Cyberscope

# Audit Report

# BoredSpace

September 2022

| | |
|---|---|
| Type | BEP20 |
| Network | BSC |
| Address | 0x5B01Fa36C56a7A2e4e0d6741a2Af5Fa3dcbD59e1 |
| Audited by | © cyberscope |

# Table of Contents

Table of Contents 1
Contract Review 3
Source Files 3
Audit Updates 3
Contract Analysis 4
ST - Stops Transactions 5
Description 5
Recommendation 5
OCTD - Transfers Contract's Tokens 7
Description 7
Recommendation 7
ELFM - Exceeds Fees Limit 8
Description 8
Recommendation 8
ULTW - Transfers Liquidity to Team Wallet 9
Description 9
Recommendation 9
BC - Blacklists Addresses 10
Description 10
Recommendation 10
Contract Diagnostics 11
ZD - Zero Division 12
Description 12
Recommendation 12
STC - Succeeded Transfer Check 13
Description 13

Recommendation 13
BLC - Business Logic Concern 14
Description 14
Recommendation 14
L01 - Public Function could be Declared External 15
Description 15
Recommendation 15
L04 - Conformance to Solidity Naming Conventions 16
Description 16
Recommendation 16
L06 - Missing Events Access Control 17
Description 17
Recommendation 17
L13 - Divide before Multiply Operation 18
Description 18
Recommendation 18
L14 - Uninitialized Variables in Local Scope 19
Description 19
Recommendation 19
Contract Functions 20
Contract Flow 23
Domain Info 24
Summary 25
Disclaimer 26
About Cyberscope 27

# Contract Review

| | |
|---|---|
| **Contract Name** | TOKEN |
| **Compiler Version** | v0.8.17+commit.8df45f5f |
| **Optimization** | 200 runs |
| **Licence** | None |
| **Explorer** | https://bscscan.com/token/0x5B01Fa36C56a7A2e4e0d6741a2Af5Fa3dcbD59e1 |
| **Symbol** | BS |
| **Decimals** | 9 |
| **Total Supply** | 100,000,000 |
| **Domain** | boredspacelabs.com |

# Source Files

| **Filename** | **SHA256** |
|---|---|
| **contract.sol** | 6cf3e82e2f05bda5d38e1b39f22136b88ccca43ba36b785db887daeb2da960e7 |

# Audit Updates

| | |
|---|---|
| **Initial Audit** | 2nd October 2022 |
| **Corrected** | |

# Contract Analysis

● Critical ● Medium ● Minor / Informative ● Pass

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

# ST - Stops Transactions

| | |
|---|---|
| **Criticality** | critical |
| **Location** | contract.sol#L285,354,358,373 |
| **Status** | Unresolved |

## Description

The contract owner has the authority to stop the sales for all users excluding the owner. The owner may take advantage of it by setting the sale fees to a high value. As a result, the sender's amount will not be sufficient and the contract may operate as a honeypot.

```
if (isSell) {
    swapFee = _sellFunFee + _sellLPFee + _sellBuybackFee ;
} else {
```

Additionally, the contract owner has the authority to stop the transactions for all users excluding the owner. The owner may take advantage of it by setting the `goMoonBlock` or the `maxBuyAmount` or the `walletLimit` to zero.

```
if (0 == goMoonBlock) {
    require(false);
}
//...
require(tAmount <= maxBuyAmount,"over max buy amount");
//...
require((balanceOf(recipient) + tAmount - feeAmount) <= walletLimit,"over max
wallet limit");
```

## Recommendation

The contract could embody a check for not allowing setting the the `maxBuyAmount` or the `walletLimit` less than a reasonable amount. A suggested implementation could check that the maximum amount should be more than a fixed percentage of the total supply.

Read more about the fees manipulation in the corresponding section.

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.

# OCTD - Transfers Contract's Tokens

| | |
|---|---|
| **Criticality** | minor / informative |
| **Location** | contract.sol#L503 |
| **Status** | Unresolved |

## Description

The contract owner has the authority to claim all the balance of the contract. The owner may take advantage of it by calling the `claimTokens` function.

```
function claimToken(address token, uint256 amount, address to) external
onlyFunder {
    IERC20(token).transfer(to, amount);
}
```

## 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.

# ELFM - Exceeds Fees Limit

| | |
|---|---|
| **Criticality** | critical |
| **Location** | contract.sol#L454-470 |
| **Status** | Unresolved |

## 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 `setBuyFundFee, setBuyBuyBackFee, setSellBuyBackFee, setSellFundFee or setSellLPFee` function with a high percentage value.

```
function setSellFundFee(uint256 fundFee) external onlyOwner {
    _sellFunFee = fundFee;
}
```

