Cyberscope

Audit Report

The ClubHouse Token

August 2022

Type	BEP20
Network	BSC
Address	0x31599da060c8e919729b6de167a1397b0c9de99e
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	ClubHouse
Compiler Version	v0.8.0+commit.c7dfd78e
Optimization	200 runs
Licence	None
Explorer	https://bscscan.com/token/0x31599Da060c8E919729B6De167A1397B0C9De99e
Symbol	TCH
Decimals	9
Total Supply	100,000,000

Source Files

Filename	**SHA256**
contract.sol	555fba434a27b14996a2ba0748c6a27e063b5da7f635831232ee7d98be36beb9

Audit Updates

Initial Audit	20th August 2022
Corrected	

Contract Analysis

Critical Medium Minor 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	Unresolved
●	MT	Mints Tokens	Passed
●	BT	Burns Tokens	Passed
●	BC	Blacklists Addresses	Unresolved

ULTW - Transfers Liquidity to Team Wallet

Criticality	minor
Location	contract.sol#L1075
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 `withdrawStuckBNB` method.

```
function withdrawStuckBNB() external onlyOwner {
    payable(msg.sender).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	medium
Location	contract.sol#L953
Status	Unresolved

Description

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

```
require(!blacklisted[from] && !blacklisted[to], "Transfer made by blacklisted address");
```

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	Status
●	ZD	Zero Division	Unresolved
●	CR	Code Repetition	Unresolved
●	L01	Public Function could be Declared External	Unresolved
●	L02	State Variables could be Declared Constant	Unresolved
●	L03	Redundant Statements	Unresolved
●	L04	Conformance to Solidity Naming Conventions	Unresolved
●	L07	Missing Events Arithmetic	Unresolved
●	L09	Dead Code Elimination	Unresolved

ZD - Zero Division

Criticality	critical
Location	contract.sol#L1006
Status	Unresolved

Description

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

```
uint256 tokensToLiquify =
contractBalance.mul(sellLiquidityFee).div(sellTotalFees).div(2);
```

Recommendation

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

CR - Code Repetition

Criticality	minor
Location	contract.sol#L989
Status	Unresolved

Description

There are code segments that are repetitive in the contract. Those segments increase the code size of the contract unnecessarily.

```
if(automatedMarketMaker[to]) {
    uint256 sellFees = amount.mul(sellTotalFees).div(100);
    amount = amount.sub(sellFees);
    super._transfer(from, address(this), sellFees);
}
//Buy
else {
    uint256 buyFees = amount.mul(buyTotalFees).div(100);
    amount = amount.sub(buyFees);
    super._transfer(from, address(this), buyFees);
}
```

Recommendation

Create an internal function that contains the code segment and remove it from all the sections. For instance, a takeFee() method could replace both statements.

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L280,288,305,312,331,339,350,368,390,409,565,574,876,883,890,896,902,908,914,921,931,941,945
Status	Unresolved

Description

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

```
name
symbol
decimals
totalSupply
transfer
allowance
approve
transferFrom
increaseAllowance
...
```

Recommendation

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

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L813
Status	Unresolved

Description

Constant state variables should be declared constant to save gas.

```
router
```

Recommendation

Add the constant attribute to state variables that never change.

L03 - Redundant Statements

Criticality	minor
Location	contract.sol#L148
Status	Unresolved

Description

The contract contains statements that are not used and have no effect. As a result, those segments increase the code size of the contract unnecessarily.

```
Context
```

Recommendation

Remove the redundant statements in order to decrease the code size.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L594,595,631,632,649,669,844,921,810,811,812,814,821,822,826
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.

```
DOMAIN_SEPARATOR
PERMIT_TYPEHASH
MINIMUM_LIQUIDITY
WETH
distributedProtocolBNB
_amm
_projectWallet
_operationsWallet
_stakingWallet
...
```

Recommendation

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

L07 - Missing Events Arithmetic

Criticality	minor
Location	contract.sol#L876,883,890,936
Status	Unresolved

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.

```
buyTotalFees = newBuyLiqFee.add(newBuyMarketingFee)
sellTotalFees = newsellLiqFee.add(newSellMarketingFee)
BNBThreshold = newThreshold * 10 ** 17
walletMax = newLimit * 10 ** 9
```

Recommendation

Emit an event for critical parameter changes.

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L473
Status	Unresolved

Description

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

