Cyberscope

Audit Report

DOGEBEER

November 2022

SHA256 568658547032f3d107d5be852ced548a1f44f9253be05509aacaf50244ab9457

Audited by © cyberscope

Table of Contents

Contract Review

Contract Name	DOGEBEER
Compiler Version	v0.8.4+commit.c7e474f2
Optimization	200 runs
Licence	None
Symbol	BEERS
Decimals	9
Total Supply	420,000,000

Source Files

Filename	**SHA256**
contract.sol	568658547032f3d107d5be852ced548a1f44f9253be05509aacaf50244ab9457

Audit Updates

Initial Audit	23rd July 2022 https://github.com/cyberscope-io/audits/tree/main/beers/v1/audit.pdf
Corrected Phase 1	19th August 2022 https://github.com/cyberscope-io/audits/tree/main/beers/v2/audit.pdf
Corrected Phase 2	23rd August 2022 https://github.com/cyberscope-io/audits/tree/main/beers/v3/audit.pdf
Corrected Phase 3	17th November

Contract Analysis

● Critical ● Medium ● Minor / Informative ● 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	Passed

ULTW - Transfers Liquidity to Team Wallet

Criticality	minor / informative
Location	contract.sol#L1998
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 `Sweep` method.

```
function Sweep() external onlyOwner {
    uint256 balance = address(this).balance;
    payable(owner()).transfer(balance);
}
```

Recommendation

The contract could embody a check for the maximum amount of funds that can be swapped. Since a huge amount may volatile the token’s price.

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
●	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
●	L14	Uninitialized Variables in Local Scope	Unresolved

L01 - Public Function could be Declared External

Criticality	minor / informative
Location	contract.sol#L589,618,184,1087,622,594,600,1012,1071,585,636,199,189,571,559,614,1020,567,1016,653,210,605,610,626,203,580,195,563
Status	Unresolved

Description

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

```
approve
minimumTokensBeforeSwapAmount
renounceOwnership
GetSwapMinutes
buyBackSellLimitAmount
transferFrom
increaseAllowance
isExcludedFromFee
GetBuyBackTimeInterval
...
```

Recommendation

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

L02 - State Variables could be Declared Constant

Criticality	minor / informative
Location	contract.sol#L507,498,497
Status	Unresolved

Description

Constant state variables should be declared constant to save gas.

```
_isEnabledBuyBackAndBurn
_buyBackMaxTimeForHistories
_buyBackTimeInterval
```

Recommendation

Add the constant attribute to state variables that never change.

L03 - Redundant Statements

Criticality	minor / informative
Location	contract.sol#L20
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 / informative
Location	contract.sol#L1188,482,985,1169,498,1071,1194,477,487,432,1128,1215,1199,1087,467,507,256,272,449,470,497,1136,991,255,496,448,1146,494,480,474,1091,476,450,479,1207,473,1067,293,489,1082,485,1132,444,1141,495
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.

```
_to
_buyBackRangeRate
_amount
_router
_buyBackMaxTimeForHistories
GetBuyBackTimeInterval
Sweep
_sellLiquidityFee
LPDivisor
...
```

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 / informative
Location	contract.sol#L1128,1104,1115,1099,1119,1091,1067,1095,1082,1124,1111
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.

```
minimumTokensBeforeSwap = _minimumTokensBeforeSwap
_sellTaxFee = sellTaxFee
buyBackSellLimit = buyBackSellSetLimit
_buyTaxFee = buyTaxFee
_maxTxAmount = maxTxAmount
_intervalMinutesForSwap = newMinutes * 60
_buyBackDivisor = newDivisor
_taxFee = taxFee
_buyBackRangeRate = newPercent
...
```

Recommendation

Emit an event for critical parameter changes.

L09 - Dead Code Elimination

Criticality	minor / informative
Location	contract.sol#L133,137,142,102,129,125
Status	Unresolved

Description

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

```
functionCallWithValue
_functionCallWithValue
isContract
functionCall
```

Recommendation

Remove unused functions.

L14 - Uninitialized Variables in Local Scope

Criticality	minor / informative
Location	contract.sol#L700
Status	Unresolved

Description

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

```
sellHistory
```

Recommendation

All the local scoped variables should be initialized.

Contract Functions

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

	_functionCallWithValue	Private	✓	
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	getUnlockTime	Public		-
	getTime	Public		-
	lock	Public	✓	onlyOwner
	unlock	Public	✓	-
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		-