## 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.

# ULTW - Transfers Liquidity to Team Wallet

| | |
|---|---|
| **Criticality** | minor / informative |
| **Location** | contract.sol#L499 |
| **Status** | Unresolved |

## Description

The contract owner has the authority to transfer funds without limit to the team wallet. These funds have been accumulated from fees collected from the contract. The owner may take advantage of it by calling the `claimBalance` method.

```
function claimBalance() external {
    payable(fundAddress).transfer(address(this).balance);
}
```

## 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.

# BC - Blacklists Addresses

| | |
|---|---|
| **Criticality** | critical |
| **Location** | contract.sol#L536 |
| **Status** | Unresolved |

## Description

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

```
function manage_bl(address[] calldata addresses, bool status) public onlyOwner {
    require(addresses.length < 201);
    for (uint256 i; i < addresses.length; ++i) {
        _blackList[addresses[i]] = status;
    }
}
```

## 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 / Informative

| Severity | Code | Description | Status |
|---|---|---|---|
| ● | ZD | Zero Division | Unresolved |
| ● | STC | Succeeded Transfer Check | Unresolved |
| ● | BLC | Business Logic Concern | Unresolved |
| ● | L01 | Public Function could be Declared External | Unresolved |
| ● | L04 | Conformance to Solidity Naming Conventions | Unresolved |
| ● | L06 | Missing Events Access Control | Unresolved |
| ● | L13 | Divide before Multiply Operation | Unresolved |
| ● | L14 | Uninitialized Variables in Local Scope | Unresolved |

# ZD - Zero Division

| | |
|---|---|
| **Criticality** | critical |
| **Location** | contract.sol#L389 |
| **Status** | Unresolved |

## Description

The contract is using variables that may be set to zero as denominators. As a result, the transactions will revert.

```
uint256 lpAmount = tokenAmount * lpFee / swapFee;
//...
uint256 ALL_fundAmount = fistBalance * (_buyFunFee + _sellFunFee +
_buyBuybackFee + _sellBuybackFee) * 2 / swapFee;
```

## Recommendation

The contract should prevent those variables to be set to zero or should not allow to execute the corresponding statements.

# STC - Succeeded Transfer Check

| | |
|---|---|
| **Criticality** | minor / informative |
| **Location** | contract.sol#L412,428,386,508 |
| **Status** | Unresolved |

## Description

According to the ERC20 specification, the transfer methods should be checked if the result is successful. Otherwise, the contract may wrongly assume that the transfer has been established.

```
FIST.transferFrom(address(_tokenDistributor), address(this), ALL_fundAmount);
//...
FIST.transferFrom(address(_tokenDistributor), address(this), fistBalance -
ALL_fundAmount);
//
recipient.transfer(amount);
//
IERC20(token).transfer(to, amount);
```

## Recommendation

The contract should check if the result of the transfer methods is successful.

# BLC - Business Logic Concern

| | |
|---|---|
| **Criticality** | critical |
| **Location** | contract.sol#L404 |
| **Status** | Unresolved |

## Description

Since the `_fist` is an external source, it should be checked if the if the amounts that are expected to be transferred have been transferred. Otherwise diversion may be produced between the expected and the actual values.

```
IERC20 FIST = IERC20(_fist);
uint256 fistBalance = FIST.balanceOf(address(_tokenDistributor));
uint256 ALL_fundAmount = fistBalance * (_buyFunFee + _sellFunFee +
_buyBuybackFee + _sellBuybackFee) * 2 / swapFee;

uint256 fundAmount_Buyback = ALL_fundAmount / (_buyFunFee + _sellFunFee +
_buyBuybackFee + _sellBuybackFee) * (_buyBuybackFee + _sellBuybackFee);
uint256 fundAmount_market = ALL_fundAmount - fundAmount_Buyback;

// FIST.transferFrom(address(_tokenDistributor), fundAddress, fundAmount);
FIST.transferFrom(address(_tokenDistributor), address(this), ALL_fundAmount);
IWETH(_fist).withdraw(fundAmount_market);
...
```

## Recommendation

The contract could embody try-catch statements and balance validations on every interaction with external addresses.

# L01 - Public Function could be Declared External

| | |
|---|---|
| **Criticality** | minor / informative |
| **Location** | contract.sol#L83,92,97,223,231,236,240,245,258,266,377,513,536 |
| **Status** | Unresolved |

## Description

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

```
owner
renounceOwnership
transferOwnership
totalSupply
transfer
allowance
approve
transferFrom
setLimitEnable
...
```

## Recommendation

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

# L04 - Conformance to Solidity Naming Conventions

| | |
|---|---|
| **Criticality** | minor / informative |
| **Location** | contract.sol#L37,73,377,513,527,536,115,126,127,133,134,135,141,143,144,146,147,149,153 |
| **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.

```
WETH
_owner
_maxBuyAmount
_walletLimit
manage_wl
multiTransfer_fixed
manage_bl
_buyBackToken
_feeWhiteList
...
```

## Recommendation

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

# L06 - Missing Events Access Control

| | |
|---|---|
| **Criticality** | minor / informative |
| **Location** | contract.sol#L449 |
| **Status** | Unresolved |

## Description

Detected missing events for critical access control parameters. There are functions that have no event emitted, so it is difficult to track off-chain changes.

```
fundAddress = addr
```

## Recommendation

Emit an event for critical parameter changes.

