Smart Contract Audit Report

Audit conducted on the iHelp Smart Contract System

Smart Contract	Smart Contract Code Review and Security Analysis Report for iHelp.
Type Of Utility	Staking Pools; Dividend Rewards System; Charity Donations
Platform	Ethereum Virtual Machine
Language	Solidity
Code Repository	https://github.com/iHelp-Finance/ihelp-contracts
Time Of Audit	commit/9a6ab8bc24bd3ec181ec98eda8e826f9386a99f4

15.04.2022

Scope of the audit

This Audit Report mainly focuses on the overall security of the IHelp Smart Contract System. This audit was conducted with rigorous attention to the general implementation of the contract and by examining the overall architectural layout of the software implementation. The reliability and correctness of this smart contract's codebase are being assessed.

Security Scope

Identifying security related issues within each contract and the system of contract.

General Code Quality

A full assessment of the code quality and general software architecture patterns and best practices used.

Auditing Methods Used

Rigorous testing of the project has been performed. Detailed code base analysis was conducted, reviewing the smart contract architecture to ensure it is structured and safe.

A detailed, line by line inspection of the codebase was conducted to find any potential security vulnerabilities such as denial of service attacks, race conditions, transaction-ordering dependence, timestamp dependence, and denial of service attacks.

Automated and manual testing was employed that included:
– Analysis of on-chain data security
– Analysis of the code in-depth and detailed, manual review of the code, line-by-line.
– Deployment of the code on an in-house testnet blockchain and running live tests●
– Determining failure preparations and if worst-case scenario protocols are in place
– Analysis of any third-party code use and verifying the overall security of this

Tools Used:
Remix IDE, Ganache, Solhint, VScode, Mythril, Hardhat

Assessing Possible Issues

Any issue detected during the conduction of this audit will be categorized under one of 3 severity levels: low, medium and high.

Low level Severity Issues

Issues that do not pose any serious threat to the functionality of the software

Medium level Severity issues

Issues that can cause potential problems to the overall health of the software application but that can be fixed without having any breaking changes on the current functionality

High level Severity issues

Critical issues that affect the smart contract’s overall performance and functionality. These issues should be fixed urgently.

Code Base General Issues Report

General issues that were found during manual and automatic assessments

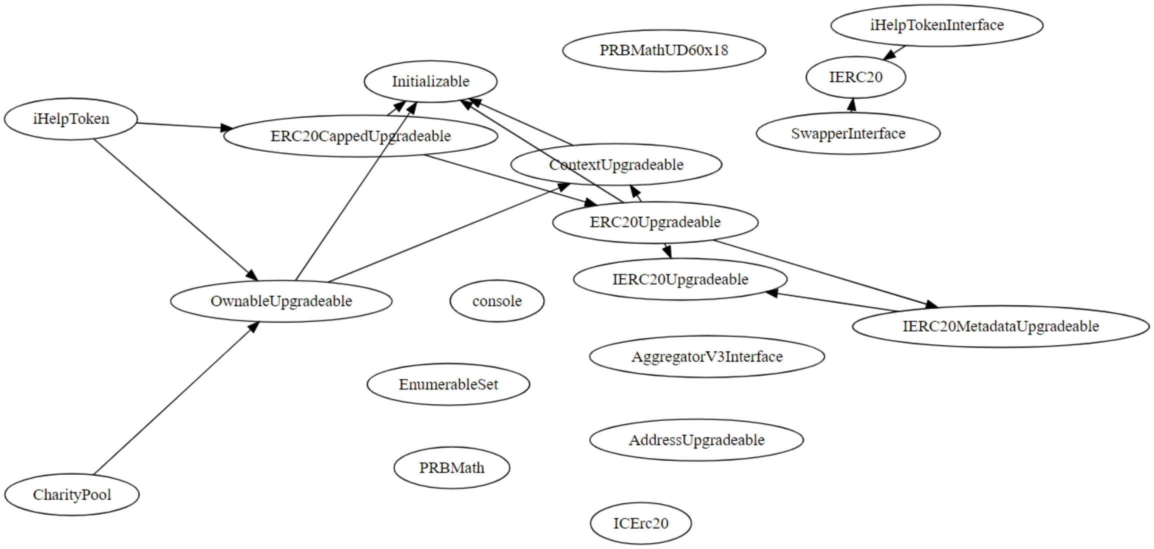
No	Issue Verification	Status
1	Compiler warnings	Passed
2	Reentrancy and Race Conditions.	Passed
3	Possible delays in data delivery.	Passed
4	Oracle calls.	Passed
5	DoS with block gas limit.	Passed
6	DoS with Revert.	Passed
7	Timestamp dependence.	Passed
8	Methods execution permissions.	Passed
9	Economy model.	Passed
10	Exchange impact rate on the logic.	Passed
11	Private user data leaks.	Passed
12	Scoping and Declarations.	Passed
13	Arithmetic accuracy.	Passed

