



Smart Contract Security Audit Report

ORIUM

September 2022

Security Status



www.hacksafe.io



Audit Details



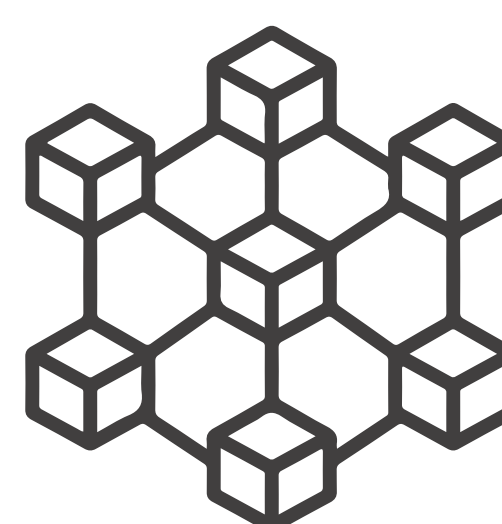
Audited project
ORIUM



Deployer address
0x106CeAE9db9A3249084AA2140AdD5fAb057F8Ed6



Client contacts
ORIUM Team



Blockchain
Ethereum



Website
Not provided

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 ORIUM to perform an audit of smart contract:

- <https://etherscan.io/token/0xd51e852630debc24e9e1041a03d80a0107f8ef0c#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 to understand 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 19.09.2022

Token Type	: ERC20
Contract name	: ORIUMCOIN
Contract address	: 0xd51e852630DeBC24E9e1041a03d80A0107F8Ef0C
Compiler version	: v0.4.11+commit.68ef5810
Total supply	: 200,000,000
Token ticker	: ORM
Decimals	: 0
Token holders	: 3,680
Transactions count	: 10,332
Contract deployer address	: 0x106CeAE9db9A3249084AA2140AdD5fAb057F8Ea6
Owner address	: 0x106ceae9db9a3249084aa2140add5fab057f8ea6

Social profiles

Twitter Profile	: https://twitter.com/OriumOfficial
Telegram profile	: https://t.me/ORIUMarmy
Facebook profile	: https://www.facebook.com/oriumofficial/
Coin Gecko profile	: https://www.coingecko.com/en/coins/orium/

Claimed Smart Contract Features

Claimed Feature Detail

Tokenomics :

- Name : ORIUMCOIN
- Symbol : ORM
- Decimals : 0
- Protocol : ERC20
- Total supply : 200,000,000
- Contract address : 0xd51e852630DeBC24E9e1041a03d80A0107F8Ef0C

Our Observation

YES, this is valid.

Audit Summary

According to the standard audit assessment, Customer`s solidity smart contracts are “well 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	Secure	Well-secured
----------	------	--------	--------------


