HACKEN

SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT

Customer: OneInchExchange
Date: November 4th, 2020

Hacken OÜ
Parda 4, Kesklinn, Tallinn,
10151 Harju Maakond, Eesti,
Kesklinna, Estonia
support@hacken.io

Document

Name	Smart Contract Code Review and Security Analysis Report for 1inch
Approved by	Andrew Matiukhin | CTO Hacken OU
Type	DEX aggregator
Platform	Ethereum / Solidity
Methods	Architecture Review, Functional Testing, Computer-Aided Verification, Manual Review
Repository	https://github.com/CryptoManiacsZone/1inch-contract/
Commit	aa1d1c54546f38b912a24722134ab0c2ae94860d
Deployed contract	https://etherscan.io/address/0x111111125434b319222cdbf8c261674adb56f3ae
Timeline	01 NOV 2020 - 04 NOV 2020
Changelog	04 NOV 2020 - INITIAL AUDIT

Hacken OÜ
Parda 4, Kesklinn, Tallinn,
10151 Harju Maakond, Eesti,
Kesklinna, Estonia
support@hacken.io

Table of contents

Hacken OÜ
Parda 4, Kesklinn, Tallinn,
10151 Harju Maakond, Eesti,
Kesklinna, Estonia
support@hacken.io

Introduction

Hacken OÜ (Consultant) was contracted by One Inch Exchange (Customer) to conduct a Smart Contract Code Review and Security Analysis. This report presents the findings of the security assessment of Customer's smart contract and its code review conducted between November 01st, 2020 - November 04th, 2020.

Scope

The scope of the project is smart contracts in the repository:

```
Contract deployment address:
https://etherscan.io/address/0x111111125434b319222cdbf8c261674adb56f3ae
Repository  https://github.com/CryptoManiacsZone/1inch-contract/
Commit      aa1d1c54546f38b912a24722134ab0c2ae94860d
Files:
      OneInchExchange.sol
      OneInchFlags.sol
      helpers/RevertReasonParser.sol
      helpers/UniERC20.sol
      OneInchCaller.sol
      GasDiscountCalculator.sol
```

We have scanned this smart contract for commonly known and more specific vulnerabilities. Here are some of the commonly known vulnerabilities that are considered:

Category	Check Item
Code review	▪ Reentrancy ▪ Ownership Takeover ▪ Timestamp Dependence ▪ Gas Limit and Loops ▪ DoS with (Unexpected) Throw ▪ DoS with Block Gas Limit ▪ Transaction-Ordering Dependence ▪ Style guide violation ▪ Costly Loop ▪ ERC20 API violation ▪ Unchecked external call ▪ Unchecked math ▪ Unsafe type inference ▪ Implicit visibility level ▪ Deployment Consistency ▪ Repository Consistency ▪ Data Consistency

Hacken OÜ
Parda 4, Kesklinn, Tallinn,
10151 Harju Maakond, Eesti,
Kesklinna, Estonia
support@hacken.io

Functional review	■ Business Logics Review ■ Functionality Checks ■ Access Control & Authorization ■ Escrow manipulation ■ Token Supply manipulation ■ Assets integrity ■ User Balances manipulation ■ Data Consistency manipulation ■ Kill-Switch Mechanism ■ Operation Trails & Event Generation

Executive Summary

According to the assessment, the Customer's smart contracts are secure and can be used in production.

Insecure	Poor secured	Secured	Well-secured

You are here

Our team performed an analysis of code functionality, manual audit, and automated checks with Mythril and Slither. All issues found during automated analysis were manually reviewed, and important vulnerabilities are presented in the Audit overview section. A general overview is presented in AS-IS section, and all found issues can be found in the Audit overview section.

Security engineers found **2** low severity issues during the audit.