Issues Found

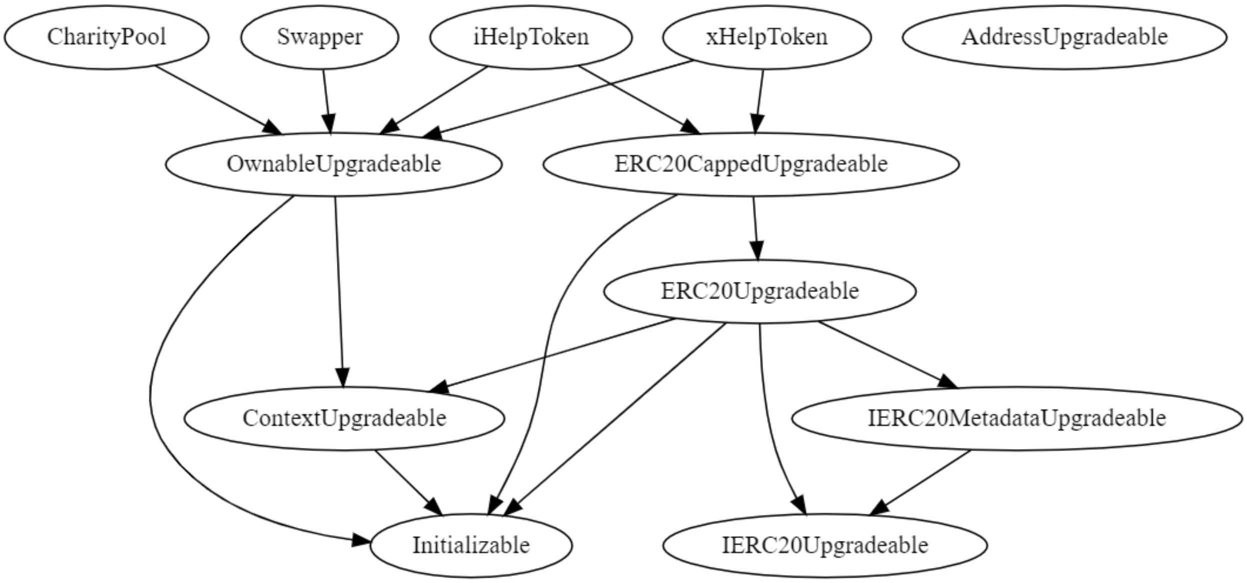
Low Level Severity	Medium Level Severity	HighLevel Severity
0	0	0

