



Smart Contract Security Audit Report

OryxFI

June 2022

Security Status



www.hacksafe.io



Audit Details



Audited project

Oryx



Deployer address

0x0f6B28676626C43Fb4A9E59e7C46c52Fa0eB717F



Client contacts

ORYXFI team



Blockchain

Binance Smart Chain



Website

<https://oryxfi.io/>

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 TechRate and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (HackSafe) owe no duty of care towards you or any other person, nor does HackSafe 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 HackSafe 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, HackSafe hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against HackSafe, 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.

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.

Procedure

Step 1 - In-Depth Manual Review

Manual line-by-line code reviews to ensure the logic behind each function is sound and safe from various attack vectors. This is the most important and lengthy portion of the audit process (as automated tools often cannot find the nuances that lead to exploits such as flash loan attacks).

Step 2 - Automated Testing

Simulation of a variety of interactions with your Smart Contract on a test blockchain leveraging a combination of automated test tools and manual testing to determine if any security vulnerabilities exist.

Step 3 – Leadership Review

The engineers assigned to the audit will schedule meetings with our leadership team to review the contracts, any comments or findings, and ask questions to further apply adversarial thinking to discuss less common attack vectors.

Step 4 - Resolution of Issues

Consulting with the team to provide our recommendations to ensure the code's security and optimize its gas efficiency, if possible. We assist project team's in resolving any outstanding issues or implementing our recommendations.

Step 5 - Published Audit Report

Boiling down results and findings into an easy-to-read report tailored to the project. Our audit reports highlight resolved issues and any risks that exist to the project or its users, along with any remaining suggested remediation measures. Diagrams are included at the end of each report to help users understand the interactions which occur within the project.

Background

HackSafe was commissioned by ORYXFI token to perform an audit of smart contracts:

- <https://bscscan.com/address/0x10bb58010cb58e7249099ef2efdffe342928b639#code>

The purpose of the audit was to achieve the

- Ensure that the smart contract functions as intended.
- Identify potential security issues with the smart contract.

The information in this report should be understood the risk exposure of the smart contract, and as a guide to improve the security posture of the smart contract by remediating the issues that were identified.

Contract Details

Token contract details for 27.06.2022

Token Type	: ERC20
Contract name	: ORYX
Contract address	: 0x10bb58010CB58e7249099eF2efdFfe342928B639
Compiler version	: v0.8.7+commit.e28d00a7
Total supply	: 10,000,000
Token Ticker	: ORYX
Decimals	: 18
Token Holders	: 1,771
Top 100 token holder's dominance	: 89.06%
Transactions count	: 48,730
Contract deployer address	: 0x0f6B28676626C43Fb4A9E59e7C46c52Fa0eB717F
Owner address	: 0x0f6B28676626C43Fb4A9E59e7C46c52Fa0eB717F

Claimed Smart Contract Features

Claimed Feature Detail	Our Observation
<p>Tokenomics :</p> <ul style="list-style-type: none">• Name : Oryx• Symbol : ORYX• Decimals : 18• Protocol : ERC20• Max Total supply : 10,000,000	<p>Yes, This is valid.</p>

Audit Summary

According to the standard audit assessment, Customer`s solidity smart contracts are “Secure”. This token contract does contain owner control, which do not make it fully decentralized as owner does have control over smart contract.

Insecure	Poor secured	Secure	Well-secured
----------	--------------	--------	--------------

