



Cyberscope

Audit Report

Emergency Tasker Token

March 2022

Type BEP20

Network BSC

Address 0x85d819d4C6366Ad783F4611964B64c8B0dB8d817

Audited by © cyberscope

Table of Contents

Table of Contents	1
Contract Review	3
Audit Updates	3
Contract Analysis	4
Contract Diagnostics	5
CO - Code Optimization	6
Description	6
Recommendation	7
L01 - Public Function could be Declared External	8
Description	8
Recommendation	8
L02 - State Variables could be Declared Constant	9
Description	9
Recommendation	9
L04 - Conformance to Solidity Naming Conventions	10
Description	10
Recommendation	10
L09 - Dead Code Elimination	11
Description	11
Recommendation	11
L15 - Local Scope Variable Shadowing	12
Description	12
Recommendation	12
L08 - Tautology or Contradiction	13
Description	13
Recommendation	13

L13 - Divide before Multiply Operation	14
Description	14
Recommendation	14
Contract Functions	15
Contract Flow	20
Domain Info	21
Summary	22
Disclaimer	23
About Cyberscope	24

Contract Review

Contract Name	emergency
Compiler Version	v0.8.4+commit.c7e474f2
Optimization	200 runs
Licence	MIT
Explorer	https://bscscan.com/token/0x85d819d4C6366Ad783F4611964B64c8B0dB8d817
Symbol	ETT
Decimals	18
Total Supply	1,000,000,000,000,000
Source	emergency-imports.sol, emergency.sol
Domain	emergencytaskertoken.com

Audit Updates

Initial Audit	11th 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

Contract Diagnostics

● Critical ● Medium ● Minor

Severity	Code	Description
●	CO	Code Optimization
●	L01	Public Function could be Declared External
●	L02	State Variables could be Declared Constant
●	L04	Conformance to Solidity Naming Conventions
●	L09	Dead Code Elimination
●	L15	Local Scope Variable Shadowing
●	L08	Tautology or Contradiction
●	L13	Divide before Multiply Operation

CO - Code Optimization

Criticality	minor
Location	contract.sol#L1

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.

For instance, there are no arguments required in order to call the `_getAntiwhaleFees` function.

```
function _getSumOfFees(address sender, uint256 amount) internal view override
returns (uint256){
    return _getAntiwhaleFees(balanceOf(sender), amount);
}
////
function _getAntiwhaleFees(uint256, uint256) internal view returns (uint256){
    return sumOfFees;
}
```

The `_takeFeeToETH` calls the `_takeFee`, so there is not need for defining the ExternalToETH fee.

```
else if ( name == FeeType.ExternalToETH){
    _takeFeeToETH( amount, currentRate, value, recipient, index );
}
else {
    _takeFee( amount, currentRate, value, recipient, index );
}
////
function _takeFeeToETH(uint256 amount, uint256 currentRate, uint256 fee,
address recipient, uint256 index) private {
    _takeFee(amount, currentRate, fee, recipient, index);
}
```

Recommendation

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

L01 - Public Function could be Declared External

Criticality	minor
Location	emergency-imports.sol#L94,98,103,106,112,127 emergency.sol#L256,261,321

Description

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

```
isExcludedFromFee  
decreaseAllowance  
increaseAllowance  
manager  
unlock  
lock  
getUnlockTime  
transferOwnership  
renounceOwnership
```

Recommendation

Use the external attribute for functions never called from the contract

L02 - State Variables could be Declared Constant

Criticality	minor
Location	emergency.sol#L494,496,75,73,74

Description

Constant state variables should be declared constant to save gas.

```
marketing
foundations
burnAddress
_mainnetRouterV2Address
_mainnetRouterV1Address
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	emergency-imports.sol#L143 emergency.sol#L49,54,60,65,155,156,157,159,160,500 and 2 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.

```
emergency
_pair
_router
_isExcludedFromRewards
_isExcludedFromFee
_allowances
_balances
_reflectedBalances
numberOfTokensToSwapToLiquidity
...
```

Recommendation

Follow the Solidity naming convention.

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

L09 - Dead Code Elimination

Criticality	minor
Location	emergency-imports.sol#L66,41,42,43,44,58,61,50,53,36 and 1 more emergency.sol#L455,460,465,481,667,102,107,137

Description

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

```
getCollectedFeeTotal  
_addFees  
_addFee  
_approveDelegate  
_takeTransactionFees  
_isV2Pair  
_getSumOfFees  
_beforeTokenTransfer  
sendValue  
...
```

Recommendation

Remove unused functions.

L15 - Local Scope Variable Shadowing

Criticality

minor

Location

emergency.sol#L391,413,693,725

Description

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

```
name
_env
sumOfFees
```

Recommendation

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

L08 - Tautology or Contradiction

Criticality

minor

Location

emergency.sol#L123

Description

Detects expressions that are tautologies or contradictions. For instance, an uint variable will always be greater than or equal to zero.

