SMART CONTRACT AUDIT

zokyo

November 29th 2022 | v. 1.0

Security Audit Score

PASS

Zokyo Security has concluded that this smart contract passes security qualifications to be listed on digital asset exchanges.

TECHNICAL SUMMARY

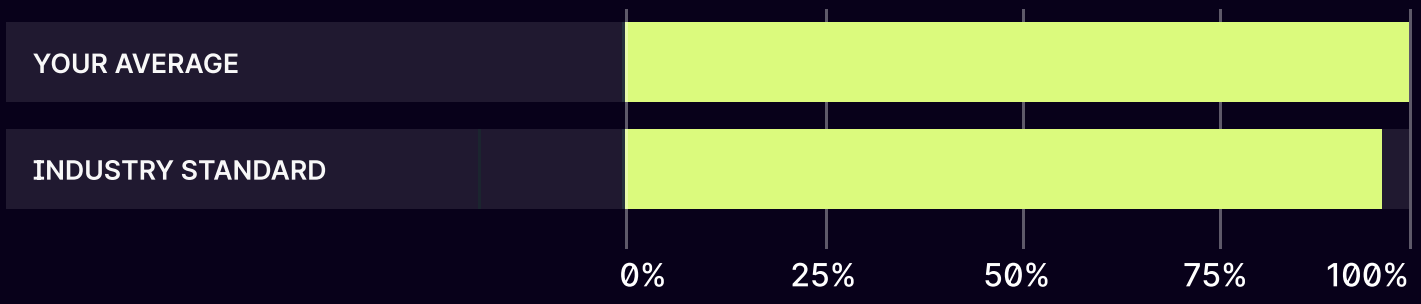
This document outlines the overall security of the IPOR smart contracts evaluated by the Zokyo Security team.

The scope of this audit was to analyze and document the IPOR smart contract codebase for quality, security, and correctness.

Contract Status

There were no critical issues found during the audit. (See Complete Analysis)

Testable Code



The testable code is 100%, which is above the industry standard of 95%. (See Complete Analysis)
It should be noted that this audit is not an endorsement of the reliability or effectiveness of the contract but rather limited to an assessment of the logic and implementation. In order to ensure a secure contract that can withstand the Ethereum network's fast-paced and rapidly changing environment, we recommend that the IPOR team put in place a bug bounty program to encourage further active analysis of the smart contract.

Table of Contents

AUDITING STRATEGY AND TECHNIQUES APPLIED

The source code of the smart contract was taken from the IPOR repository:
https://github.com/IPOR-Labs/ipor-protocol

Last commit: a1a36578574afe99ae51e6d56c93960c9bc1a350

Within the scope of this audit, the team of auditors reviewed the following contract(s):

- IporToken.sol

During the audit, Zokyo Security ensured that the contract:

- Implements and adheres to the existing standards appropriately and effectively;
- The documentation and code comments match the logic and behavior;
- Distributes tokens in a manner that matches calculations;
- Follows best practices, efficiently using resources without unnecessary waste;
- Uses methods safe from reentrance attacks;
- Is not affected by the most resent vulnerabilities;
- Meets best practices in code readability, etc.

Zokyo Security has followed best practices and industry-standard techniques to verify the implementation of IPOR smart contracts. To do so, the code was reviewed line by line by our smart contract developers, who documented even minor issues as they were discovered. Part of this work includes writing a test suite using the Hardhat testing framework. In summary, our strategies consist largely of manual collaboration between multiple team members at each stage of the review:

01	Due diligence in assessing the overall code quality of the codebase.	03	Testing contract logic against common and uncommon attack vectors.
02	Cross-comparison with other, similar smart contracts by industry leaders.	04	Thorough, manual review of the codebase, line-by-line.

Executive Summary

Zokyo auditing team has run a deep investigation of IPOR's smart contract. During the auditing process, there were no issues found. The contract is in excellent condition, well written and structured.

