Cyberscope

# Audit Report

# Seed

October 2022

Github https://github.com/moonappxxx/moonapp-contracts

Commit 41a55dded0e77f34b71201e13f9448d0ed5dc4d4

Audited by © cyberscope

# Table of Contents

Table of Contents 1
Introduction 3
Roles 3
Contract Review 4
Audit Updates 4
Source Files 5
Contract Diagnostics 7
BLC - Business Logic Concern 8
Description 8
Recommendation 8
CO - Code Optimization 9
Description 9
Recommendation 9
MC - Missing Check 10
Description 10
Recommendation 11
L04 - Conformance to Solidity Naming Conventions 12
Description 12
Recommendation 12
L13 - Divide before Multiply Operation 13
Description 13
Recommendation 13
Contract Functions 14
Contract Flow 18
Domain Info 19
Summary 20

**Disclaimer** 21

**About Cyberscope** 22

# Introduction

The Seed contract manages the vesting contracts for the investors.

## Roles

The “admin” role has the ability to vest a number of tokens for each investor. The investors can be added until the initialization of the vesting.

# Contract Review

| | |
|---|---|
| **Contract Name** | Seed |
| **Compiler Version** | v0.8.11+commit.d7f03943 |
| **Github** | https://github.com/moonappxxx/moonapp-contracts |
| **Commit** | 41a55dded0e77f34b71201e13f9448d0ed5dc4d4 |
| **Testing Deploy** | https://testnet.bscscan.com/token/0x4144a1D4480126af171E006E555A8d087cF9ce9D |
| **Domain** | https://moonapp.org |

# Audit Updates

| | |
|---|---|
| **Initial Audit** | 3rd October 2022 |
| **Corrected** | |

# Source Files

| Filename | SHA256 |
| --- | --- |
| @openzeppelin/contracts/access/Ownable.sol | 9353af89436556f7ba8abb3f37a6677249aa4df6024fbfaa94f79ab2f44f3231 |
| @openzeppelin/contracts/token/ERC20/ERC20.sol | 5031430cc2613c32736d598037d3075985a2a09e61592a013dbd09a5bc2041b8 |
| @openzeppelin/contracts/token/ERC20/extensions/draft-IERC20Permit.sol | 3e7aa0e0f69eec8f097ad664d525e7b3f0a3fda8dcdd97de5433ddb131db86ef |
| @openzeppelin/contracts/token/ERC20/extensions/ERC20Burnable.sol | 0344809a1044e11ece2401b4f7288f414ea41fa9d1dad24143c84b737c9fc02e |
| @openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol | af5c8a77965cc82c33b7ff844deb9826166689e55dc037a7f2f790d057811990 |
| @openzeppelin/contracts/token/ERC20/IERC20.sol | 94f23e4af51a18c2269b355b8c7cf4db8003d075c9c541019eb8dcf4122864d5 |
| @openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol | fa36a21bd954262006d806b988e4495562e7b50420775e2aa0deecb596fd1902 |
| @openzeppelin/contracts/utils/Ad | 1e0922f6c0bf6b1b8b4d480dcabb691b1359195a297bde6dc5172e79f3a1f826 |

| | |
|---|---|
| **dress.sol** | |
| **@openzeppelin/contracts/utils/Context.sol** | 1458c260d010a08e4c20a4a517882259a23a4baa0b5bd9add9fb6d6a1549814a |
| **@openzeppelin/contracts/utils/math/Math.sol** | 929523c09910460ad708c75878d89b9fbed12b65cb5d8b670200c793131072f4 |
| **@openzeppelin/contracts/utils/math/SafeMath.sol** | 0dc33698a1661b22981abad8e5c6f5ebca0dfe5ec14916369a2935d888ff257a |
| **contracts/MoonappToken.sol** | 38e1865c3da8717a5a7176c2b46f184f99da9157a412df9676668018a011dd53 |
| **contracts/Seed.sol** | ce96edd88919581706853629b616797f6a13329772fa89a41c02e34bb45300d1 |
| **contracts/TokenVesting.sol** | 2f5262e07f85df4f5a54308df0e6cb28fc36e37192573da570ad944fcdf9786f |

# Contract Diagnostics

Critical Medium Minor / Informative

| Severity | Code | Description | Status |
|---|---|---|---|
| | BLC | Business Logic Concern | Unresolved |
| | CO | Code Optimization | Unresolved |
| | MC | Missing Check | Unresolved |
| | L04 | Conformance to Solidity Naming Conventions | Unresolved |
| | L13 | Divide before Multiply Operation | Unresolved |

# BLC - Business Logic Concern

| | |
|---|---|
| **Criticality** | minor / informative |
| **Location** | contract.sol#L84 |
| **Status** | Unresolved |

## Description

The business logic seems peculiar. The implementation may not follow the expected behavior.