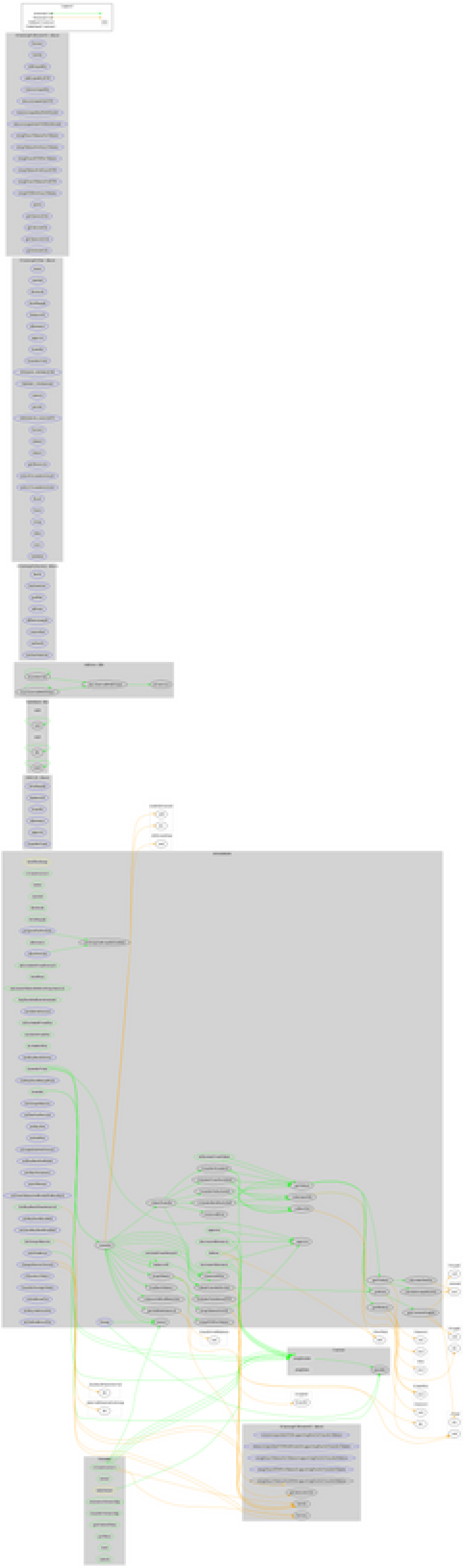
	PERMIT_TYPEHASH	External		-
	nonces	External		-
	permit	External	✓	-
	MINIMUM_LIQUIDITY	External		-
	factory	External		-
	token0	External		-
	token1	External		-
	getReserves	External		-
	price0CumulativeLast	External		-
	price1CumulativeLast	External		-
	kLast	External		-
	burn	External	✓	-
	swap	External	✓	-
	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	✓	-
DOGEBEER	Implementation	Context, IERC20, Ownable		
	<Constructor>	Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	isExcludedFromReward	Public		-
	totalFees	Public		-
	minimumTokensBeforeSwapAmount	Public		-
	buyBackSellLimitAmount	Public		-
	deliver	Public	✓	-
	reflectionFromToken	Public		-

	tokenFromReflection	Public		-
	excludeFromReward	Public	✓	onlyOwner
	includeInReward	External	✓	onlyOwner
	_approve	Private	✓	
	_transfer	Private	✓	
	swapTokens	Private	✓	lockTheSwap
	buyBackTokens	Private	✓	lockTheSwap
	swapTokensForEth	Private	✓	
	swapETHForTokens	Private	✓	
	_tokenTransfer	Private	✓	
	_transferStandard	Private	✓	
	_transferToExcluded	Private	✓	
	_transferFromExcluded	Private	✓	
	_transferBothExcluded	Private	✓	
	_reflectFee	Private	✓	
	_getValues	Private		
	_getTValues	Private		
	_getRValues	Private		
	_getRate	Private		
	_getCurrentSupply	Private		
	_takeLiquidity	Private	✓	
	calculateTaxFee	Private		
	calculateLiquidityFee	Private		
	removeAllFee	Private	✓	
	restoreAllFee	Private	✓	
	isExcludedFromFee	Public		-
	excludeFromFee	Public	✓	onlyOwner
	includeInFee	Public	✓	onlyOwner
	_getSellBnBAmount	Private		
	_removeOldSellHistories	Private	✓	
	SetBuyBackDivisor	External	✓	onlyOwner
	GetBuyBackTimeInterval	Public		-
	SetBuyBackRangeRate	External	✓	onlyOwner
	GetSwapMinutes	Public		-

	SetSwapMinutes	External	✓	onlyOwner
	setTaxFeePercent	External	✓	onlyOwner
	setBuyFee	External	✓	onlyOwner
	setSellFee	External	✓	onlyOwner
	setLiquidityFeePercent	External	✓	onlyOwner
	setBuyBackSellLimit	External	✓	onlyOwner
	setMaxTxAmount	External	✓	onlyOwner
	setLPDivisor	External	✓	onlyOwner
	setNumTokensSellToAddToBuyBack	External	✓	onlyOwner
	setLPAddress	External	✓	onlyOwner
	setSwapAndLiquifyEnabled	Public	✓	onlyOwner
	setBuyBackEnabled	External	✓	onlyOwner
	setAutoBuyBackEnabled	External	✓	onlyOwner
	prepareForPreSale	External	✓	onlyOwner
	afterPreSale	External	✓	onlyOwner
	transferToAddressETH	Private	✓	
	changeRouterVersion	External	✓	onlyOwner
	<Receive Ether>	External	Payable	-
	transferForeignToken	External	✓	onlyOwner
	Sweep	External	✓	onlyOwner
	setAddressFee	External	✓	onlyOwner
	setBuyAddressFee	External	✓	onlyOwner
	setSellAddressFee	External	✓	onlyOwner

Contract Flow



Summary

The Smart Contract analysis reported one minor severity issue. The contract owner has the authority to transfer funds to the team's wallet. Other than that, the contract owner can access some admin functions that can not be used in a malicious way to disturb the users' transactions. There is also a max limit of 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