Based on the conducted audit, we give a score of 100 to the aforementioned contract. Zokyo auditing team can state that the contract is fully production ready and bear no security or operational risk.

STRUCTURE AND ORGANIZATION OF DOCUMENT

For the ease of navigation, sections are arranged from the most to the least critical one. Issues are tagged as “Resolved” or “Unresolved” depending on whether they have been fixed or addressed. The issues that are tagged as “Verified” contain unclear or suspicious functionality that either needs explanation from the Customer’s side or remains disregarded by the Customer. Furthermore, the severity of each issue is written as assessed by the risk of exploitation or other unexpected or otherwise unsafe behavior:

Critical

The issue affects the contract in such a way that funds may be lost, allocated incorrectly, or otherwise result in a significant loss.

High

The issue affects the ability of the contract to compile or operate in a significant way.

Medium

The issue affects the ability of the contract to operate in a way that doesn’t significantly hinder its behavior.

Low

The issue has minimal impact on the contract’s ability to operate.

Informational

The issue has no impact on the contract’s ability to operate.

COMPLETE ANALYSIS

During the auditing process (both manual part and testing part) no issues were identified.

	IporToken.sol
Re-entrancy	Pass
Access Management Hierarchy	Pass
Arithmetic Over/Under Flows	Pass
Unexpected Ether	Pass
Delegatecall	Pass
Default Public Visibility	Pass
Hidden Malicious Code	Pass
Entropy Illusion (Lack of Randomness)	Pass
External Contract Referencing	Pass
Short Address/ Parameter Attack	Pass
Unchecked CALL Return Values	Pass
Race Conditions / Front Running	Pass
General Denial Of Service (DOS)	Pass
Uninitialized Storage Pointers	Pass
Floating Points and Precision	Pass
Tx.Origin Authentication	Pass
Signatures Replay	Pass
Pool Asset Security (backdoors in the underlying ERC-20)	Pass

CODE COVERAGE AND TEST RESULTS FOR ALL FILES

Tests written by Zokyo Security

As a part of our work assisting IPOR in verifying the correctness of their contract code, our team was responsible for writing integration tests using the Hardhat testing framework.

The tests were based on the functionality of the code, as well as a review of the IPOR contract requirements for details about issuance amounts and how the system handles these.

IporToken
✓ Should be deployed correctly (153ms)
✓ The total supply is correctly initialized.
✓ Owner's balance is correctly initialized.
✓ User's balance is correctly initialized.
✓ Attacker's balance is correctly initialized.
✓ The total supply is the user and owner balance.
✓ The address 0x0 should not receive tokens.
✓ Allowance can be changed. (58ms)
✓ Balance of one user must be less or equal to the total supply. (49ms)
✓ Balance of the crytic users must be less or equal to the total supply.
✓ Using transfer to send tokens to the address 0x0 will revert. (250ms)
✓ Using transferFrom to send tokens to the address 0x0 will revert. (52ms)
✓ Self transfering tokens using transferFrom works as expected. (59ms)
✓ Transfering tokens to other address using transferFrom works as expected. (67ms)
✓ Self transfering tokens using transfer works as expected. (50ms)
✓ Transfering tokens to other address using transfer works as expected. (49ms)
✓ Transfering more tokens than the balance will revert. (52ms)

17 passing (1s)

The resulting code coverage (i.e., the ratio of tests-to-code) is as follows:

FILE	% STMTS	% BRANCH	% FUNCS	% LINES	% Uncovered Lines
IporToken.sol	100	100	100	100	
All files	**100**	**100**	**100**	**100**	

We are grateful for the opportunity to work with the IPOR team.

The statements made in this document should not be interpreted as an investment or legal advice, nor should its authors be held accountable for the decisions made based on them.

Zokyo Security recommends the IPOR team put in place a bug bounty program to encourage further analysis of the smart contract by third parties.

zokyo