The token should give permission (approve) over the Seed address before transfer transactions.

```
SafeERC20.safeTransfer(
        IERC20(token),
        address(vesting),
        tokensAmount
);
```

## Recommendation

The team is advised to carefully check if the implementation follows the expected business logic.

The contract could approve the entire vesting amount before transfer transcactions.

# CO - Code Optimization

| | |
|---|---|
| **Criticality** | minor / informative |
| **Location** | contract.sol#L70 |
| **Status** | Unresolved |

## Description

There are code segments that could be optimized. A segment may be optimized so that it becomes a smaller size, consumes less memory, executes more rapidly, or performs fewer operations.

Since the contract can not releaseToken after the start time, then the if statement is redundant.

```
if (investorVestings[investors[i]] != address(0x0)) continue;
```

## Recommendation

Rewrite some code segments so the runtime will be more performant.

# MC - Missing Check

| | |
|---|---|
| **Criticality** | minor / informative |
| **Location** | contract.sol#L47,76,27 |
| **Status** | Unresolved |

## Description

The contract is processing variables that have not properly sanitized and checked that they form the proper shape. These variables may produce vulnerability issues.

The contract is not taking into consideration all the invested amount.

```
require(
    amount <= availableTokens,
    "ADD_INVESTOR: not enought tokens left."
);
```

The following variables _cliff, _releaseRate, _initialReleaseRate are not properly sanitized before the vesting token initialization.

```
function releaseTokens(
    uint256 _start,uint256 _cliff, uint256 _releaseRate,  uint256 _initialReleaseRate
) external {
    //..
    uint256 initialReleaseAmont = (tokensAmount / 100) *
        _initialReleaseRate; // release % of the tokens on listing
    TokenVesting vesting = new TokenVesting(
        investors[i],
        startTime,
        _cliff,
        _releaseRate,
        initialReleaseAmont
    );
```

The variable availableTokens is not properly sanitized before its initialization in the contractor.

```
  constructor(address tokenAddress, uint256 _availableTokens) {
    token = MoonappToken(tokenAddress);
    admin = msg.sender;

    availableTokens = _availableTokens * (10**18);
  }
```

## Recommendation

The contract should properly check the variables according to the required specifications.

The contract should check if the total amount of tokens across investors is less than the available tokens.

It is recommended to pre-check variables before using them with other contracts.

- The variable start has to be greater than the current timestamp.
- The variable cliff has to be greater than zero
- The variable releaseRate have to be lower than 100%

The contract could embody a check on the contractor for the availableTokens variable to be greater than zero in order to avoid initializing the contract with zero available amount.

# L04 - Conformance to Solidity Naming Conventions

| | |
|---|---|
| **Criticality** | minor / informative |
| **Location** | contracts/Seed.sol#L35,59,58,61,98,60,106 |
| **Status** | Unresolved |

## Description

Solidity defines a naming convention that should be followed. Rule exceptions:

- Allow constant variable name/symbol/decimals to be lowercase.
- Allow _ at the beginning of the mixed_case match for private variables and unused parameters.

```
_tokensAmount
_investor
_cliff
_start
_initialReleaseRate
_releaseRate
```

## Recommendation

Follow the Solidity naming convention.
https://docs.soliditylang.org/en/v0.4.25/style-guide.html#naming-conventions.

# L13 - Divide before Multiply Operation

| | |
|---|---|
| **Criticality** | minor / informative |
| **Location** | contracts/Seed.sol#L57 |
| **Status** | Unresolved |

## Description

Performing divisions before multiplications may cause lose of prediction.

```
initialReleaseAmont = (tokensAmount / 100) * _initialReleaseRate
```

## Recommendation

The multiplications should be prior to the divisions.