You are here




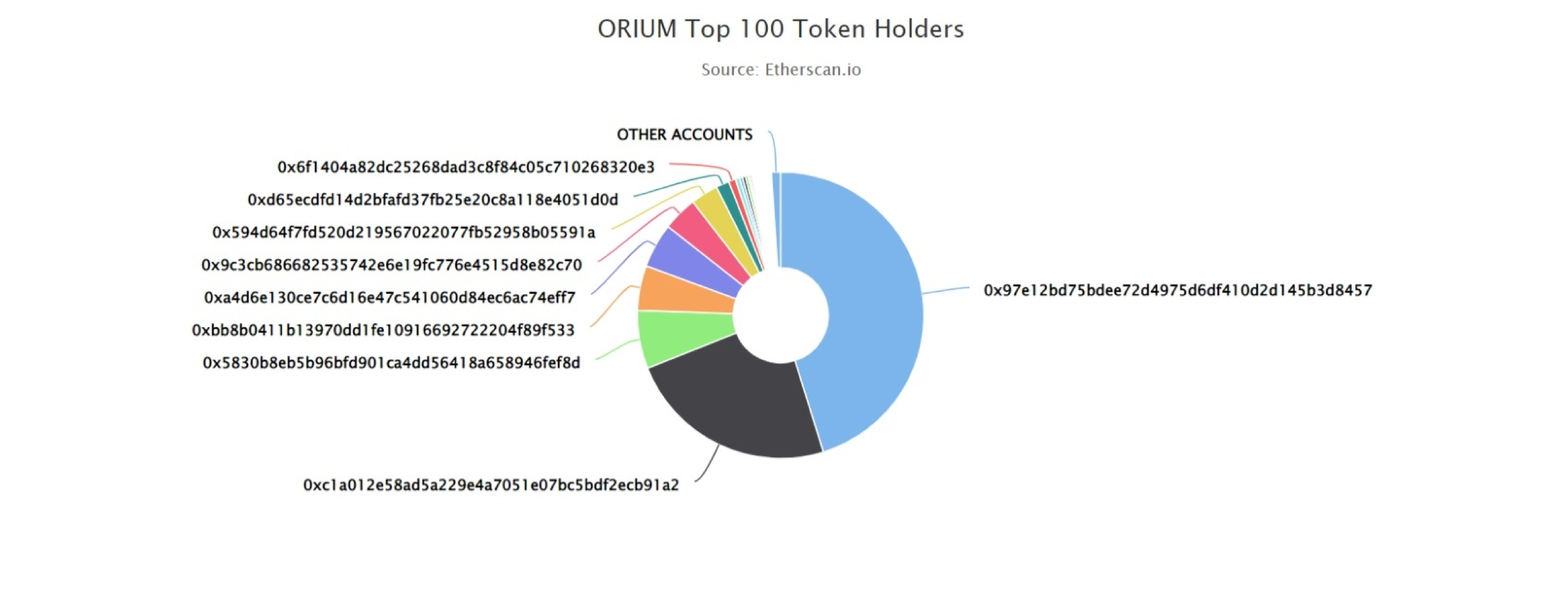
We used various tools like Slither, Mythril and Remix IDE. At the same time this finding is based on critical analysis of the manual audit. All issues found during automated analysis were manually reviewed and applicable vulnerabilities are presented in the issues checking status.

