# Cyberscope

## Audit Report

# MoonappToken

October 2022

# Table of Contents

# Contract Review

| | |
|---|---|
| **Contract Name** | MoonappToken |
| **Compiler Version** | v0.8.9+commit.e5eed63a |
| **Github** | https://github.com/moonappxxx/moonapp-contracts |
| **Commit** | d4bb65891395e9fa805d5fae91544e7055a0a2e4 |
| **Testing Deploy** | https://testnet.bscscan.com/token/0x9e6f485A2eed3afe0D8fd4E7CAEA5B6f8ddF3E0e |
| **Symbol** | tst |
| **Decimals** | 18 |
| **Total Supply** | 99,999 |
| **Domain** | moonapp.org |

# Audit Updates

| | |
|---|---|
| **Initial Audit** | 3rd October 2022<br>https://github.com/cyberscope-io/audits/blob/main/1-xxx/v1/moonappToken.pdf |
| **Corrected Phase 1** | 7th October 2022<br>https://github.com/cyberscope-io/audits/blob/main/1-xxx/v2/moonappToken.pdf |
| **Corrected Phase 2** | 11th October 2022<br>https://github.com/cyberscope-io/audits/blob/main/1-xxx/v3/moonappToken.pdf |
| **Corrected Phase 3** | 14th October 2022 |

# Source Files

| Filename | SHA256 |
|---|---|
| @openzeppelin/contracts/token/ERC20/ERC20.sol | 5031430cc2613c32736d598037d3075985a2a09e61592a013dbd09a5bc2041b8 |
| @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/utils/Context.sol | 1458c260d010a08e4c20a4a517882259a23a4baa0b5bd9add9fb6d6a1549814a |
| @openzeppelin/contracts/utils/math/SafeMath.sol | 0dc33698a1661b22981abad8e5c6f5ebca0dfe5ec14916369a2935d888ff257a |
| contracts/Governed.sol | 5d2710dedb78bf95d62e48cda61bc378bd2924692f5153f39aabc5d4f2d616a9 |
| contracts/MoonappToken.sol | d5a47d5289583908466caf61a7fbe53a14e654a211fd60ee45115315a9cbe2d9 |

# Contract Analysis

● Critical  ● Medium  ● Minor / Informative  ● Pass

| Severity | Code | Description | Status |
|:---:|---|---|---|
| ● | ST | Stops Transactions | Passed |
| ● | OCTD | Transfers Contract's Tokens | Passed |
| ● | OTUT | Transfers User's Tokens | Passed |
| ● | ELFM | Exceeds Fees Limit | Passed |
| ● | ULTW | Transfers Liquidity to Team Wallet | Passed |
| ● | MT | Mints Tokens | Unresolved |
| ● | BT | Burns Tokens | Passed |
| ● | BC | Blacklists Addresses | Passed |

# MT - Mints Tokens

| Criticality | critical |
|---|---|
| Location | contract.sol#L41 |
| Status | Unresolved |

## Description

The contract owner has the authority to mint tokens. The owner may take advantage of it by calling the mint function. As a result, the contract tokens will be highly inflated.

There is a maximum mint limit number. The contract owner can mint up to 1000000000000000000000000000 tokens. That is 1000 times more than the initial total supply.

```solidity
function mint(address _account, uint256 _amount) external onlyGovernor {
    require(mintLockTime < block.timestamp, "mint is locked");

    uint256 _totalSupply = totalSupply();
    require(
        _totalSupply.add(_amount) <= totalSupplyLimit,
        "We are reached the limit in the total supply"
    );

    _mint(_account, _amount);
}
```

## Recommendation

The owner should carefully manage the credentials of the owner's account. We advised considering an extra-strong security mechanism that the actions may be quarantined by many users instead of one. The owner could also renounce the contract ownership for a period of time or pass the access to the zero address.

# Contract Diagnostics

● Critical    ● Medium    ● Minor / Informative

| Severity | Code | Description | Status |
|:---:|:---:|---|---|
| ● | L01 | Public Function could be Declared External | Unresolved |
| ● | L04 | Conformance to Solidity Naming Conventions | Unresolved |
| ● | L11 | Unnecessary Boolean equality | Unresolved |
| ● | L15 | Local Scope Variable Shadowing | Unresolved |

# L01 - Public Function could be Declared External

| Criticality | minor / informative |
| --- | --- |
| Location | @openzeppelin/contracts/token/ERC20/ERC20.sol#L62,70,87,101,113,136,158, 181,201<br><br>@openzeppelin/contracts/token/ERC20/extensions/ERC20Burnable.sol#L20,35 |
| Status | Unresolved |

## Description

Public functions that are never called by the contract should be declared external to save gas.

```
name
symbol
decimals
balanceOf
transfer
approve
transferFrom
increaseAllowance
decreaseAllowance
...
```

## Recommendation

Use the external attribute for functions never called from the contract.

