Cyberscope

# Audit Report

# Battle Snake NFT

March 2022

| | |
|---|---|
| Type | BEP20 |
| Network | BSC |
| Address | 0xA0a39B2D5f5F82C86c027F98792197D068Db483d |
| Audited by | © cyberscope |

# Table of Contents

Table of Contents 1
Contract Review 3
Source Files 3
Audit Updates 3
Contract Analysis 4
ELFM - Exceed Limit Fees Manipulation 5
Description 5
Recommendation 5
MT - Mint Tokens 6
Description 6
Recommendation 6
BC - Blacklisted Contracts 7
Description 7
Recommendation 7
Contract Diagnostics 8
L01 - Public Function could be Declared External 9
Description 9
Recommendation 9
L04 - Conformance to Solidity Naming Conventions 10
Description 10
Recommendation 10
L11 - Unnecessary Boolean equality 11
Description 11
Recommendation 11
Contract Functions 12
Contract Flow 14

**Domain Info** 15

**Summary** 16

**Disclaimer** 17

**About Cyberscope** 18

# Contract Review

| | |
|---|---|
| **Contract Name** | CoinToken |
| **Compiler Version** | v0.4.24+commit.e67f0147 |
| **Optimization** | 200 runs |
| **Licence** | None |
| **Explorer** | https://bscscan.com/token/0xA0a39B2D5f5F82C86c027F98792197D068Db483d |
| **Symbol** | SNAKE |
| **Decimals** | 18 |
| **Total Supply** | 5,000,000 |
| **Domain** | battlesnakenft.com |

# Source Files

| **Filename** | **SHA256** |
|---|---|
| **contract.sol** | bb1bf689dfc02fe92fd154749b0eb05b9a6e816cb81579b190d9f8d2ca71d9d4 |

# Audit Updates

| | |
|---|---|
| **Initial Audit** | 28th March 2022 |
| **Corrected** | |

# Contract Analysis

Critical Medium Minor Pass

| Severity | Code | Description |
|---|---|---|
| ● | ST | Contract Owner is not able to stop or pause transactions |
| ● | OCTD | Contract Owner is not able to transfer tokens from specific address |
| ● | OTUT | Owner Transfer User's Tokens |
| ● | ELFM | Contract Owner is not able to increase fees more than a reasonable percent (25%) |
| ● | ULTW | Contract Owner is not able to increase the amount of liquidity taken by dev wallet more than a reasonable percent |
| ● | MT | Contract Owner is not able to mint new tokens |
| ● | BT | Contract Owner is not able to burn tokens from specific wallet |
| ● | BC | Contract Owner is not able to blacklist wallets from selling |

# ELFM - Exceed Limit Fees Manipulation

| | |
|---|---|
| **Criticality** | critical |
| **Location** | contract.sol#L289 |

## Description

The contract owner has the authority to increase over the allowed limit of 25%. The owner may take advantage of it by calling the `updateFee` function with a high percentage value.

```
      function updateFee(uint256 _txFee,uint256 _burnFee,address _FeeAddress)
onlyOwner public{
          txFee = _txFee;
          burnFee = _burnFee;
          FeeAddress = _FeeAddress;
      }
```

## Recommendation

The contract could embody a check for the maximum acceptable value.

The team should carefully manage the private keys of the owner's account. We strongly recommend a powerful security mechanism that will prevent a single user from accessing the contract admin functions. That risk can be prevented by temporarily locking the contract or renouncing ownership.

# MT - Mint Tokens

| | |
|---|---|
| **Criticality** | critical |
| **Location** | contract.sol#L304 |

## 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

```
function mint(address account, uint256 amount) onlyOwner public {

    totalSupply = totalSupply.add(amount);
    balances[account] = balances[account].add(amount);
    emit Mint(address(0), account, amount);
    emit Transfer(address(0), 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.

# BC - Blacklisted Contracts

| | |
|---|---|
| **Criticality** | medium |
| **Location** | contract.sol#L257 |

## Description

The contract owner has the authority to stop contracts from transactions. The owner may take advantage of it by calling the `blacklistAddress` function.