You are here



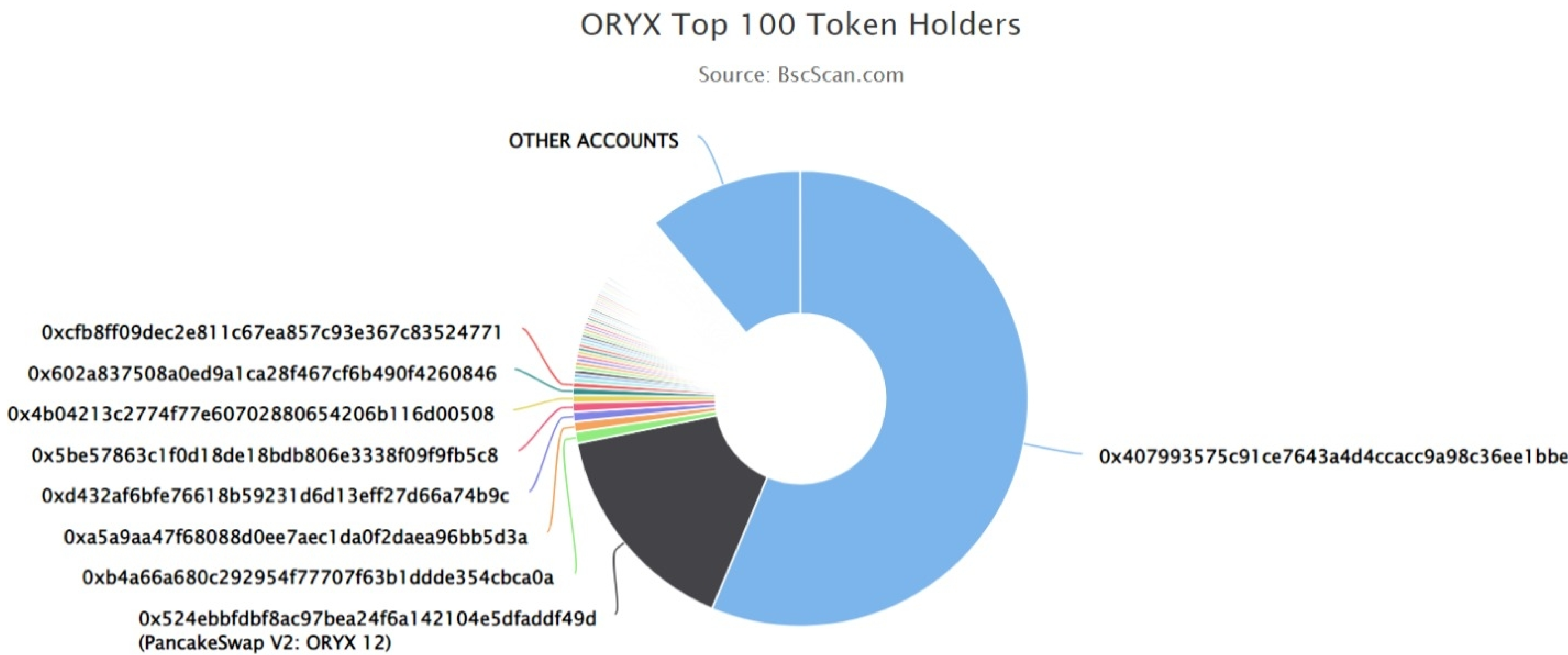
We used various tools like Slither, Mythril and Remix IDE. At the same time this finding is based on critical analysis of the manual audit. All issues found during analysis were manually reviewed and applicable vulnerabilities are presented in the issues checking status.

We found 0 critical, 0 high, 0 medium and 1 low and some very low-level issues. These issues are not critical ones.

