AUDIT.SC

Smart Contract Audit

BRING TRUST IN YOUR PROJECT

AUDIT-SC PARTNER

Halcyon

WWW.AUDIT.SC

2021

FULL SMART CONTRACT AUDIT

SOLIDITY CHECK

Audit SC Guarantees that every smart contract that has been audited has gone through both automated Smart Contract Scanner Softwares and is manually verified by one of our highly experienced smart contract experts.

Table of Contents

AUDIT-SC

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 below 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 AUDIT-SC and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (AUDIT-SC) owe no duty of care towards you or any other person, nor does AUDIT-SC 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 AUDIT-SC 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, AUDIT-SC hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against AUDIT-SC, 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 use of this report, and any reliance on this report.

OVERVIEW

PROJECT SUMMARY

Project Name	**Halcyon**
Platform	N/A
Language	Solidity

AUDIT SUMMARY

Date	14-11-2021
Audit Type	Static Analysis, Manual Review
Audit Result	**Passed**

RISK SUMMARY

Risk Level	Total	Found	Pending	Solved	Acknowledgde	Objected
Critical	0	0	0	0	0	0
Major	0	0	0	0	0	0
Medium	0	0	0	0	0	0
Minor	1	0	1	0	0	0
Informative	0	0	0	0	0	0
Discussion	0	0	0	0	0	0

FINDINGS

Centralization Risk

Description:
The owner of the contract has sole power over the following functions:

- setAirDrop()
- unsetAirDrop()
- switchAirDrop()
- setLiquifyRate()
- setReflectionRate()
- setBurnRate()
- setMaxTransferRate()
- setRouter()
- swapAndLiquify()
- sendAirDrops()

Without obtaining external consensus (of the holders or community).

Recommendation:

In order to mitigate the security risks involved with potential centralization, we recommend that the owner account's power is distributed across multiple roles, or it's privileges being part of a decentralized protocol to improve the project's security. In case the client choses to maintain the current distribution of privilege, we recommend the private key being stored in a secure place, and security enhanced to multi-signature wallets.

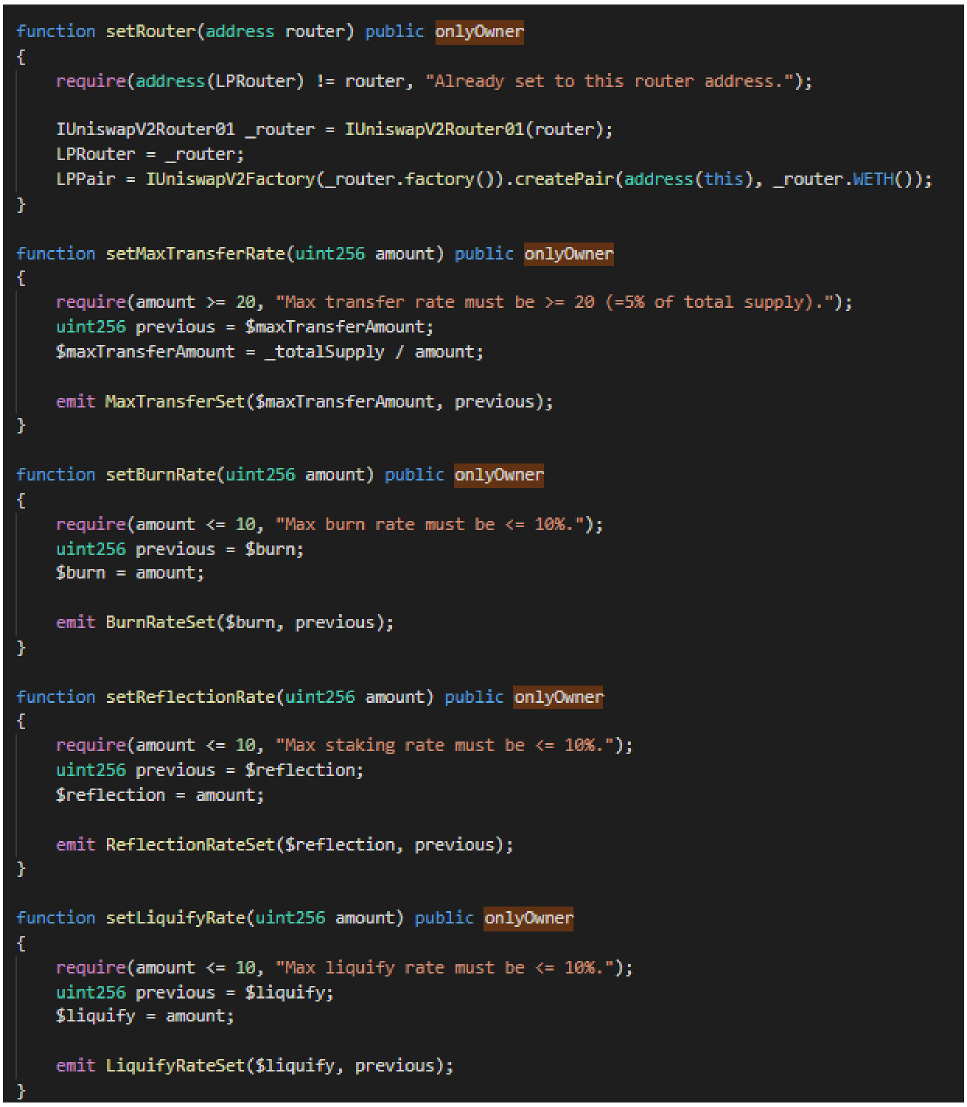
A further improvement on the fairness and awareness of the privileged protocols could be made by adding a mandatory latency on privileged functions. This way, the community has reasonable time to respond and adjust to centralized changes.

