

Security Review Report for 1inch

August 2025

Table of Contents

- 1. About Hexens
- 2. Executive summary
- 3. Security Review Details
 - Security Review Lead
 - Scope
 - Changelog
- 4. Severity Structure
 - Severity characteristics
 - Issue symbolic codes
- 5. Findings Summary
- 6. Weaknesses
 - Redundant conversion from address to Address when encoding

1. About Hexens

Hexens is a pioneering cybersecurity firm dedicated to establishing robust security standards for Web3 infrastructure, driving secure mass adoption through innovative protection technology and frameworks. As an industry elite experts in blockchain security, we deliver comprehensive audit solutions across specialized domains, including infrastructure security, Zero Knowledge Proof, novel cryptography, DeFi protocols, and NFTs.

Our methodology combines industry-standard security practices combined with unique methodology of two teams per audit, continuously advancing the field of Web3 security. This innovative approach has earned us recognition from industry leaders.

Since our founding in 2021, we have built an exceptional portfolio of enterprise clients, including major blockchain ecosystems and Web3 platforms.

2. Executive Summary

This audit covers 1 inch Network Fusion Atomic Swaps, a two-party swap mechanism optimized for EVM-compatible chains, with well-aligned incentives to ensure fair and fast execution for all participants.

Our security assessment spanned four days and involved a thorough review of the Solidity smart contracts intended for deployment on EVM-compatible chains.

During the audit, we did not identify any major vulnerabilities. However, we did identify one optimization.

The reported issue was addressed by the development team and subsequently verified by our team

As a result, we can confidently state that the protocol's security posture and code quality have improved following our audit.

3. Security Review Details

Review Led by

Trung Dinh, Lead Security Researcher

Scope

The analyzed resources are located on:

- https://github.com/1inch/cross-chain-swap/
 - contracts/BaseEscrow.sol
 - contracts/BaseEscrowFactory.sol
 - contracts/Escrow.sol
 - contracts/EscrowDst.sol
 - contracts/EscrowFactory.sol
 - contracts/EscrowSrc.sol
 - contracts/interfaces/IBaseEscrow.sol
 - contracts/interfaces/IEscrowFactory.sol
 - contracts/libraries/ImmutablesLib.sol
 - contracts/mocks/ResolverExample.sol
 - contracts/zkSync/EscrowDstZkSync.sol
 - contracts/zkSync/EscrowFactoryZkSync.sol
 - contracts/zkSync/EscrowSrcZkSync.sol
 - contracts/zkSync/EscrowZkSync.sol
- Commit: d0a59ab2c4b6be5c9769d5775769681873fcf162

The issues described in this report were fixed in the following commit:

- https://github.com/1inch/cross-chain-swap/pull/139
- ★ Commit: 339ab27e5a542737de610e0ebdae6f95dc20cd63

Changelog

11 August 2025 Audit start

13 August 2025 Initial report

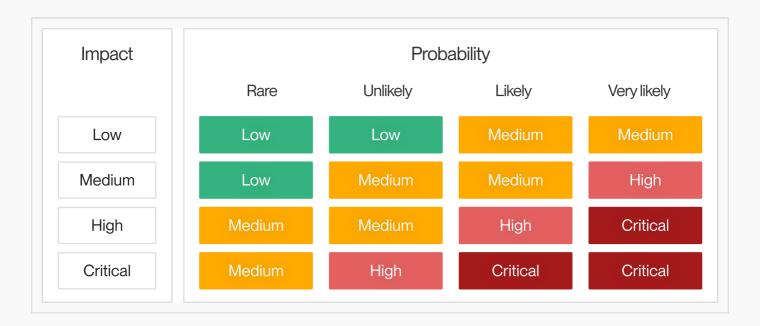
19 August 2025 Revision received

21 August 2025 Final report

4. Severity Structure

The vulnerability severity is calculated based on two components:

- 1. Impact of the vulnerability
- 2. Probability of the vulnerability



Severity Characteristics

Smart contract vulnerabilities can range in severity and impact, and it's important to understand their level of severity in order to prioritize their resolution. Here are the different types of severity levels of smart contract vulnerabilities:

Critical

Vulnerabilities that are highly likely to be exploited and can lead to catastrophic outcomes, such as total loss of protocol funds, unauthorized governance control, or permanent disruption of contract functionality.

High

Vulnerabilities that are likely to be exploited and can cause significant financial losses or severe operational disruptions, such as partial fund theft or temporary asset freezing.

Medium	Vulnerabilitie
	result in mo
	financial imp

Vulnerabilities that may be exploited under specific conditions and result in moderate harm, such as operational disruptions or limited financial impact without direct profit to the attacker.

Low

Vulnerabilities with low exploitation likelihood or minimal impact, affecting usability or efficiency but posing no significant security risk.

Informational

Issues that do not pose an immediate security risk but are relevant to best practices, code quality, or potential optimizations.

Issue Symbolic Codes

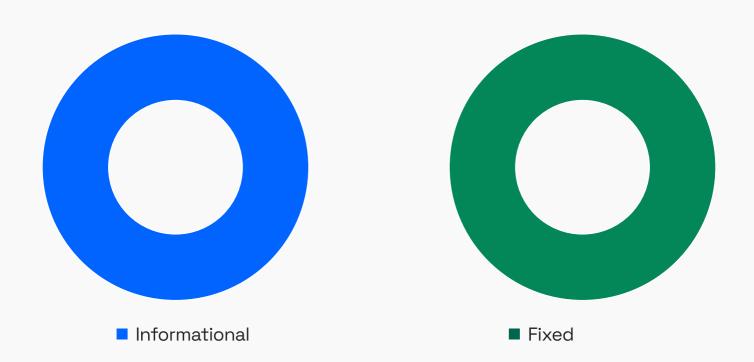
Each identified and validated issue is assigned a unique symbolic code during the security research stage.

Due to the structure of the vulnerability reporting flow, some rejected issues may be missing.

5. Findings Summary

Severity	Number of findings
Critical	0
High	0
Medium	0
Low	0
Informational	1

Total:

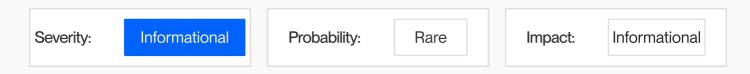


6. Weaknesses

This section contains the list of discovered weaknesses.

OIN11-2 | Redundant conversion from address to Address when encoding





Path:

contracts/BaseEscrowFactory.sol#L141-L146

Description:

In the function **BaseEscrowFactory._postInteraction()**, the **immutablesComplement.parameters** value is computed (line 141 - line 146) as:

```
parameters: abi.encode(
    protocolFeeAmount,
    integratorFeeAmount,
    Address.wrap(uint160(protocolFeeRecipient)),
    Address.wrap(uint160(integratorFeeRecipient))))
```

Here, the address values protocolFeeRecipient and integratorFeeRecipient are explicitly converted to the Address type (an alias for uint256, 32 bytes).

This conversion is unnecessary because **abi.encode()** already pads **address** types to 32 bytes automatically.

Remediation:

Consider modifying line 141 - line 146 to:

```
parameters: abi.encode(
    protocolFeeAmount,
    integratorFeeAmount,

-- Address.wrap(uint160(protocolFeeRecipient)),
-- Address.wrap(uint160(integratorFeeRecipient))
++ protocolFeeRecipient,
++ integratorFeeRecipient
)
```

hexens x inch