Hacken OÜ
Parda 4, Kesklinn, Tallinn,
10151 Harju Maakond, Eesti,
Kesklinna, Estonia
support@hacken.io

Severity Definitions

Risk Level	Description
Critical	Critical vulnerabilities are usually straightforward to exploit and can lead to assets loss or data manipulations.
High	High-level vulnerabilities are difficult to exploit; however, they also have a significant impact on smart contract execution, e.g., public access to crucial functions
Medium	Medium-level vulnerabilities are important to fix; however, they can't lead to assets loss or data manipulations.
Low	Low-level vulnerabilities are mostly related to outdated, unused, etc. code snippets that can't have a significant impact on execution
Lowest / Code Style / Best Practice	Lowest-level vulnerabilities, code style violations, and info statements can't affect smart contract execution and can be ignored.

Hacken OÜ
Parda 4, Kesklinn, Tallinn,
10151 Harju Maakond, Eesti,
Kesklinna, Estonia
support@hacken.io

AS-IS overview

OneInchFlags.sol

Description

OneInchFlags is a library contract used to decode flags used during an exchange process.

GasDiscountCalculator.sol

Description

GasDiscountCalculator is a contract used for calculation of a CHI discount.

RevertReasonParser.sol

Description

RevertReasonParser is a library used to parse error reasons.

UniERC20.sol

Description

UniERC20 is a library used as a wrapper to get balance of a token or ETH.

OneInchCaller.sol

Description

OneInchCaller is a contract used to perform calls to exchanges.

Imports

OneInchCaller contract has following imports:

- IOneInchCaller
- RevertReasonParser
- DodoExtension
- GasDiscountExtension
- PatcherExtension
- SafeERC20Extension
- UniswapV2Extension

Inheritance

OneInchCaller contract is IOneInchCaller, DodoExtension, GasDiscountExtension, PatcherExtension, SafeERC20Extension, UniswapV2Extension.

Usages

OneInchCaller contract has no custom usages.

Structs

OneInchCaller contract has no custom data structures.

Enums

OneInchCaller contract has no custom enums.

Events

OneInchCaller contract has following events:

- event Error(reason);

Modifiers

OneInchCaller has no custom modifiers.

Fields

OneInchCaller contract has no custom fields and constants.

Functions

OneInchCaller has following public functions:

- ***receive***
  Description
  Allows to receive ETH only from contracts.
- ***makeCalls***
  Description
  Make multiple calls.
  Visibility
  external
  Input parameters
  - *CallDescription[] calldata calls* - a list of calls.

  Constraints
  None
  Events emit
  None
  Output
  None
- ***makeCall***

Description
Perform an external call.
Visibility
public
Input parameters
- *CallDescription calldata desc* - a call description

None
Events emit
None
Output
None

OneInchExchange.sol

Description

OneInchExchange is the exchange contract.

Imports

OneInchExchange contract has following imports:

- Ownable - from the OpenZeppelin.
- SafeERC20 - from the OpenZeppelin.
- Pausable - from the OpenZeppelin.
- IChi
- IERC20Permit
- IOneInchCaller
- RevertReasonParser
- UniERC20

Inheritance

OneInchExchange contract is Ownable and Pausable.

Usages

OneInchExchange contract has following usages:

- using *SafeMath for uint256;*
- *using* SafeERC20 *for IERC20;*
- *using UniERC20 for IERC20;*

Structs

OneInchExchange contract has following data structures:

- SwapDescription - contains main swap information.

Enums

OneInchExchange contract has no custom enums.

Events

OneInchExchange contract has following events:

- event Error(reason);
- event Swapped(address indexed sender, IERC20 indexed srcToken, IERC20 indexed dstToken, address dstReceiver, uint256 amount, uint256 spentAmount, uint256 returnAmount, uint256 minReturnAmount, uint256 guaranteedAmount, address referrer);

Modifiers

OneInchExchange has no custom modifiers.

Fields

OneInchExchange contract has following constants:

- uint256 private constant _PARTIAL_FILL = 0x01;
- uint256 private constant _REQUIRES_EXTRA_ETH = 0x02;
- uint256 private constant _SHOULD_CLAIM = 0x04;
- uint256 private constant _BURN_FROM_MSG_SENDER = 0x08;
- uint256 private constant _BURN_FROM_TX_ORIGIN = 0x10;

Functions

OneInchExchange has following public functions:

- *discountedSwap*
  Description
  Performs swap and compensate some gas by burning CHI token.
  Visibility
  external
  Input parameters
  - IOneInchCaller caller - OneInchCaller address.
  - SwapDescription calldata desc - swap description.
  - IOneInchCaller.CallDescription[] calldata calls - a list of calls.