```
_burn
```

Recommendation

Remove unused functions.

Contract Functions

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

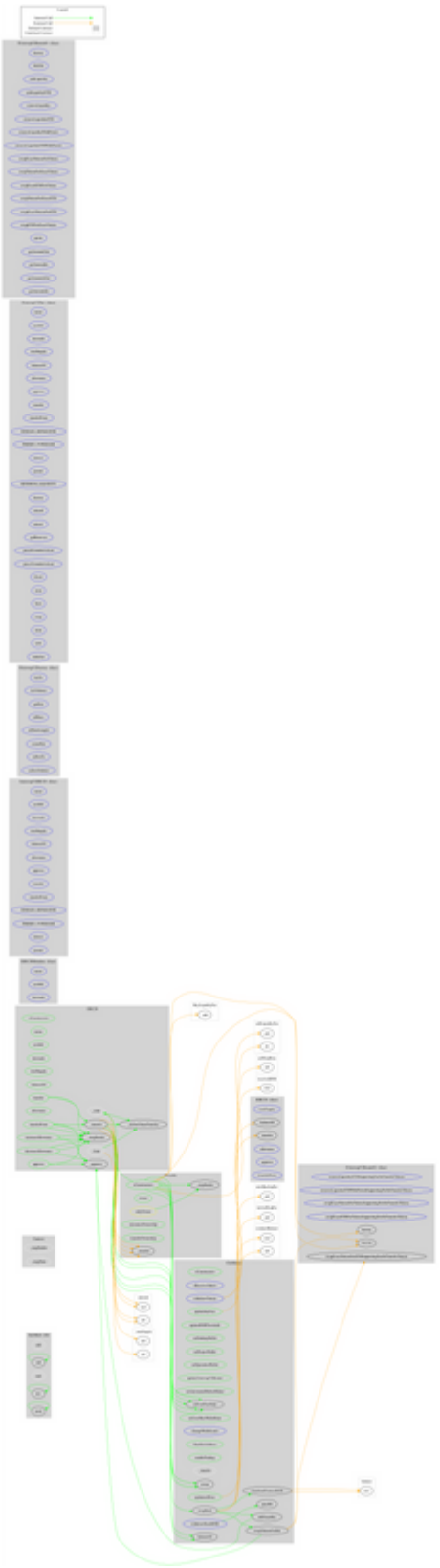
ERC20	Implementation	Context, IERC20, IERC20Metadata		
	<Constructor>	Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	_transfer	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_approve	Internal	✓	
	_beforeTokenTransfer	Internal	✓	
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
IuniswapV2ERC20	Interface			
	name	External		-
	symbol	External		-
	decimals	External		-
	totalSupply	External		-
	balanceOf	External		-
	allowance	External		-
	approve	External	✓	-

	transfer	External	✓	-
	transferFrom	External	✓	-
	DOMAIN_SEPARATOR	External		-
	PERMIT_TYPEHASH	External		-
	nonces	External		-
	permit	External	✓	-
IUniswapV2Factory	Interface			
	feeTo	External		-
	feeToSetter	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	createPair	External	✓	-
	setFeeTo	External	✓	-
	setFeeToSetter	External	✓	-
IUniswapV2Pair	Interface			
	name	External		-
	symbol	External		-
	decimals	External		-
	totalSupply	External		-
	balanceOf	External		-
	allowance	External		-
	approve	External	✓	-
	transfer	External	✓	-
	transferFrom	External	✓	-
	DOMAIN_SEPARATOR	External		-
	PERMIT_TYPEHASH	External		-
	nonces	External		-
	permit	External	✓	-
	MINIMUM_LIQUIDITY	External		-
	factory	External		-
	token0	External		-

	token1	External		-
	getReserves	External		-
	price0CumulativeLast	External		-
	price1CumulativeLast	External		-
	kLast	External		-
	mint	External	✓	-
	burn	External	✓	-
	swap	External	✓	-
	skim	External	✓	-
	sync	External	✓	-
	initialize	External	✓	-
IUniswapV2Router01	Interface			
	factory	External		-
	WETH	External		-
	addLiquidity	External	✓	-
	addLiquidityETH	External	Payable	-
	removeLiquidity	External	✓	-
	removeLiquidityETH	External	✓	-
	removeLiquidityWithPermit	External	✓	-
	removeLiquidityETHWithPermit	External	✓	-
	swapExactTokensForTokens	External	✓	-
	swapTokensForExactTokens	External	✓	-
	swapExactETHForTokens	External	Payable	-
	swapTokensForExactETH	External	✓	-
	swapExactTokensForETH	External	✓	-
	swapETHForExactTokens	External	Payable	-
	quote	External		-
	getAmountOut	External		-
	getAmountIn	External		-
	getAmountsOut	External		-
	getAmountsIn	External		-
IUniswapV2Router02	Interface	IUniswapV2Router01		

	removeLiquidityETHSupportingFeeOnTransferTokens	External	✓	-
	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External	✓	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
ClubHouse	Implementation	ERC20, Ownable		
	<Constructor>	Public	✓	ERC20
	<Receive Ether>	External	Payable	-
	updatebuyFees	Public	✓	onlyOwner
	updatesellFees	Public	✓	onlyOwner
	updateBNBThreshold	Public	✓	onlyOwner
	setStakingWallet	Public	✓	onlyOwner
	setProjectWallet	Public	✓	onlyOwner
	setOperationWallet	Public	✓	onlyOwner
	updateUniswapV2Router	Public	✓	onlyOwner
	setAutomatedMarketMaker	Public	✓	onlyOwner
	setUserFeesState	Public	✓	onlyOwner
	setUserMaxWalletState	Public	✓	onlyOwner
	changeWalletLimit	External	✓	onlyOwner
	blacklistAddress	Public	✓	onlyOwner
	enableTrading	Public	✓	onlyOwner
	_transfer	Internal	✓	
	swapBack	Internal	✓	
	addLiquidity	Private	✓	
	swapTokensForEth	Private	✓	
	distributeProtocolBNB	Private	✓	
	withdrawStuckBNB	External	✓	onlyOwner
	withdrawTokens	External	✓	onlyOwner

Contract Flow



Summary

There are some functions that can be abused by the owner like transferring funds to the team’s wallet and blacklisting addresses. 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 20% 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