



Cyberscope

Audit Report

SLUMDOGz

July 2022

Type BEP20

Network BSC

Address 0xf5f6fde09d39bb746ac95d77b175bf3082446173

Audited by © cyberscope

Table of Contents

Table of Contents	1
Contract Review	3
Source Files	3
Audit Updates	3
Contract Analysis	4
OCTD - Owner Contract Tokens Drain	5
Description	5
Recommendation	5
Contract Diagnostics	6
L01 - Public Function could be Declared External	7
Description	7
Recommendation	7
L02 - State Variables could be Declared Constant	8
Description	8
Recommendation	8
L04 - Conformance to Solidity Naming Conventions	9
Description	9
Recommendation	9
L05 - Unused State Variable	10
Description	10
Recommendation	10
L07 - Missing Events Arithmetic	11
Description	11
Recommendation	11
L09 - Dead Code Elimination	12
Description	12

Recommendation	12
L13 - Divide before Multiply Operation	13
Description	13
Recommendation	13
Contract Flow	14
Domain Info	15
Summary	16
Disclaimer	17
About Cyberscope	18

Contract Review

Contract Name	TOKEN
Compiler Version	v0.8.13+commit.abaa5c0e
Optimization	200 runs
Licence	MIT
Explorer	https://bscscan.com/token/0xf5f6fde09d39bb746ac95d77b175bf3082446173
Symbol	SDT
Decimals	18
Total Supply	1,000,000,000
Domain	slumdogz-token.com

Source Files

Filename	SHA256
contract.sol	980268320a53649cb9c7be0c9c94c6be445a8daf28bc389bf2e96c71802290e1

Audit Updates

Initial Audit	13th July 2022
Corrected	18th July 2022

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

OCTD - Owner Contract Tokens Drain

Criticality	minor
Location	contract.sol#L1072

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 `withdrawStuckedFunds` function.

```
function withdrawStuckedFunds(uint256 amount) external onlyOwner {  
    // This is the current recommended method to use.  
    (bool sent, ) = _owner.call{value: amount}("");  
    require(sent, "Failed to withdraw BNB");  
}
```

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
●	L02	State Variables could be Declared Constant
●	L04	Conformance to Solidity Naming Conventions
●	L05	Unused State Variable
●	L07	Missing Events Arithmetic
●	L09	Dead Code Elimination
●	L13	Divide before Multiply Operation

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L318,323,333,341,801,805,809,813,822,831,840,849,899,903,915,943

Description

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

```
excludeFromReward  
reflectionFromToken  
deliver  
totalFees  
transferFrom  
approve  
allowance  
transfer  
totalSupply  
...
```

Recommendation

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

L02 - State Variables could be Declared Constant

Criticality

minor

Location

contract.sol#L718

Description

Constant state variables should be declared constant to save gas.

```
_totalFees
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L295,297,415,417,448,494,997,1001,1063,1212,1216,1224,706,708,709,710,734,735,736,737,738

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.

```
_burnFee  
_devFee  
_marketingFee  
_liquidityFee  
_taxFee  
_burnAddress  
_devWalletAddress  
_marketingWalletAddress  
_isBlacklisted  
...
```

Recommendation

Follow the Solidity naming convention.

<https://docs.soliditylang.org/en/v0.4.25/style-guide.html#naming-conventions>.

L05 - Unused State Variable

Criticality

minor

Location

contract.sol#L718

Description

There are segments that contain unused state variables.

`_totalFees`

Recommendation

Remove unused state variables.

L07 - Missing Events Arithmetic

Criticality

minor

Location

contract.sol#L1005,1026,1049

Description

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

```
numTokensSellToAddToLiquidity = amount * 10 ** _decimals
_buyTaxFee = tFee
_sellTaxFee = tFee
```

Recommendation

Emit an event for critical parameter changes.

L09 - Dead Code Elimination

Criticality

minor

Location

contract.sol#L274,183,190,198,212,252,264,229,242,163,171

Description

Functions that are not used in the contract, and make the code's size bigger.

