



MOTECH AUDIT

SMART CONTRACT SECURITY AUDIT

SECURITY ASSESSMENT

**UNORE TOKEN
AUDIT REPORT**

2021



12 NOV 2021

SECURITY ASSESMENT

TABLE OF CONTENTS

Summary	3
Disclaimer	4
Background	5
Audit Details	6
Contract Details	7
Unore Token Distribution	8
Unore Token Contract Interaction Details	8
Top 10 Token Holders	9
Security Issue	10 -11
Token Logo	12
Conclusion	13



SECURITY ASSESMENT

SUMMARY

This report has been prepared for Unore to discover issues and vulnerabilities in the source code of the Unore project as well as any contract dependencies that were not part of an officially recognized library. A comprehensive examination has been performed, utilizing Static Analysis, Manual Review, and Testnet Deployment techniques.

The auditing process pays special attention to the following considerations:

- Testing the smart contracts against both common and uncommon attack vectors.
- Assessing the codebase to ensure compliance with current best practices and industry standards.
- Ensuring contract logic meets the specifications and intentions of the client.
- Cross referencing contract structure and implementation against similar smart contracts produced by industry leaders.
- Thorough line-by-line manual review of the entire codebase by industry experts.

The security assessment resulted in findings that ranged from critical to informational. We recommend addressing these findings to ensure a high level of security standards and industry practices. We suggest recommendations that could better serve the project from the security perspective:

- Enhance general coding practices for better structures of source codes;
- Add enough unit tests to cover the possible use cases given they are currently missing in the repository;
- Provide more comments per each function for readability, especially contracts are verified in public;
- Provide more transparency on privileged activities once the protocol is live.



MOTECH AUDIT
SMART CONTRACT SECURITY AUDIT

SECURITY ASSESSMENT

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



MOTECH AUDIT
SMART CONTRACT SECURITY AUDIT

SECURITY ASSESSMENT

BACKGROUND

MotechAudit was commissioned by Unore to perform an audit of smart contracts:

<https://etherscan.io/address/0x474021845c4643113458ea4414bdb7fb74a01a77>

The purpose of the audit was to achieve the following:

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

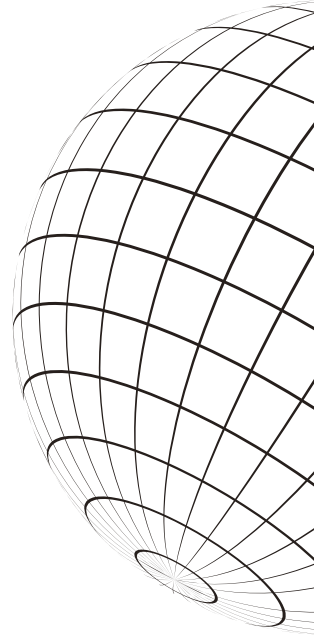
The information in this report should be used to understand the risk exposure of the smart contract, and as a guide to improve the security posture of the smart contract by remediating the issues that were identified.



MOTECH AUDIT
SMART CONTRACT SECURITY AUDIT

SECURITY ASSESSMENT

AUDIT DETAILS



AUDITED PROJECT

Unore



DEPLOYER ADDRESS

0xb782425E27A88921189a05bE7199748DdbDB71bf



CLIENT CONTACTS:

Unore team



BLOCKCHAIN

Ethereum Project



WEBSITE:

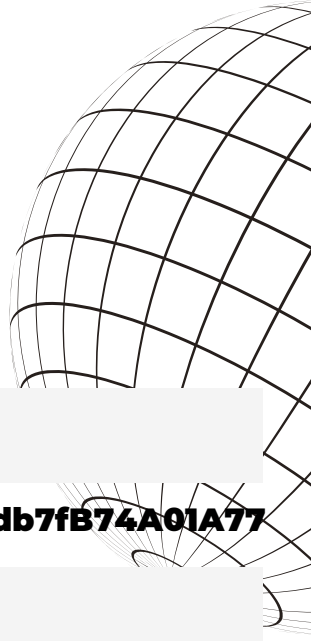
<https://unore.io/>



SECURITY ASSESSMENT

CONTRACT DETAILS

Token contract details for Apr-25-2021



Contract name	Unore
Contract address	0x474021845C4643113458ea4414bdb7fB74A01A77
Total supply	384,649,206 UNO
Token ticker	UnoRe (UNO)
Decimals	18
Token holders	2,861
Transactions count	29,598
Top 100 holders dominance	98.5442%
Contract deployer address	0xb782425E27A88921189a05bE7199748DdbDB71bf
Contract's current owner address	0xb782425E27A88921189a05bE7199748DdbDB71bf