Category	Risk Level	Number of Findings	Status
Centralization	Minor	**1**	Pending

AUDIT DETAILS

Centralization Risk

Below is a quick snapshot of a part of the privileged functions. Our recommendation is to alleviate this in part with the aforementioned substitutes or extensions



```
function setRouter(address router) public onlyOwner
{
    require(address(LPRouter) != router, "Already set to this router address.");

    IUniswapV2Router01 _router = IUniswapV2Router01(router);
    LPRouter = _router;
    LPPair = IUniswapV2Factory(_router.factory()).createPair(address(this), _router.WETH());
}

function setMaxTransferRate(uint256 amount) public onlyOwner
{
    require(amount >= 20, "Max transfer rate must be >= 20 (=5% of total supply).");
    uint256 previous = $maxTransferAmount;
    $maxTransferAmount = _totalSupply / amount;

    emit MaxTransferSet($maxTransferAmount, previous);
}

function setBurnRate(uint256 amount) public onlyOwner
{
    require(amount <= 10, "Max burn rate must be <= 10%.");
    uint256 previous = $burn;
    $burn = amount;

    emit BurnRateSet($burn, previous);
}

function setReflectionRate(uint256 amount) public onlyOwner
{
    require(amount <= 10, "Max staking rate must be <= 10%.");
    uint256 previous = $reflection;
    $reflection = amount;

    emit ReflectionRateSet($reflection, previous);
}

function setLiquifyRate(uint256 amount) public onlyOwner
{
    require(amount <= 10, "Max liquify rate must be <= 10%.");
    uint256 previous = $liquify;
    $liquify = amount;

    emit LiquifyRateSet($liquify, previous);
}
```

AUDIT RESULT

Basic Coding Bugs

1. Constructor Mismatch

o Description: Whether the contract name and its constructor are not identical to each other.

o Result: PASSED

o Severity: Critical

Ownership Takeover

o Description: Whether the set owner function is not protected.

o Result: PASSED

o Severity: Critical

Redundant Fallback Function

o Description: Whether the contract has a redundant fallback function.

o Result: PASSED

o Severity: Critical

Overflows & Underflows

Description: Whether the contract has general overflow or underflow Vulnerabilities

o Result: PASSED

o Severity: Critical

Reentrancy

o Description: Reentrancy is an issue when code can call back into your contract and change state, such as withdrawing ETHs.

o Result: PASSED

o Severity: Critical

MONEY-Giving Bug

o Description: Whether the contract returns funds to an arbitrary address.

o Result: PASSED

o Severity: High

Blackhole

o Description: Whether the contract locks ETH indefinitely: merely in without out.

o Result: PASSED

o Severity: High

Unauthorized Self-Destruct

o Description: Whether the contract can be killed by any arbitrary address.

o Result: PASSED

o Severity: Medium

Revert DoS

o Description: Whether the contractis vulnerable to DoSattack because of unexpected revert.

o Result: PASSED

o Severity: Medium

Unchecked External Call

o Description: Whether the contract has any external call without checking the return value.

o Result: PASSED

o Severity: Medium

Gasless Send

o Description: Whether the contractis vulnerable to gasless send.

o Result: PASSED

o Severity: Medium

Send Instead of Transfer

o Description: Whether the contract uses send instead of transfer.

o Result: PASSED

o Severity: Medium

Costly Loop

o Description: Whether the contract has any costly loop which may lead to Out-Of-Gas exception.

o Result: PASSED

o Severity: Medium

(Unsafe) Use of Untrusted Libraries

o Description: Whether the contract use any suspicious libraries.

o Result: PASSED

o Severity: Medium

(Unsafe) Use of Predictable Variables

o Description: Whether the contract contains any randomness variable, but its value can be predicated.

o Result: PASSED

o Severity: Medium

Transaction Ordering Dependence

o Description: Whether the final state of the contract depends on the order of the transactions.

o Result: PASSED

o Severity: Medium

. Deprecated Uses

o Description: Whether the contract use the deprecated tx.origin to perform the authorization.

o Result: PASSED

o Severity: Medium

AUDIT.SC

CONTACTUS

Audit SC Guarantees that every smart contract that has been audited has gone through both automated Smart Contract Scanner Softwares and is manually verified by

E-mail:	info@audit.sc
Website:	www.audit.sc

AUDIT.SC