  Constraints
  None
  Events emit
  Emits Swapped or Error events.
  Output
  - *uint256* returnAmount

- *swap*
  Description
  Performs swap.
  Visibility
  external
  Input parameters
  - IOneInchCaller caller - OneInchCaller address.
  - SwapDescription calldata desc - swap description.
  - IOneInchCaller.CallDescription[] calldata calls - a list of calls.

  Constraints
  - The contract should not be paused.
  - Calldata should exist.
  - minReturnAmount should be set.

  Events emit
  Emits Swapped event.
  Output
  - *uint256* returnAmount

- *rescueFunds*
  Description
  Send accidentally locked tokens or ETH to a message sender. The function is safe because *OneInchExchange* is not supposed to store any funds for exchange process.
  Visibility
  external
  Input parameters
  - IERC20 token - token address to withdraw.
  - uint256 amount - amount to transfer.

  Constraints
  - Can only be called by the contract owner.

  Events emit
  None
  Output
  None

- *pause*
  Description
  Pauses the contract
  Visibility
  external
  Input parameters
  None
  Constraints
  - Can only be called by the contract owner.

  Events emit
  None
  Output

Hacken OÜ
Parda 4, Kesklinn, Tallinn,
10151 Harju Maakond, Eesti,
Kesklinna, Estonia
support@hacken.io

None

Hacken OÜ
Parda 4, Kesklinn, Tallinn,
10151 Harju Maakond, Eesti,
Kesklinna, Estonia
support@hacken.io

Audit overview

Critical

No critical issues were found.

High

No high severity issues were found.

Medium

No medium severity issues were found.

Low

1. OneInchFlags contract is never used.
2. GasDiscountCalculator contract is never used.

Lowest / Code style / Best Practice

No lowest severity issues were found.

Hacken OÜ
Parda 4, Kesklinn, Tallinn,
10151 Harju Maakond, Eesti,
Kesklinna, Estonia
support@hacken.io

Conclusion

Smart contracts within the scope were manually reviewed and analyzed with static analysis tools. For the contract, high-level description of functionality was presented in As-Is overview section of the report.

Audit report contains all found security vulnerabilities and other issues in the reviewed code.

Smart contracts within the scope were manually reviewed and analyzed with static analysis tools. For the contract, high-level description of functionality was presented in As-Is overview section of the report.

Audit report contains all found security vulnerabilities and other issues in the reviewed code.

Security engineers found **2** low severity issues during the audit.

Violations in the following categories were found and addressed to Customer:

Category	Check Item	Comments
Code review	■ Unused code	■ Unused code can be found in the repository.

Hacken OÜ
Parda 4, Kesklinn, Tallinn,
10151 Harju Maakond, Eesti,
Kesklinna, Estonia
support@hacken.io

Disclaimers

Hacken Disclaimer

The smart contracts given for audit have been analyzed in accordance with the best industry practices at the date of this report, in relation to cybersecurity vulnerabilities and issues in smart contract source code, the details of which are disclosed in this report (Source Code); the Source Code compilation, deployment, and functionality (performing the intended functions).

The audit makes no statements or warranties on security of the code. It also cannot be considered as a sufficient assessment regarding the utility and safety of the code, bugfree status or any other statements of the contract. While we have done our best in conducting the analysis and producing this report, it is important to note that you should not rely on this report only - we recommend proceeding with several independent audits and a public bug bounty program to ensure security of smart contracts.

Technical Disclaimer

Smart contracts are deployed and executed on blockchain platform. The platform, its programming language, and other software related to the smart contract can have its vulnerabilities that can lead to hacks. Thus, the audit can't guarantee the explicit security of the audited smart contracts.