SECURITY ASSESSMENT

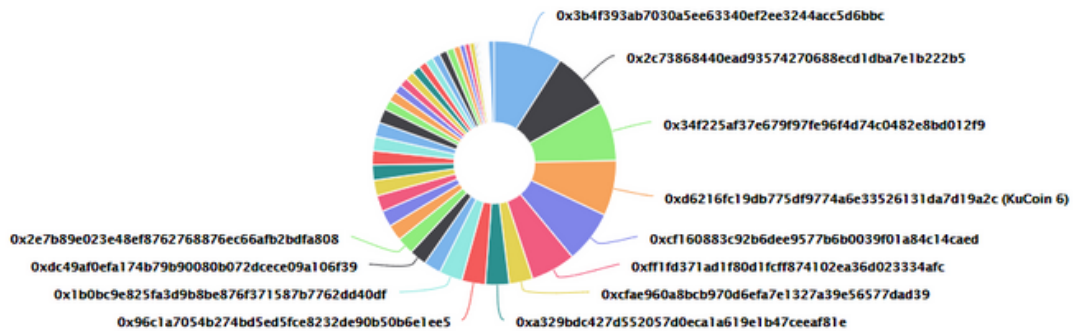
UNORE TOKEN DISTRIBUTION

The top 100 holders collectively own 99.23% (381,673,745.24 Tokens) of UnoRe

Token Total Supply: 384,649,206.00 Token | Total Token Holders: 2,861

UnoRe Top 100 Token Holders

Source: Etherscan.io

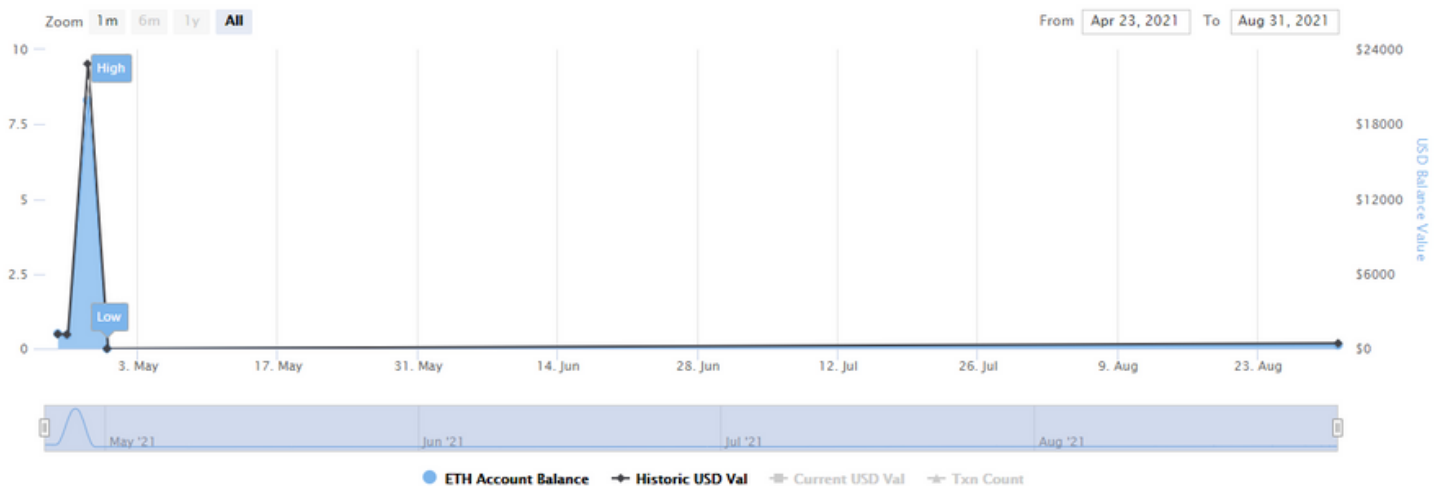


(A total of 381,673,745.24 tokens held by the top 100 accounts from the total supply of 384,649,206.00 token)