Contract Dependency Graphs

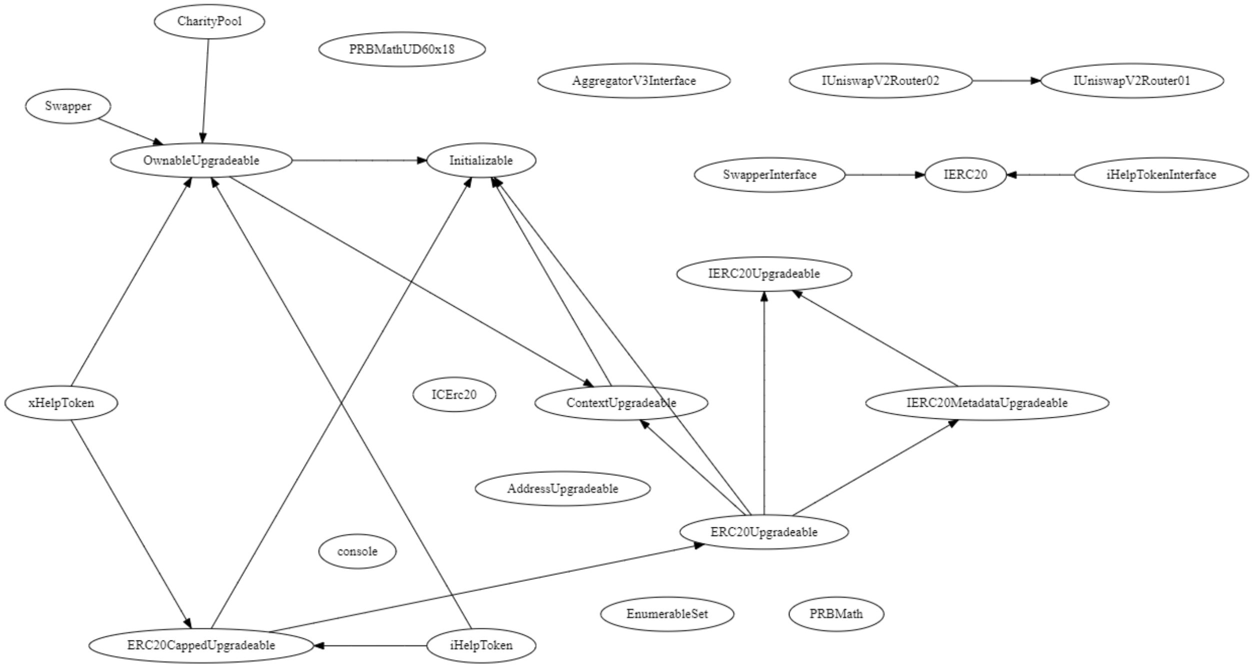
iHelp.sol



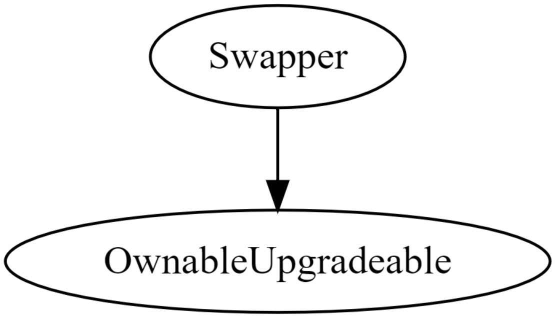
CharityPool.sol



xHelp.sol



Swapper.sol



Manual Code Inspection

The code of the target contract and its dependencies was reviewed, deployed and manually tested by our developers.

No	Contract	Issues
1	iHelpToken.sol	2
2	xHelpToken.sol	0
3	CharityPool.sol	0
4	Swapper.sol	0

Issues Found

Low Level Severity	Medium Level Severity	HighLevel Severity
2	0	0

Inspections

Contract: iHelpToken.sol
Address: TBA
Issues: 2
Notes: ERC-20 Token Implementation

1.Gas Optimization Potential
Code Line: 246
Severity: Low
Method: `dripStage1() external onlyOperatorOrOwner`

Loops can lead to large transactional costs if left unchecked. We recommend switching to a state-based implementation of the logic if possible.

2.Gas Optimization Potential
Code Line: 407
Severity: Low
Method: `distribute(uint256 tokensToCirculate) internal returns (bool)`

Loops can lead to large transactional costs if left unchecked. We recommend switching to a state-based implementation of the logic if possible.

Access Control And Privileges

iHelp.sol

Role	Methods
Owner	setProcessiongState transferOperator setTokenPhase registerCharityPool deregisterCharityPool dripStage1, dripStage2, dripStage3, dripStage4 setProcessingGasLimit
Operator	setProcessiongState transferOperator setTokenPhase registerCharityPool deregisterCharityPool dripStage1, dripStage2, dripStage3, dripStage4 setProcessingGasLimit

Notes

The identified roles do not present any security-related risk at the time this audit was conducted/ Due to the Upgradeable nature of the contract, the developers may change the implementation without notice.

xHelp.sol

Role	**Methods**
Owner	This role does not hold any special privileges over the contract.

Notes

The identified roles do not present any security-related risk at the time this audit was conducted. Due to the Upgradable nature of the contract, the developers may change the implementation without notice.

CharityPool.sol

Role	**Methods**
Owner	transferOperator, postUpgrade, setStakingPool
Operator	transferOperator, postUpgrade, setStakingPool
helpToken	redeemInterest

Notes

The identified roles do not present any security-related risk at the time this audit was conducted. Due to the Upgradable nature of the contract, the developers may change the implementation without notice.

Swapper.sol

Role	**Methods**
Owner	setRouter

Notes

The identified roles do not present any security-related risk at the time this audit was conducted. Due to the Upgradable nature of the contract, the developers may change the implementation without notice

Conclusion

The **IHelp** Smart contracts do not contain any high severity security issues!

Audit Score

Section	**Score**
Codebase Security	100%
Codebase Complexity and Practices	100%
Owner Privileges And Control	98%
Overall Score	**99%**

Disclaimer

This is a limited report on our findings based on our analysis, in accordance with good industry practice as at the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, the details of which are set out in this report. In order to get a full view of our analysis, it is crucial for you to read the full report. While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against us on the basis of what it says or doesn't say, or how we produced it, and it is important for you to conduct your own independent investigations before making any decisions. We go into more detail on this in the disclaimer below – please make sure to read it in full.

DISCLAIMER: By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis, and does not constitute investment advice. No one shall have any right to rely on the report or its contents, and Speakeasy and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (Speakeasy) owe no duty of care towards you or any other person, nor does Speakeasy make any warranty or representation to any person on the accuracy or completeness of the report. The report is provided "as is", without any conditions, warranties or other terms of any kind except as set out in this disclaimer, and Speakeasy hereby excludes all representations, warranties, conditions and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Except and only to the extent that it is prohibited by law, Speakeasy hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against BlockAudit, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and
whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of the use of this report, and any reliance on this report.
The analysis of the security is purely based on the smart contracts alone. No applications or operations were reviewed for security. No product code has been reviewed.