# Contract Functions

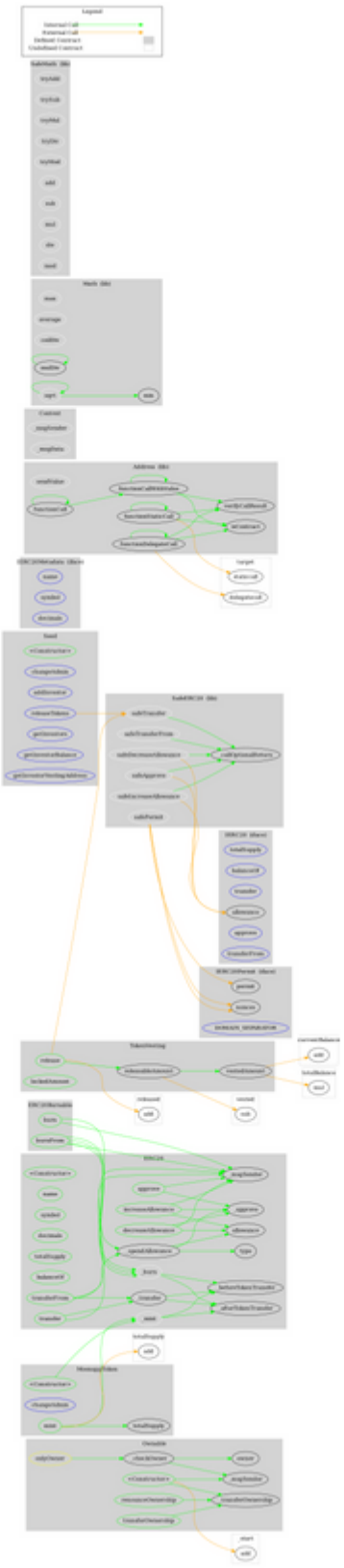
| Contract | Type | Bases | | |
|---|---|---|---|---|
| | Function Name | Visibility | Mutability | Modifiers |
| | | | | |
| **Ownable** | Implementation | Context | | |
| | <Constructor> | Public | ✓ | - |
| | owner | Public | | - |
| | _checkOwner | Internal | | |
| | renounceOwnership | Public | ✓ | onlyOwner |
| | transferOwnership | Public | ✓ | onlyOwner |
| | _transferOwnership | Internal | ✓ | |
| | | | | |
| **ERC20** | Implementation | Context, IERC20, IERC20Metadata | | |
| | <Constructor> | Public | ✓ | - |
| | name | Public | | - |
| | symbol | Public | | - |
| | decimals | Public | | - |
| | totalSupply | Public | | - |
| | balanceOf | Public | | - |
| | transfer | Public | ✓ | - |
| | allowance | Public | | - |
| | approve | Public | ✓ | - |
| | transferFrom | Public | ✓ | - |
| | increaseAllowance | Public | ✓ | - |
| | decreaseAllowance | Public | ✓ | - |
| | _transfer | Internal | ✓ | |
| | _mint | Internal | ✓ | |
| | _burn | Internal | ✓ | |
| | _approve | Internal | ✓ | |
| | _spendAllowance | Internal | ✓ | |

| | | | | |
|---|---|---|---|---|
| | _beforeTokenTransfer | Internal | ✓ | |
| | _afterTokenTransfer | Internal | ✓ | |
| | | | | |
| **IERC20Permit** | Interface | | | |
| | permit | External | ✓ | - |
| | nonces | External | | - |
| | DOMAIN_SEPARATOR | External | | - |
| | | | | |
| **ERC20Burnable** | Implementation | Context, ERC20 | | |
| | burn | Public | ✓ | - |
| | burnFrom | Public | ✓ | - |
| | | | | |
| **IERC20Metadata** | Interface | IERC20 | | |
| | name | External | | - |
| | symbol | External | | - |
| | decimals | External | | - |
| | | | | |
| **IERC20** | Interface | | | |
| | totalSupply | External | | - |
| | balanceOf | External | | - |
| | transfer | External | ✓ | - |
| | allowance | External | | - |
| | approve | External | ✓ | - |
| | transferFrom | External | ✓ | - |
| | | | | |
| **SafeERC20** | Library | | | |
| | safeTransfer | Internal | ✓ | |
| | safeTransferFrom | Internal | ✓ | |
| | safeApprove | Internal | ✓ | |
| | safeIncreaseAllowance | Internal | ✓ | |
| | safeDecreaseAllowance | Internal | ✓ | |
| | safePermit | Internal | ✓ | |
| | _callOptionalReturn | Private | ✓ | |