```
  function blackListAddress(address listAddress,  bool isBlackListed) public
whenNotPaused onlyOwner  returns (bool success) {
      return super._blackList(listAddress, isBlackListed);
  }
```

## Recommendation

The team should carefully manage the private keys of the owner’s account. We strongly recommend a powerful security mechanism that will prevent a single user from accessing the contract admin functions. That risk can be prevented by temporarily locking the contract or renouncing ownership.

# Contract Diagnostics

Critical Medium Minor

| Severity | Code | Description |
|---|---|---|
| | L01 | Public Function could be Declared External |
| | L04 | Conformance to Solidity Naming Conventions |
| | L11 | Unnecessary Boolean equality |

# L01 - Public Function could be Declared External

| | |
|---|---|
| **Criticality** | minor |
| **Location** | contract.sol#L51,85,93,101,107,253,281,285,300 |

## Description

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

```
mint
updateFee
burn
blackListAddress
allowance
balanceOf
unpause
pause
transferOwnership
```

## Recommendation

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

# L04 - Conformance to Solidity Naming Conventions

| | |
|---|---|
| **Criticality** | minor |
| **Location** | contract.sol#L128,157,161,189,196,201,207,118,233,237 and 20 more |

## 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.

```
_FeeAddress
_burnFee
_txFee
_value
_subtractedValue
_spender
_addedValue
_to
_from
...
```

## 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 |
| **Location** | contract.sol#L128,161 |

## 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)(tokenBlacklist[msg.sender] == false)
```

## Recommendation

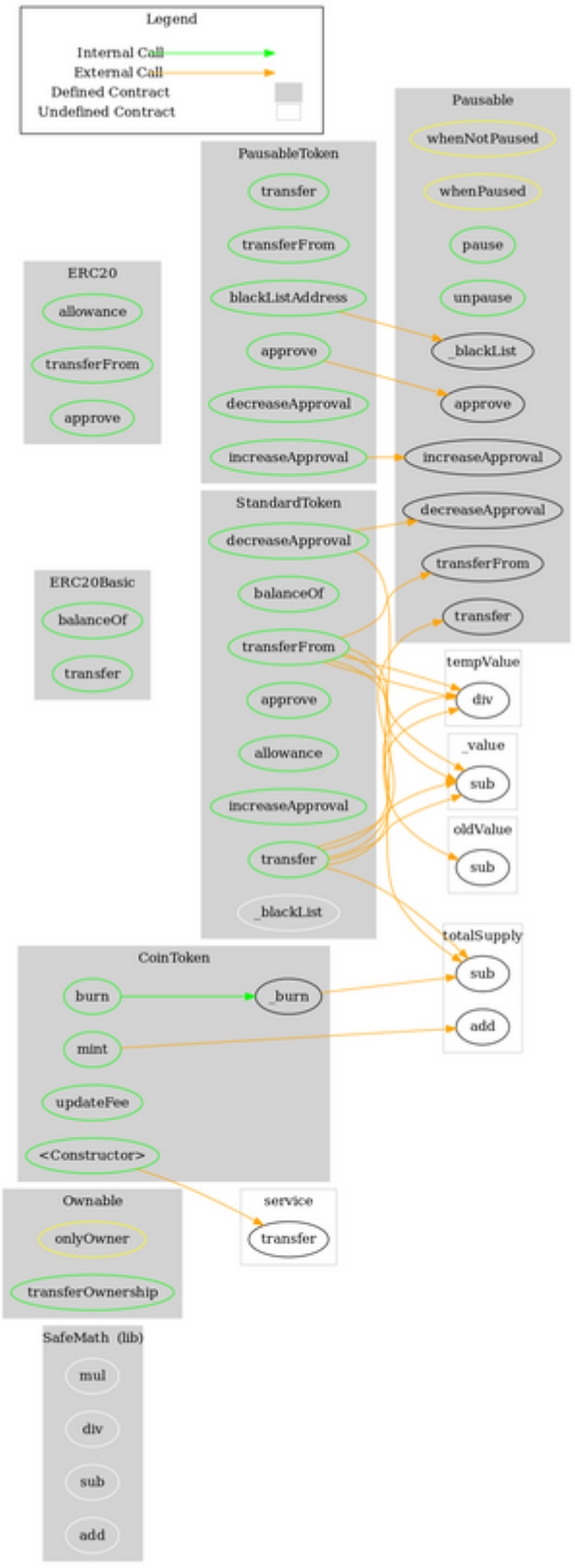
Remove the equality to the boolean constant.