UNORE TOKEN CONTRACT INTERACTION DETAILS

Ether Balance for 0xb782425e27a88921189a05be7199748ddb71bf











Source: Etherscan.io



MOTECH AUDIT
SMART CONTRACT SECURITY AUDIT

SECURITY ASSESSMENT

TOP 10 TOKEN HOLDERS

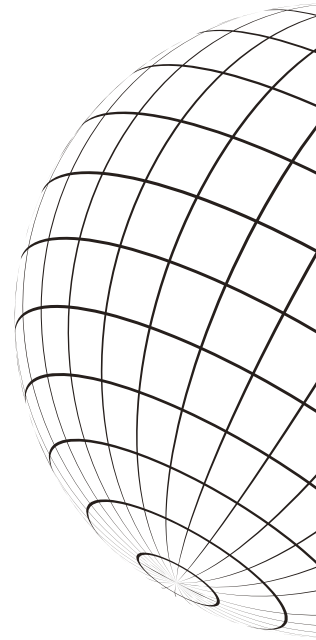
Rank	Address	Quantity	Percentage	Value	Analytics
1	0x3b4f393ab7030a5ee63340ef2ee3244acc5d6bbc	35,000,000	9.0992%	\$30,439,362.82	
2	0x2c73868440ead93574270688ecd1dba7e1b222b5	30,000,000	7.7993%	\$26,090,882.41	
3	0x34f225af37e679f97fe96f4d74c0482e8bd012f9	30,000,000	7.7993%	\$26,090,882.41	
4	KuCoin 6	28,220,292	7.3366%	\$24,543,077.34	
5	0xcf160883c92b6dee9577b6b0039f01a84c14caed	26,708,313.87	6.9436%	\$23,228,115.89	
6	0xff1fd371ad1f80d1fcff874102ea36d023334afc	22,799,369	5.9273%	\$19,828,521.86	
7	0xcfae960a8bcb970d6efa7e1327a39e56577dad39	12,069,047.5	3.1377%	\$10,496,403.31	
8	0xa329bdc427d552057d0eca1a619e1b47ceeaf81e	12,069,047.5	3.1377%	\$10,496,403.31	
9	0x96c1a7054b274bd5ed5fce8232de90b50b6e1ee5	12,069,047.5	3.1377%	\$10,496,403.31	
10	0x1b0bc9e825fa3d9b8be876f371587b7762dd40df	11,856,786.5	3.0825%	\$10,311,800.75	

source:<https://etherscan.io/>



SECURITY ASSESSMENT

SECURITY ISSUES



Critical Severity Issues

1. The contract owner can withdraw all funds from premium pools.

Contracts: Cohort.sol

Function: transferPremium

Recommendation: contract owner should not be able to withdraw funds that does not belong to him.

Status: fixed.

High Severity Issues

1. The function can be called before `cohortActiveFrom` time and premium rewards will be locked on the low level.

Contracts: Cohort.sol

Function: leaveFromPool

Status: fixed.

Medium Severity Issues

1. New `Cohort` is supposed to be created via the `Actuary` contract but the `newCohort` function of the `CohortFactory` is not restricted from external access.

Contracts: CohortFactory.sol

Function: newCohort

Recommendation: restrict access to the function.

Status: fixed.

2. The function is unnecessarily declared as a `write` function. Though it can be simplified to a `read` function. `_premiumReward` value is set only once and never gets changed.

Contracts: PremiumPool.sol

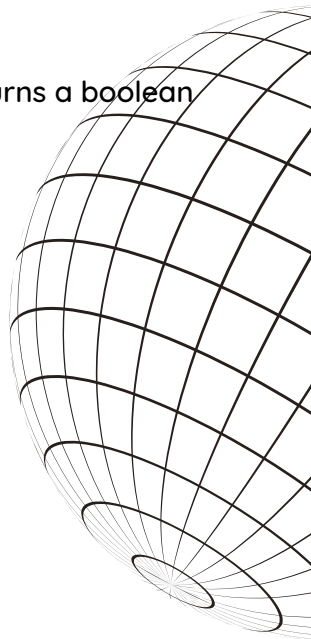
Function: premiumRewardOf

Recommendation: simplify the function.

Status: fixed.



MOTECH AUDIT
SMART CONTRACT SECURITY AUDIT



Low Severity Issues

1. Comparison with `true` is redundant. The `isCohortCreator` function already returns a boolean value. And the requirement will pass if the function returns `true`.