# L13 - Divide before Multiply Operation

| | |
|---|---|
| **Criticality** | minor / informative |
| **Location** | contract.sol#L386 |
| **Status** | Unresolved |

## Description

Performing divisions before multiplications may cause lose of prediction.

```
fundAmount_Buyback = ALL_fundAmount / (_buyFunFee + _sellFunFee + _buyBuybackFee
+ _sellBuybackFee) * (_buyBuybackFee + _sellBuybackFee)
```

## Recommendation

The multiplications should be prior to the divisions.

# L14 - Uninitialized Variables in Local Scope

| | |
|---|---|
| **Criticality** | minor / informative |
| **Location** | contract.sol#L281,538,515,350,280 |
| **Status** | Unresolved |

## Description

The are variables that are defined in the local scope and are not initialized.

```
isSell
i
feeAmount
takeFee
```

## Recommendation

All the local scoped variables should be initialized.

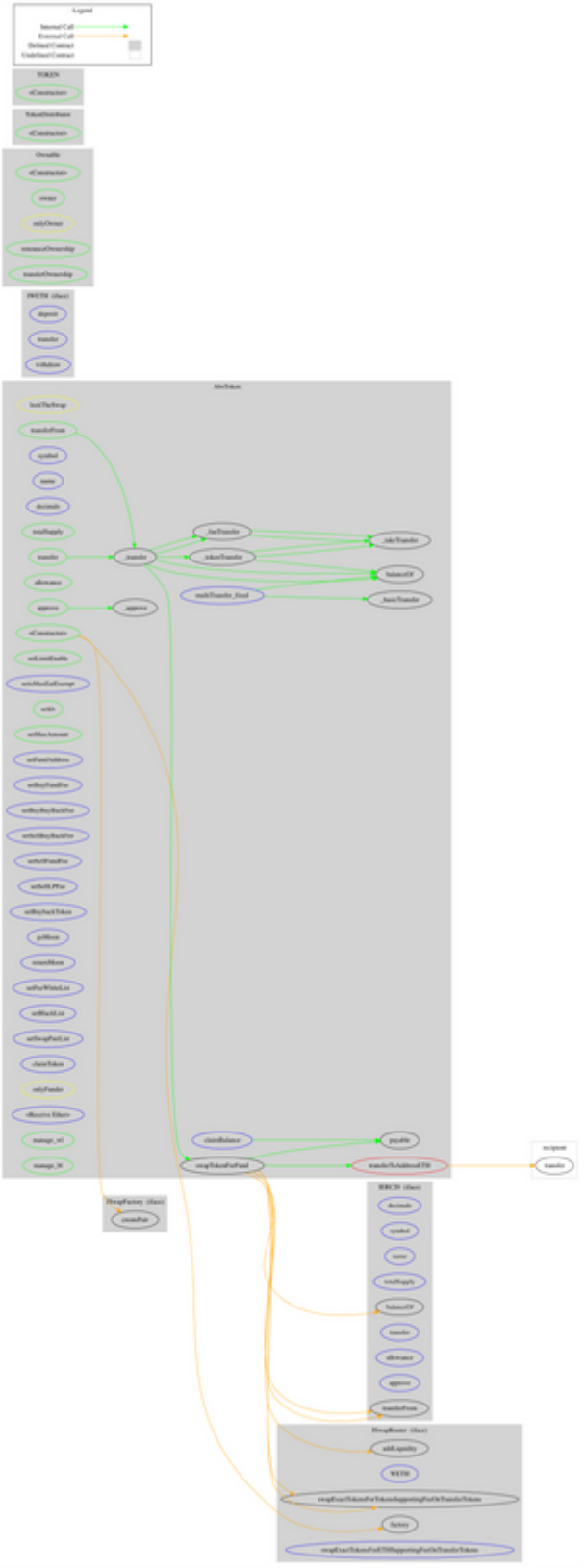
# Contract Functions

| Contract | Type | Bases | | |
|---|---|---|---|---|
| | Function Name | Visibility | Mutability | Modifiers |
| | | | | |
| IERC20 | Interface | | | |
| | decimals | External | | - |
| | symbol | External | | - |
| | name | External | | - |
| | totalSupply | External | | - |
| | balanceOf | External | | - |
| | transfer | External | ✓ | - |
| | allowance | External | | - |
| | approve | External | ✓ | - |
| | transferFrom | External | ✓ | - |
| | | | | |
| IWETH | Interface | | | |
| | deposit | External | Payable | - |
| | transfer | External | ✓ | - |
| | withdraw | External | ✓ | - |
| | | | | |
| ISwapRouter | Interface | | | |
| | factory | External | | - |
| | WETH | External | | - |
| | swapExactTokensForTokensSupportingFeeOnTransferTokens | External | ✓ | - |
| | addLiquidity | External | ✓ | - |
| | swapExactTokensForETHSupportingFeeOnTransferTokens | External | ✓ | - |
| | | | | |
| ISwapFactory | Interface | | | |
| | createPair | External | ✓ | - |
| | | | | |
| Ownable | Implementation | | | |

| | | | | |
|---|---|---|---|---|
| | <Constructor> | Public | ✓ | - |
| | owner | Public | | - |
| | renounceOwnership | Public | ✓ | onlyOwner |
| | transferOwnership | Public | ✓ | onlyOwner |
| | | | | |
| **TokenDistributor** | Implementation | | | |
| | <Constructor> | Public | ✓ | - |
| | | | | |
| **AbsToken** | Implementation | IERC20, Ownable | | |
| | <Constructor> | Public | ✓ | - |
| | symbol | External | | - |
| | name | External | | - |
| | decimals | External | | - |
| | totalSupply | Public | | - |
| | balanceOf | Public | | - |
| | transfer | Public | ✓ | - |
| | allowance | Public | | - |
| | approve | Public | ✓ | - |