```
sendValue  
isContract  
functionStaticCall  
functionDelegateCall  
functionCallWithValue  
functionCall  
_verifyCallResult  
...
```

Recommendation

Remove unused functions.

L13 - Divide before Multiply Operation

Criticality

minor

Location

contract.sol#L1329

Description

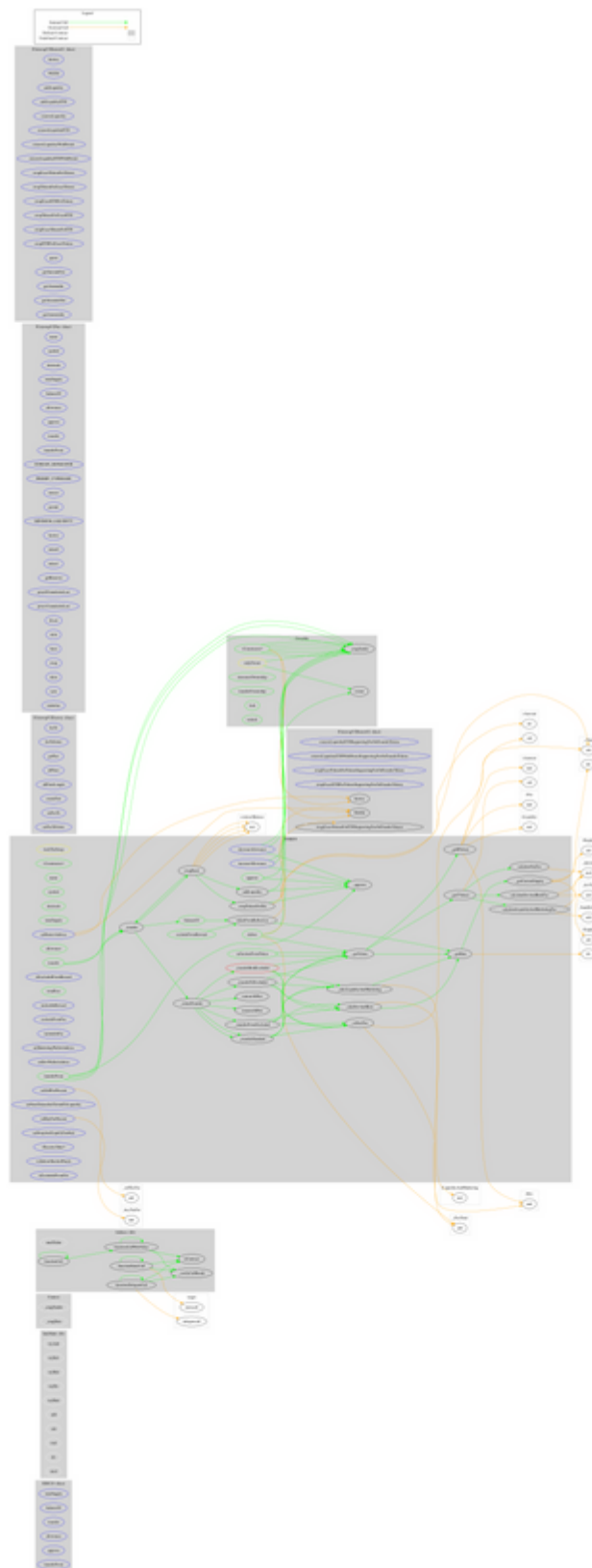
Performing divisions before multiplications may cause lose of prediction.

```
devTokens = contractBalance.mul(_sellDevFee).div(100)
marketingTokens = contractBalance.mul(_sellMarketingFee).div(100)
tokensForLiquidity = contractBalance.mul(_sellLiquidityFee).div(100)
```

Recommendation

The multiplications should be prior to the divisions.

Contract Flow



Domain Info

Domain Name	slumdogz-token.com
Registry Domain ID	2682308114_DOMAIN_COM-VRSN
Creation Date	2022-03-17T00:00:00Z
Updated Date	2022-03-17T00:00:00Z
Registry Expiry Date	2023-03-17T00:00:00Z
Registrar WHOIS Server	whois.cronon.net
Registrar URL	http://www.cronon.net
Registrar	Cronon AG
Registrar IANA ID	141

The domain has been created in 8 months before the creation of the audit.

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

Summary

SLUMDOGz is an interesting project that has a friendly and growing community. There is a function that can be abused by the owner to claim all the balance of the contract. A multi-wallet signing pattern will provide security against potential hacks. Temporarily locking the contract or renouncing ownership will eliminate the threat. 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>