OryxFi Distribution



💡 The top 100 holders collectively own 88.99% (8,898,611.35 Tokens) of ORYX

💡 Token Total Supply: 10,000,000.00 Token | Total Token Holders: 1,774



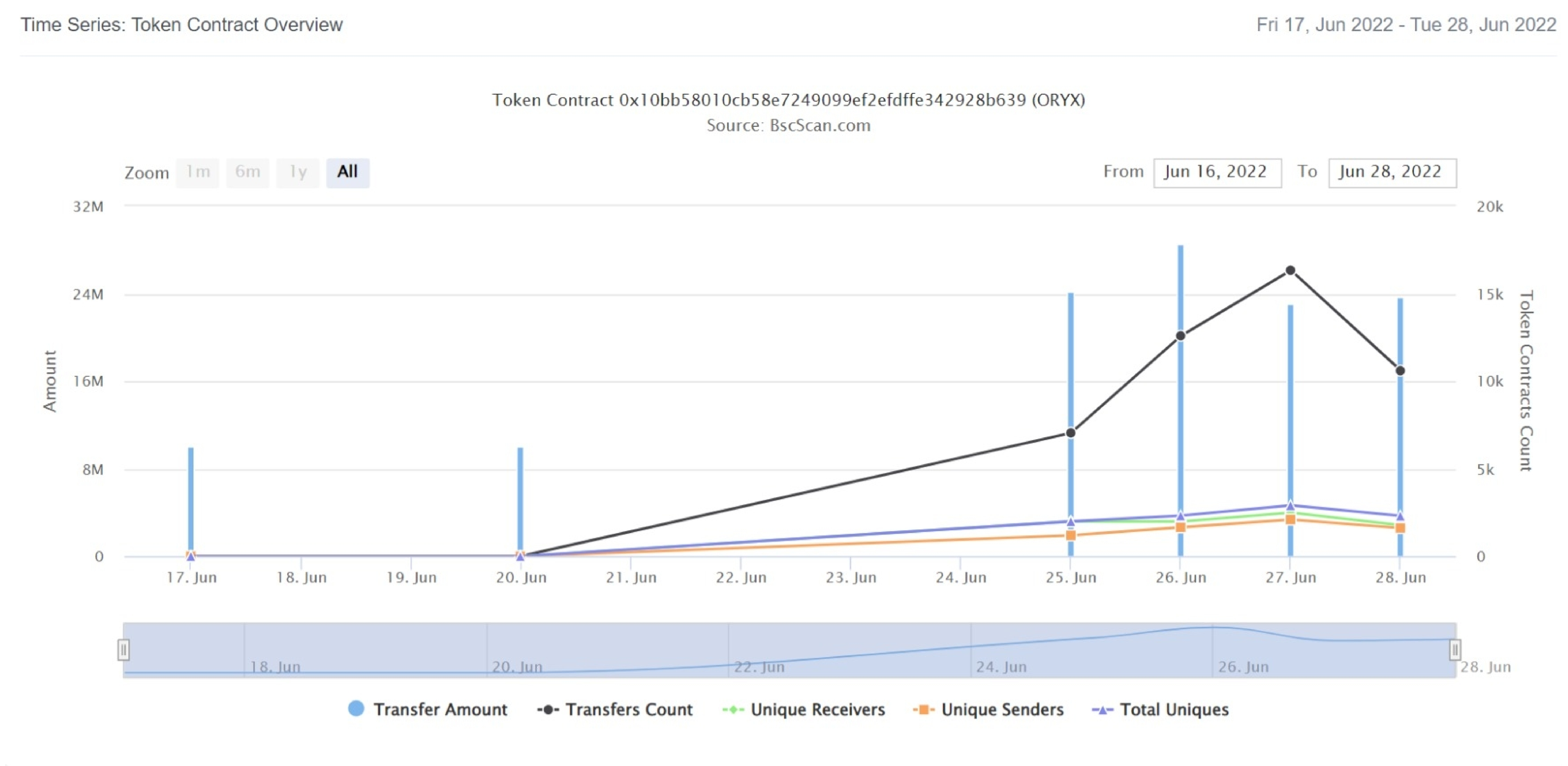
OryxFi Top 20 Token Holders

(A total of 8,898,611.35 tokens held by the top 100 accounts from the total supply of 10,000,000.00 token)

Rank	Address	Quantity (Token)	Percentage
1	 0x407993575c91ce7643a4d4ccacc9a98c36ee1bbe	5,634,800	56.3480%
2	 PancakeSwap V2: ORYX 12	1,550,331.532115957748334546	15.5033%
3	0xb4a66a680c292954f77707f63b1ddde354cbca0a	81,246.0622066485217373	0.8125%
4	0xa5a9aa47f68088d0ee7aec1da0f2daea96bb5d3a	72,086.936280553563332454	0.7209%
5	0xd432af6bfe76618b59231d6d13eff27d66a74b9c	68,642.347728172676624218	0.6864%
6	0x5be57863c1f0d18de18bdb806e3338f09f9fb5c8	64,388.384645454815930719	0.6439%
7	0x4b04213c2774f77e60702880654206b116d00508	56,000	0.5600%
8	0x602a837508a0ed9a1ca28f467cf6b490f4260846	49,616.295587583720319298	0.4962%
9	0xcfb8ff09dec2e811c67ea857c93e367c83524771	37,921.082183665493298515	0.3792%
10	0xe0fce607cb496327dbf40a6d7e11347a018b80bc	31,269.910976524549788184	0.3127%
11	0x3c53fbe27a44df64eca248d85662626ae13efef3	29,639.384314100961187918	0.2964%
12	0x73de8abe57a1840da3bfccf25e4390b67f13f03e	29,169.171640096559678889	0.2917%
13	0xe07c3c936d5f22321c43cf48fa7344521eaae2d7	28,227.305996166099168956	0.2823%
14	0x7bb421e4a0b44d71d6731048c7e269a48ba49651	27,960.677140069267871707	0.2796%
15	0x12d56867e40bb3c19d8aae2e01526ec201de4f78	27,541.430585521546945782	0.2754%
16	0x17bc2565b1ea93c23d9a30060cf657657d491473	25,770.185340229733085651	0.2577%
17	0x93576d1c949f1629a8c047022a2539efce6330cb	25,531.64909215042571605	0.2553%
18	0x51d43f6d95e4f7026baf2f94f14fc6a06f149c5f	25,155.699764982506839327	0.2516%
19	0xef82e4141e0ad57f533b2cd71699cb0a33fb9c0f	24,908.173632151399825582	0.2491%
20	0xe14eaaa3a4b988ebe13cef2bf1708413c39f9b93	23,803.892456168033458506	0.2380%