| | | | | |
|---|---|---|---|---|
| **Address** | Library | | | |
| | isContract | Internal | | |
| | sendValue | Internal | ✓ | |
| | functionCall | Internal | ✓ | |
| | functionCall | Internal | ✓ | |
| | functionCallWithValue | Internal | ✓ | |
| | functionCallWithValue | Internal | ✓ | |
| | functionStaticCall | Internal | | |
| | functionStaticCall | Internal | | |
| | functionDelegateCall | Internal | ✓ | |
| | functionDelegateCall | Internal | ✓ | |
| | verifyCallResult | Internal | | |
| | | | | |
| **Context** | Implementation | | | |
| | _msgSender | Internal | | |
| | _msgData | Internal | | |
| | | | | |
| **Math** | Library | | | |
| | max | Internal | | |
| | min | Internal | | |
| | average | Internal | | |
| | ceilDiv | Internal | | |
| | mulDiv | Internal | | |
| | mulDiv | Internal | | |
| | sqrt | Internal | | |
| | sqrt | Internal | | |
| | | | | |
| **SafeMath** | Library | | | |
| | tryAdd | Internal | | |
| | trySub | Internal | | |
| | tryMul | Internal | | |
| | tryDiv | Internal | | |
| | tryMod | Internal | | |
| | add | Internal | | |

| | | | | |
|---|---|---|---|---|
| | sub | Internal | | |
| | mul | Internal | | |
| | div | Internal | | |
| | mod | Internal | | |
| | sub | Internal | | |
| | div | Internal | | |
| | mod | Internal | | |
| | | | | |
| **MoonappToken** | Implementation | ERC20, ERC20Burnable | | |
| | <Constructor> | Public | ✓ | ERC20 |
| | changeAdmin | External | ✓ | - |
| | burnFrom | Public | ✓ | - |
| | mint | Public | ✓ | - |
| | | | | |
| **Seed** | Implementation | | | |
| | <Constructor> | Public | ✓ | - |
| | changeAdmin | External | ✓ | - |
| | addInvestor | External | ✓ | - |
| | releaseTokens | External | ✓ | - |
| | getInvestors | External | | - |
| | getInvestorBalance | External | | - |
| | getInvestorVestingAddress | External | | - |
| | | | | |
| **TokenVesting** | Implementation | Ownable | | |
| | <Constructor> | Public | ✓ | - |
| | release | Public | ✓ | - |
| | releasableAmount | Public | | - |
| | vestedAmount | Public | | - |
| | lockedAmount | Public | | - |

# Contract Flow



# Domain Info

| | |
|---|---|
| **Domain Name** | moonapp.org |
| **Registry Domain ID** | ebf9cc2ae696406f89ddb496f15a1e47-LROR |
| **Creation Date** | 2022-01-23T16:44:59Z |
| **Updated Date** | 2022-03-25T03:49:23Z |
| **Registry Expiry Date** | 2023-01-23T16:44:59Z |
| **Registrar WHOIS Server** | http://whois.reg.com |
| **Registrar URL** | http://www.reg.com |
| **Registrar** | Registrar of Domain Names REG.RU LLC |
| **Registrar IANA ID** | 1606 |

The domain was created 8 months before the creation of the audit. It will expire in 4 months.

There is no public billing information, the creator is protected by the privacy settings.

# Summary

The smart contract analysis reported no critical or compiler issues. This audit focused on investigating security issues and potential improvements.

# Disclaimer

All the content provided in this document is for general information only and should not be used as financial advice or a reason to buy any investment.

Cyberscope team provides no guarantees against the sale of team tokens or the removal of liquidity by the project audited in this document. Always Do your own research and protect yourselves from being scammed.

The Cyberscope team has audited this project for general information and only expresses their opinion based on similar projects and checks from popular diagnostic tools. Under no circumstances did Cyberscope receive a payment to manipulate those results or change the awarding badge that we will be adding in our website.

Always Do your own research and protect yourselves from scams. This document should not be presented as a reason to buy or not buy any particular token.

The Cyberscope team disclaims any liability for the resulting losses.

# About Cyberscope

Coinscope audit and K.Y.C. service has been rebranded to Cyberscope.

Coinscope is the leading early coin listing, voting and auditing authority firm. The audit process is analyzing and monitoring many aspects of the project. That way, it gives the community a good sense of security using an informative report and a generic score.

Cyberscope and Coinscope are aiming to make crypto discoverable and efficient globally. They provide all the essential tools to assist users draw their own conclusions.

The Cyberscope team

https://www.cyberscope.io