| | transferFrom | Public | ✓ | - |
| | _approve | Private | ✓ | |
| | setLimitEnable | Public | ✓ | onlyOwner |
| | setisMaxEatExempt | External | ✓ | onlyOwner |
| | setkb | Public | ✓ | onlyOwner |
| | _transfer | Private | ✓ | |
| | _funTransfer | Private | ✓ | |
| | _tokenTransfer | Private | ✓ | |
| | setMaxAmount | Public | ✓ | onlyOwner |
| | transferToAddressETH | Private | ✓ | |
| | swapTokenForFund | Private | ✓ | lockTheSwap |
| | _takeTransfer | Private | ✓ | |
| | setFundAddress | External | ✓ | onlyFunder |
| | setBuyFundFee | External | ✓ | onlyOwner |
| | setBuyBuyBackFee | External | ✓ | onlyOwner |
| | setSellBuyBackFee | External | ✓ | onlyOwner |

| | | | | |
|---|---|---|---|---|
| | setSellFundFee | External | ✓ | onlyOwner |
| | setSellLPFee | External | ✓ | onlyOwner |
| | setBuybackToken | External | ✓ | onlyOwner |
| | goMoon | External | ✓ | onlyOwner |
| | returnMoon | External | ✓ | onlyOwner |
| | setFeeWhiteList | External | ✓ | onlyFunder |
| | setBlackList | External | ✓ | onlyOwner |
| | setSwapPairList | External | ✓ | onlyFunder |
| | claimBalance | External | ✓ | - |
| | claimToken | External | ✓ | onlyFunder |
| | <Receive Ether> | External | Payable | - |
| | manage_wl | Public | ✓ | onlyOwner |
| | _basicTransfer | Internal | ✓ | |
| | multiTransfer_fixed | External | ✓ | onlyOwner |
| | manage_bl | Public | ✓ | onlyOwner |
| | | | | |
| **TOKEN** | Implementation | AbsToken | | |
| | <Constructor> | Public | ✓ | AbsToken |

# Contract Flow



# Domain Info

| | |
|---|---|
| **Domain Name** | boredspacelabs.com |
| **Registry Domain ID** | 2727642744_DOMAIN_COM-VRSN |
| **Creation Date** | 2022-09-25T02:15:19Z |
| **Updated Date** | 2022-09-25T02:15:20Z |
| **Registry Expiry Date** | 2023-09-25T02:15:19Z |
| **Registrar WHOIS Server** | |
| **Registrar URL** | |
| **Registrar** | Epik Inc. |
| **Registrar IANA ID** | 617 |

The domain was created 8 days 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

There are some functions that can be abused by the owner like stopping transactions, transferring tokens to the team’s wallet, transferring the user's tokens, manipulating fees and massively blacklisting addresses. The contract can be converted into a honeypot and prevent users from selling if the owner abuses the admin functions. A multi-wallet signing pattern will provide security against potential hacks. Temporarily locking the contract or renouncing ownership will eliminate all the contract threats. There is also a limit of max 25% fees.

# 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