# L04 - Conformance to Solidity Naming Conventions

| Criticality | minor / informative |
| --- | --- |
| Location | contracts/Governed.sol#L54<br><br>contracts/MoonappToken.sol#L36,41 |
| 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.

```
_newGovernor
_mintLockTime
_account
_amount
```

## Recommendation

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

# L11 - Unnecessary Boolean equality

| | |
|---|---|
| **Criticality** | minor / informative |
| **Location** | contracts/Governed.sol#L54,34 |
| **Status** | Unresolved |

## Description

The comparison to boolean constants is redundant. Boolean constants can be used directly and do not need to be compared to true or false.

```
require(bool,string)(governors[_newGovernor].ownershipAccepted !=
true,Permissions granted already)
require(bool,string)(governors[msg.sender].ownershipAccepted == true,Only
Governor can call)
```

## Recommendation

Remove the equality to the boolean constant.

# L15 - Local Scope Variable Shadowing

| | |
|---|---|
| **Criticality** | minor / informative |
| **Location** | contracts/MoonappToken.sol#L44 |
| **Status** | Unresolved |

## Description

The are variables that are defined in the local scope containing the same name from an upper scope.

```
_totalSupply
```

## Recommendation

The local variables should have different names from the upper scoped variables.

# Contract Functions

| Contract | Type | Bases | | |
|---|---|---|---|---|
| | Function Name | Visibility | Mutability | Modifiers |
| | | | | |
| 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 | ✓ | |
| | | | | |
| 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 | ✓ | - |
| | | | | |
| **Context** | Implementation | | | |
| | _msgSender | Internal | | |
| | _msgData | 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 | | |
| | | | | |
| **Governed** | Implementation | | | |
| | _initialize | Internal | ✓ | |

| | addOwnership | External | ✓ | onlyGovernor |
|---|---|---|---|---|
| | acceptOwnership | External | ✓ | - |
| | | | | |
| **MoonappToke n** | Implementation | ERC20, ERC20Burn able, Governed | | |
| | <Constructor> | Public | ✓ | ERC20 |
| | lockMint | External | ✓ | onlyGovernor |
| | mint | External | ✓ | onlyGovernor |

# Contract Flow
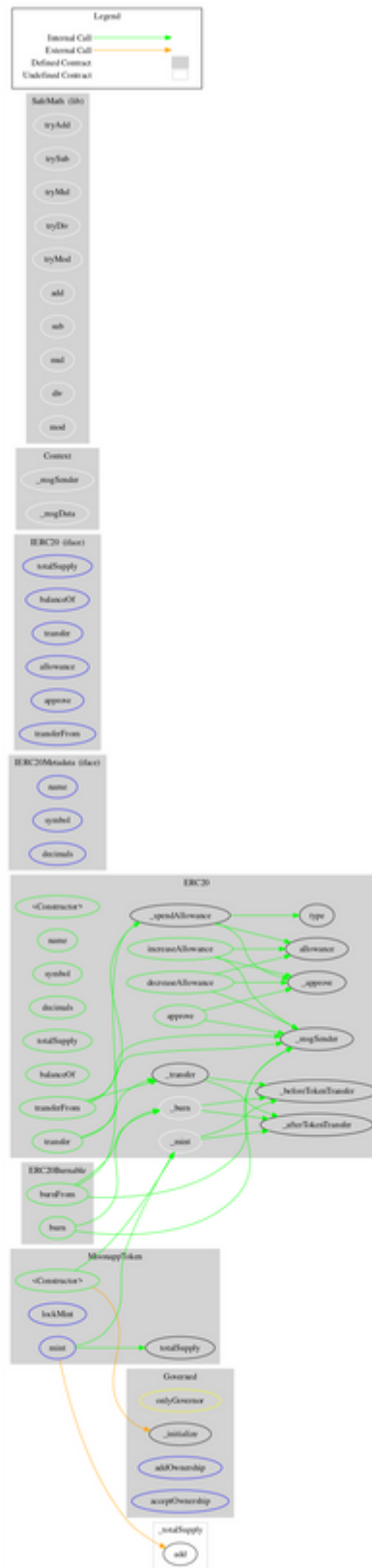
# Domain Info

| | |
|---|---|
| **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

There is a function that can be abused by the owner to mint tokens. If the contract owner abuses the mint functionality, then the contract will be highly inflated. A multi-wallet signing pattern will provide security against potential hacks. Temporarily locking the contract or renouncing ownership will eliminate all the contract threats.

# 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