```
require(bool,string)(index >= 0 && index <
fees.length,FeesSettings._getFeeStruct: Fee index out of bounds)
```

Recommendation

Fix the incorrect comparison by changing the value type or the comparison.

L13 - Divide before Multiply Operation

Criticality

minor

Location

emergency.sol#L419,470,747,755

Description

Performing divisions before multiplications may cause lose of prediction.

```
tAmount = amount.mul(fee).div(FEES_DIVISOR)
tBurn = amount.mul(fee).div(FEES_DIVISOR)
tFee = amount.mul(fee).div(FEES_DIVISOR)
tTotalFees = tAmount.mul(feesSum).div(FEES_DIVISOR)
```

Recommendation

The multiplications should be prior to the divisions.

Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
IERC20Metadata	Interface	IERC20		
	name	External		-
	symbol	External		-
	decimals	External		-
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
SafeMath	Library			
	add	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	mod	Internal		
	sub	Internal		
Address	Library			
	isContract	Internal		
	sendValue	Internal	✓	

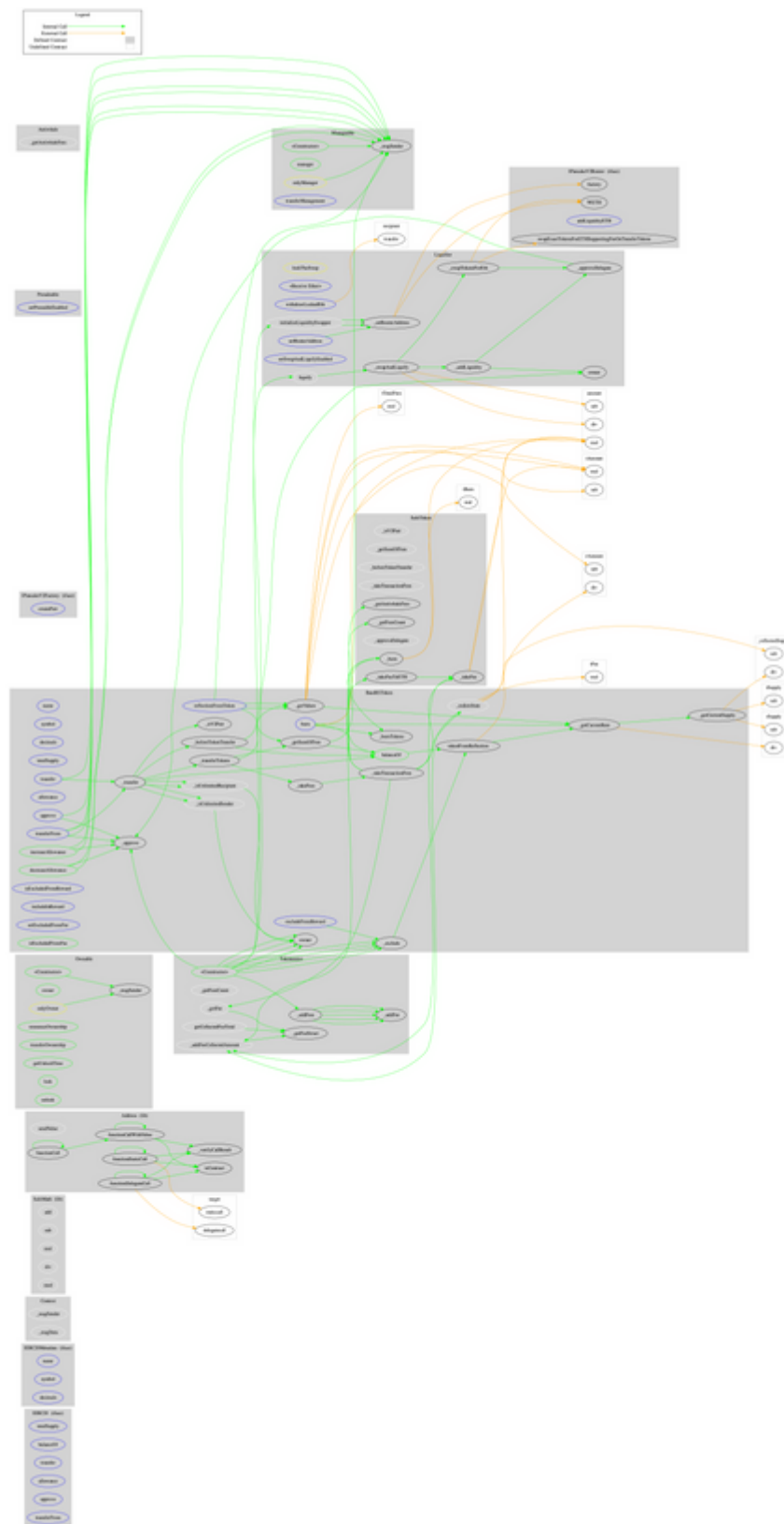
	functionCall	Internal	✓	
	functionCall	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionStaticCall	Internal		
	functionStaticCall	Internal		
	functionDelegateCall	Internal	✓	
	functionDelegateCall	Internal	✓	
	_verifyCallResult	Private		
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	getUnlockTime	Public		-
	lock	Public	✓	onlyOwner
	unlock	Public	✓	-
Manageable	Implementation	Context		
	<Constructor>	Public	✓	-
	manager	Public		-
	transferManagement	External	✓	onlyManager
IPancakeV2Factory	Interface			
	createPair	External	✓	-
IPancakeV2Router	Interface			
	factory	External		-
	WETH	External		-
	addLiquidityETH	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-

Tokenomics	Implementation			
	<Constructor>	Public	✓	-
	_addFee	Private	✓	
	_addFees	Private	✓	
	_getFeesCount	Internal		
	_getFeeStruct	Private		
	_getFee	Internal		
	_addFeeCollectedAmount	Internal	✓	
	getCollectedFeeTotal	Internal		
Presaleable	Implementation	Manageable		
	setPreseableEnabled	External	✓	onlyManager
BaseRfiToken	Implementation	IERC20, IERC20Meta data, Ownable, Presaleable, Tokenomics		
	<Constructor>	Public	✓	-
	name	External		-
	symbol	External		-
	decimals	External		-
	totalSupply	External		-
	balanceOf	Public		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
	burn	External	✓	-
	_burnTokens	Internal	✓	