OryxFi Distribution

OryxFi Contract Overview



Contract functions details

+[Lib] SafeMath

- [Int] add
- [Int] sub
- [Int] sub
- [Int] mul
- [Int] div
- [Int] div
- [Int] mod
- [Int] mod

+[Lib] Address

- [Int] isContract
- [Int] sendValue
- [Int] functionCall
- [Int] functionCall
- [Int] functionCallWithValue
- [Int] functionCallWithValue
- [Pvt] _functionCallWithValue

+[Int] IBEP20

- [Ext] totalSupply
- [Ext] balanceOf
- [Ext] transfer
- [Ext] allowance
- [Ext] approve
- [Ext] transferFrom

+ Context

- [Int] _msgSender
- [Int] _msgData

+ Ownable (Context)

- <constructor> #
- [Pub] owner #
- [Pub] transferOwnership #
- modifiers: onlyOwner

+ HasForeignAsset (Ownable)

- [Ext] assetBalance
- [Ext] getAsset
- modifiers: onlyOwner

Contract functions details

+ ORYX (IBEP20, HasForeignAsset)

-<constructor>

-[Ext] receive \$

-[Pub] name

-[Pub] symbol

-[Pub] decimals

-[Pub] totalSupply

-[Pub] balanceOf

-[Pub] transfer #

-[Pub] allowance #

-[Pub] approve #

-[Pub] transferFrom #

-[Pub] increaseAllowance #

-[Pub] decreaseAllowance #

-[Int] _transfer #

-[Int] _mintOnce#

-[Int] _burn #

-[Pub] burn #

-[Int] _approve #

-[Int] _setupDecimals

-[Int] _beforeTokenTransfer #

(\$) = payable function

= non-constant function

Issues Checking Status

No.	Title	Status
1.	Unlocked Compiler Version	Low issue
2.	Missing Input Validation	Passed
3.	Race conditions and Reentrancy. Cross-function race conditions.	Passed
4.	Possible delays in data delivery	Passed
5.	Oracle calls.	Passed
6.	Timestamp dependence.	Passed
7.	Integer Overflow and Underflow	Passed
8.	DoS with Revert.	Passed
9.	DoS with block gas limit.	Passed
10.	Methods execution permissions.	Passed
11.	Economy model of the contract.	Passed
12.	Private use data leaks.	Passed
13.	Malicious Event log.	Passed
14.	Scoping and Declarations.	Passed
15.	Uninitialized storage pointers.	Passed
16.	Arithmetic accuracy.	Passed
17.	Design Logic.	Passed
18.	Safe Open Zeppelin contracts implementation and usage.	Passed
19.	Incorrect Naming State Variable	Passed
20.	Compiler version too old	Passed

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.

Security Issues

✔ Critical Severity Issues

No critical severity issue found.

✔ High Severity Issues

No high severity issue found.

✔ Medium Severity Issues

No medium severity issues found.

✔ Low Severity Issues

One low severity issue found.

1. Unlocked Compiler Version.

- **Description**

The contract utilizes an unlocked compiler version. An unlocked compiler version in the contract's source code permits the user to compile it at or above a particular version. This, in turn, leads to differences in the generated bytecode between compilations due to differing compiler version numbers. This can lead to ambiguity when debugging as compiler-specific bugs may occur in the codebase that would be difficult to identify over a span of multiple compiler versions rather than a specific one.

- **Recommendation**

It is advisable that the compiler version is alternatively locked at the lowest version possible so that the contract can be compiled. For example, for version `^0.8.2` contract should contain the following line:

```
pragma solidity 0.8.7;
```

Centralization

Owner Privileges

- OryxFi Contract:
 - Owner can transfer ownership.
 - Owner can get assets balance of contract.

This smart contract has some functions which can be executed by the Admin (Owner) only. If the admin wallet private key would be compromised, then it would create trouble. The ownership of the smart contract can't renounced for the token, which means the owners are able to modify contract behavior. Please exercise with extra caution if you are investing in this asset. Following are Admin functions:

- Transferownership
- Getasset

Conclusion

Smart contract contains low severity issues! The further transfer and operations with the fund raised are not related to this particular contract.

HackSafe note: Please check the disclaimer above and note, the audit makes no statements or warranties on business model, investment attractiveness or code sustainability. The report is provided for the only contract mentioned in the report and does not include any other potential contracts deployed by Owner.