Contracts: Actuary.sol

Function: onlyCohortCreator

Recommendation: remove redundant comparison.

Status: fixed.

2. The `_cohortAddr` variable is used only in the return statement. `type(X).max`

Contracts: CohortFactory.sol

Function: newCohort

Recommendation: remove redundant local variable declaration.

Status: fixed.

3. `MAX_INTEGER` value is hardcoded but solidity built in constant can be used instead.

Contracts: Cohort.sol

Recommendation: use `type(uint256).max` instead of hardcoded value.

Status: fixed.

4. Result of the comparison can be set to `hasEnough` variable instead of explicitly set `true` or `false`.

Contracts: Cohort.sol

Function: hasEnoughCapital

Recommendation: remove redundant comparison.

Status: fixed.

Recommendations

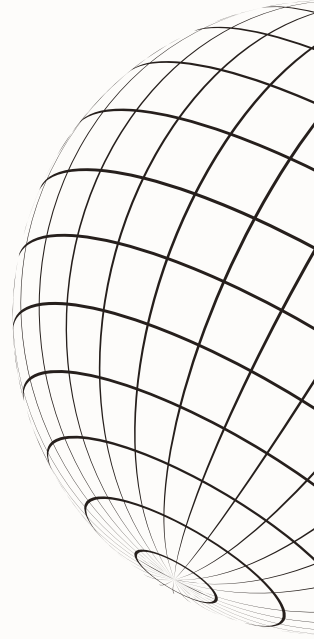
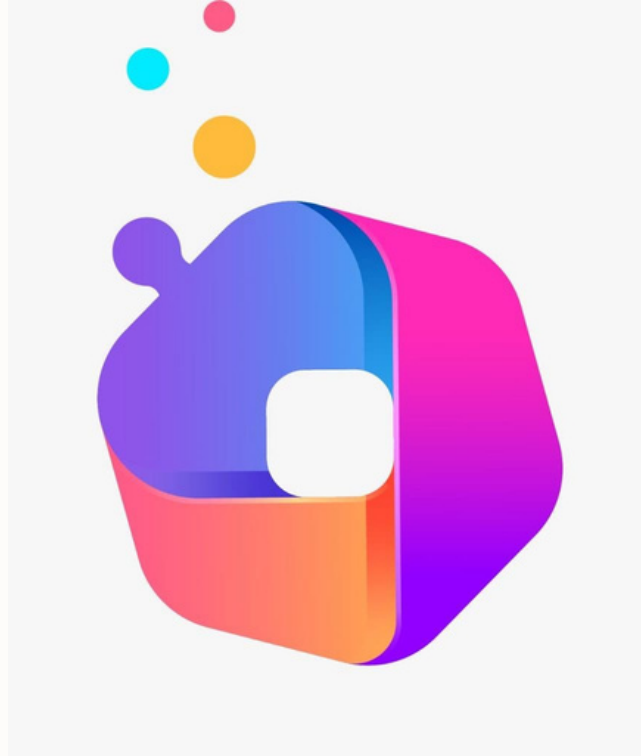
1. Initial pool capital is set as `MAX_INTEGER` without any need of such assignment. Such assignment leads to additional conditions in the `enterInPool`.

Contracts: Cohort.sol

Function: createRiskPool

Recommendation: do not set a pool capital as `MAX_INTEGER`. The code can be simplified.

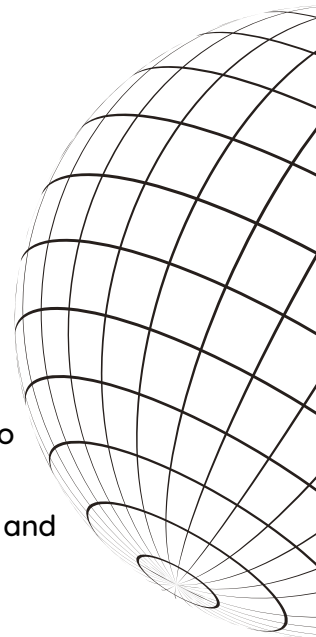
SECURITY ASSESMENT TOKEN LOGO



SECURITY ASSESMENT CONCLUSION

Smart contracts contain owner privileges!

Motech Audit 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.



MOTECH AUDIT
SMART CONTRACT SECURITY AUDIT