# Contract Functions

| Contract | Type | Bases | | |
|---|---|---|---|---|
| | Function Name | Visibility | Mutability | Modifiers |
| | | | | |
| **SafeMath** | Library | | | |
| | mul | Internal | | |
| | div | Internal | | |
| | sub | Internal | | |
| | add | Internal | | |
| | | | | |
| **Ownable** | Implementation | | | |
| | transferOwnership | Public | ✓ | onlyOwner |
| | | | | |
| **Pausable** | Implementation | Ownable | | |
| | pause | Public | ✓ | onlyOwner whenNotPaused |
| | unpause | Public | ✓ | onlyOwner whenPaused |
| | | | | |
| **ERC20Basic** | Implementation | | | |
| | balanceOf | Public | | - |
| | transfer | Public | ✓ | - |
| | | | | |
| **ERC20** | Implementation | ERC20Basic | | |
| | allowance | Public | | - |
| | transferFrom | Public | ✓ | - |
| | approve | Public | ✓ | - |
| | | | | |
| **StandardToken** | Implementation | ERC20 | | |
| | transfer | Public | ✓ | - |
| | balanceOf | Public | | - |

| | | | | |
|---|---|---|---|---|
| | transferFrom | Public | ✓ | - |
| | approve | Public | ✓ | - |
| | allowance | Public | | - |
| | increaseApproval | Public | ✓ | - |
| | decreaseApproval | Public | ✓ | - |
| | _blackList | Internal | ✓ | |
| | | | | |
| **PausableToken** | Implementation | StandardToken, Pausable | | |
| | transfer | Public | ✓ | whenNotPaused |
| | transferFrom | Public | ✓ | whenNotPaused |
| | approve | Public | ✓ | whenNotPaused |
| | increaseApproval | Public | ✓ | whenNotPaused |
| | decreaseApproval | Public | ✓ | whenNotPaused |
| | blackListAddress | Public | ✓ | whenNotPaused onlyOwner |
| | | | | |
| **CoinToken** | Implementation | PausableToken | | |
| | <Constructor> | Public | Payable | - |
| | burn | Public | ✓ | - |
| | updateFee | Public | ✓ | onlyOwner |
| | _burn | Internal | ✓ | |
| | mint | Public | ✓ | onlyOwner |

# Contract Flow



# Domain Info

| | |
|---|---|
| **Domain Name** | battlesnakenft.com |
| **Registry Domain ID** | 953372 |
| **Creation Date** | 2022-03-12 |
| **Updated Date** | 2022-03-12 |
| **Registry Expiry Date** | 2023-03-12 |
| **Registrar WHOIS Server** | whois.ownregistrar.com |
| **Registrar URL** | http://www.ownregistrar.com |
| **Registrar** | OwnRegistrar, Inc. |
| **Registrar IANA ID** | 1250 |

The domain has been created 1 day before the creation of the audit. It will expire in 12 months.

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

# Summary

Battle Snake NFT is an interesting project that has a friendly and growing community. There are some functions that can be abused by the owner, like manipulating fees up to 100% and blacklisting users from transactions. The team can also mint new tokens after initial deployment and highly inflate the total supply. 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 provides all the essential tools to assist users draw their own conclusions.

The Cyberscope team

https://www.cyberscope.io