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

ORIUM Token Distribution

 The top 100 holders collectively own 98.95% (197,906,807.00 Tokens) of ORIUM

 Token Total Supply: 200,000,000.00 Token | Total Token Holders: 3,680



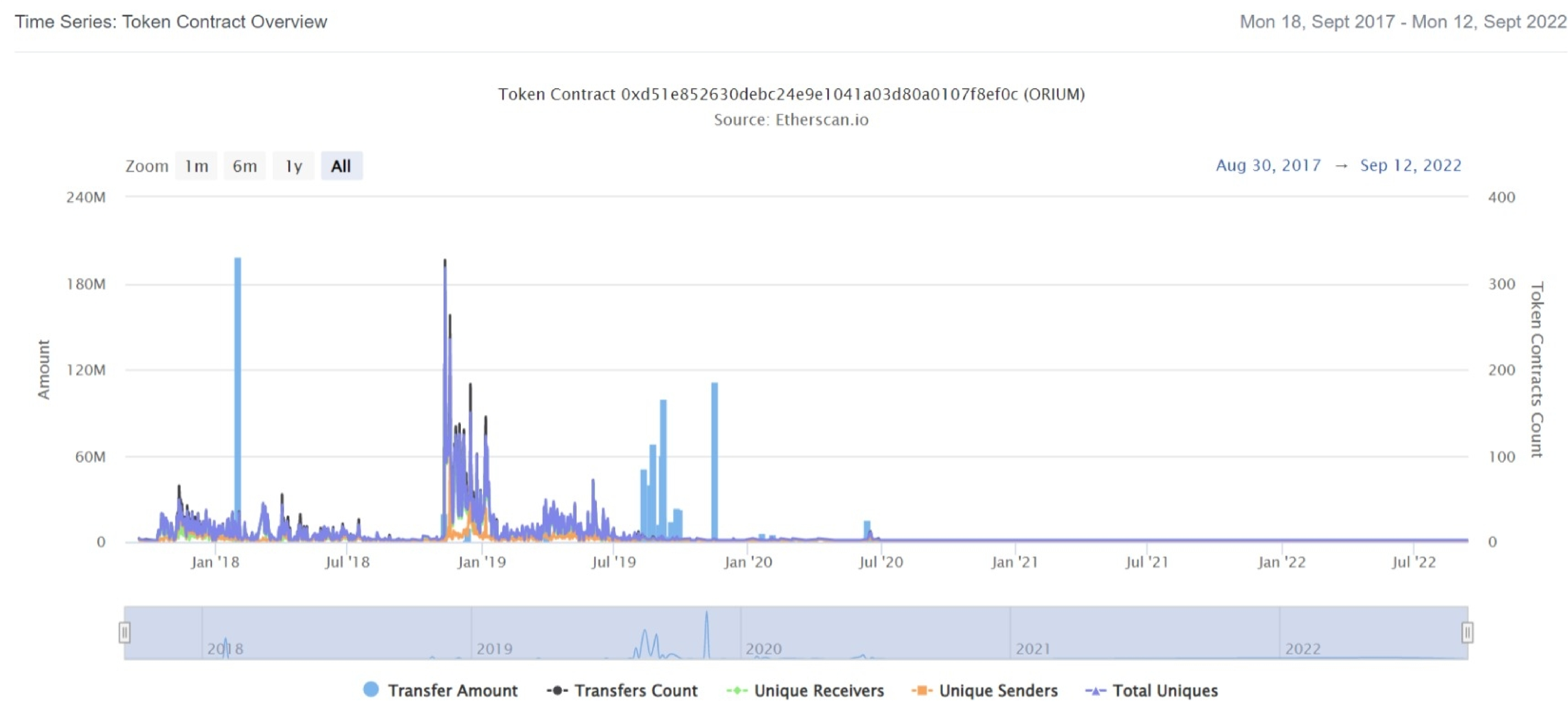
ORIUM Top 20 Token Holders

(A total of 197,906,807.00 tokens held by the top 100 accounts from the total supply of 200,000,000.00 token)

Rank	Address	Quantity (Token)	Percentage
1	0x97e12bd75bdee72d4975d6df410d2d145b3d8457	90,402,798	45.2014%
2	0xc1a012e58ad5a229e4a7051e07bc5bdf2ecb91a2	47,557,658	23.7788%
3	0x5830b8eb5b96bfd901ca4dd56418a658946fef8d	13,123,838	6.5619%
4	0xbb8b0411b13970dd1fe1091669272204f89f533	10,117,100	5.0586%
5	0xa4d6e130ce7c6d16e47c541060d84ec6ac74eff7	10,000,000	5.0000%
6	0x9c3cb686682535742e6e19fc776e4515d8e82c70	7,816,484	3.9082%
7	0x594d64f7fd520d219567022077fb52958b05591a	6,187,600	3.0938%
8	0xd65ecd14d2bfaf37fb25e20c8a118e4051d0d	3,011,488	1.5057%
9	0x6f1404a82dc25268dad3c8f84c05c710268320e3	1,525,245	0.7626%
10	Fake_Phishing311	980,467	0.4902%
11	0xa83acdfaea0e19343be77c73846ba23196f888bb	672,582	0.3363%
12	0x5e3e13b0a8a747a06c11ee460c71de3bdf2094bb	668,481	0.3342%
13	0x963e351972cc55664881575b4d4612736c91fd9a	618,929	0.3095%
14	0x5e6a6c6294f326e64f4b7715ccb8c606c50f91ea	403,100	0.2016%
15	0x77b63a2e95d6f08f9ad9d509cce00ac409670d1b	393,015	0.1965%
16	0xc9303bbbaec1ee921bc521e8af55ed26a2b5599	331,886	0.1659%
17	0xd4afc89ee8efe5fe6ae2032ceaf6335564bb6b7d	320,000	0.1600%
18	0x352b8089d41b278cec3b104e28fc4a0e8c63b234	289,304	0.1447%
19	0x8b9a999059c6beaf7e61e1357b11bf56337cd2b7	221,060	0.1105%
20	0x2b7a59945356b4100b1265b0588ce1a7a11d2fcd	172,790	0.0864%

ORIUM Token Distribution

ORIUM Contract Overview



Contract functions details

+ SafeMath

-[Int] mul

-[Int] div

-[Int] sub

-[Int] add

+ ORIUMCOIN

-ORIUMCOIN

-balanceOf

-allowance

-transfer #

-transferFrom #

- approve #

(\$) = payable function

= non-constant function

Issues Checking Status

No.	Title	Status
1.	Unlocked Compiler Version	Passed
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.	Too old version	Low issue

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 issues found.

✔ Medium Severity Issues

No medium severity issues found.

✔ Low Severity Issues

One low severity issue found.

1. Too old compiler version.

- **Description**

Contract has been deployed using too old compiler version.

- **Recommendation**

It is advisable that the compiler version of solidity should be among the new compiler versions.

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.