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	isExcludedFromReward	External		-
	reflectionFromToken	External		-
	tokenFromReflection	Internal		
	excludeFromReward	External	✓	onlyOwner

	_exclude	Internal	✓	
	includeInReward	External	✓	onlyOwner
	setExcludedFromFee	External	✓	onlyOwner
	isExcludedFromFee	Public		-
	_approve	Internal	✓	
	_isUnlimitedSender	Internal		
	_isUnlimitedRecipient	Internal		
	_transfer	Private	✓	
	_transferTokens	Private	✓	
	_takeFees	Private	✓	
	_getValues	Internal		
	_getCurrentRate	Internal		
	_getCurrentSupply	Internal		
	_beforeTokenTransfer	Internal	✓	
	_getSumOfFees	Internal		
	_isV2Pair	Internal		
	_redistribute	Internal	✓	
	_takeTransactionFees	Internal	✓	
Liquifier	Implementation	Ownable, Manageable		
	<Receive Ether>	External	Payable	-
	initializeLiquiditySwapper	Internal	✓	
	liquify	Internal	✓	
	_setRouterAddress	Private	✓	
	_swapAndLiquify	Private	✓	lockTheSwap
	_swapTokensForEth	Private	✓	
	_addLiquidity	Private	✓	
	setRouterAddress	External	✓	onlyManager
	setSwapAndLiquifyEnabled	External	✓	onlyManager
	withdrawLockedEth	External	✓	onlyManager
	_approveDelegate	Internal	✓	
Antiwhale	Implementation	Tokenomics		
	_getAntiwhaleFees	Internal		

SafeToken	Implementation	BaseRfiToken, Liquifier, Antiwhale		
	<Constructor>	Public	✓	-
	_isV2Pair	Internal		
	_getSumOfFees	Internal		
	_beforeTokenTransfer	Internal	✓	
	_takeTransactionFees	Internal	✓	
	_burn	Private	✓	
	_takeFee	Private	✓	
	_takeFeeToETH	Private	✓	
	_approveDelegate	Internal	✓	
emergency	Implementation	SafeToken		
	<Constructor>	Public	✓	SafeToken

Contract Flow



Domain Info

Domain Name	
Registry Domain ID	2661099326_DOMAIN_COM-VRSN
Creation Date	2021-12-12T06:51:30.00Z
Updated Date	0001-01-01T00:00:00.00Z
Registry Expiry Date	2022-12-12T06:51:30.00Z
Registrar WHOIS Server	whois.namecheap.com
Registrar URL	http://www.namecheap.com
Registrar	NAMECHEAP INC
Registrar IANA ID	1068

The domain has been created 3 months before the creation of the audit. It will expire in 9 months.

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

Summary

The Smart Contract analysis reported no compiler error or critical issues. The contract Owner can access some admin functions that can not be used in a malicious way to disturb the users' transactions. The fees are fixed to 10%.

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.

Cyberscope 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>