Cyberscope

Audit Report

Husky Inu

March 2022

Type	BEP20
Network	BSC
Address	0x40a5Ad300Ee5a5b8bA6B50719647C19F2E52d5BE
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	HuskyInu
Compiler Version	v0.8.13+commit.abaa5c0e
Optimization	200 runs
Licence	MIT
Explorer	https://bscscan.com/token/0x40a5Ad300Ee5a5b8bA6B50719647C19F2E52d5BE
Symbol	HUSKY
Decimals	18
Total Supply	10,000,000,000
Domain	huskyinubsc.finance

Source Files

Filename	**SHA256**
contract.sol	9f8bb736a9582f7d2fd953cb3099bc9436a81507f793b5f16c43ad372f8d2de5

Audit Updates

Initial Audit	29th 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

ST - Stop Transactions

Criticality	critical
Location	contract.sol#L1210

Description

The contract owner has the authority to stop transactions for all users excluding the owner. The owner may take advantage of it by setting the limitSell to true and minSell to zero in the setLimitSell function.

```
if(recipient == uniswapV2Pair && // Swap Tokens to BNB (SELL)
            sender != address(this) &&
            sender != owner() &&
            !excludeFee[sender]){
            bool allowToSell = true;

            if(limitSell == true) {
                if(amount < minSell*10**18 || amount > maxSell*10**18) {
                    allowToSell = false;
                }
            }

            if(allowToSell == true) {
                if(taxStatus == true){
```

Recommendation

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

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 - Blacklisted Contracts

Criticality	medium
Location	contract.sol#L1200

Description

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

```
require(!blacklist[sender] && !blacklist[recipient], "BLACKLISTED");
```

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
	MAL	Misused Algorithmic Logic
	L01	Public Function could be Declared External
	L02	State Variables could be Declared Constant
	L04	Conformance to Solidity Naming Conventions
	L07	Missing Events Arithmetic
	L09	Dead Code Elimination
	L11	Unnecessary Boolean equality
	L15	Local Scope Variable Shadowing

MAL - Misused Algorithmic Logic

Criticality	minor
Location	contract.sol#L1231

Description

The contract transfers tokens to the usiswapV2pair. Usually the uniswapV2pair is used for inserting liquidity. The tokens that are accumulated in the pair are useless.

```
super._transfer(sender, uniswapV2Pair, _LiquidityFee);
```

Recommendation

The contract should add liquidity to the pair instead of using tokens.

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L800,808,893,900,907,914,921

Description

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

```
balanceOf
totalSupply
symbol
decimals
name
transferOwnership
renounceOwnership
```

Recommendation

Use the external attribute for functions never called from the contract

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L1144,1143,1149,1142,1140,1136,1150,1141,1139

Description

Constant state variables should be declared constant to save gas.

```
rewardSellFee
rewardBuyFee
rewardAddress
maxSupply
marketingSellFee
marketingBuyFee
marketingAddress
liquidityFee
burnFee
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L27,1164,1170

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.

```
_min
_max
WETH
```

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#L1164,1170

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.

```
maxBuy = _max
maxSell = _max
```

Recommendation

Emit an event for critical parameter changes.

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L350,360,379,393,439,449,412,422,325,466 and 1 more

Description

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

```
_burnFrom
verifyCallResult
sendValue
functionStaticCall
functionDelegateCall
functionCallWithValue
functionCall
...
```

Recommendation

Remove unused functions.

L11 - Unnecessary Boolean equality

Criticality	minor
Location	contract.sol#L1192 and 1 more

Description

The comparison to boolean constants is redundant. Boolean constants can be used directly and do not need to be compared to true or false.

```
allowToSell == true
require(bool,string)(allowToTransfer == true,Not Allow to Transfer)
limitSell == true
preventBotContract == true && sender == uniswapV2Pair && recipient != address(this)
taxStatus == true
allowToTransfer == true
preventBotContract == true && recipient == uniswapV2Pair && sender != address(this)
sender.isContract() == true
limitBuy == true
...
```

Recommendation

Remove the equality to the boolean constant.

L15 - Local Scope Variable Shadowing

Criticality	minor
Location	contract.sol#L876,1158

Description

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

```
_symbol
_name
symbol
name
```

Recommendation

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

Contract Functions

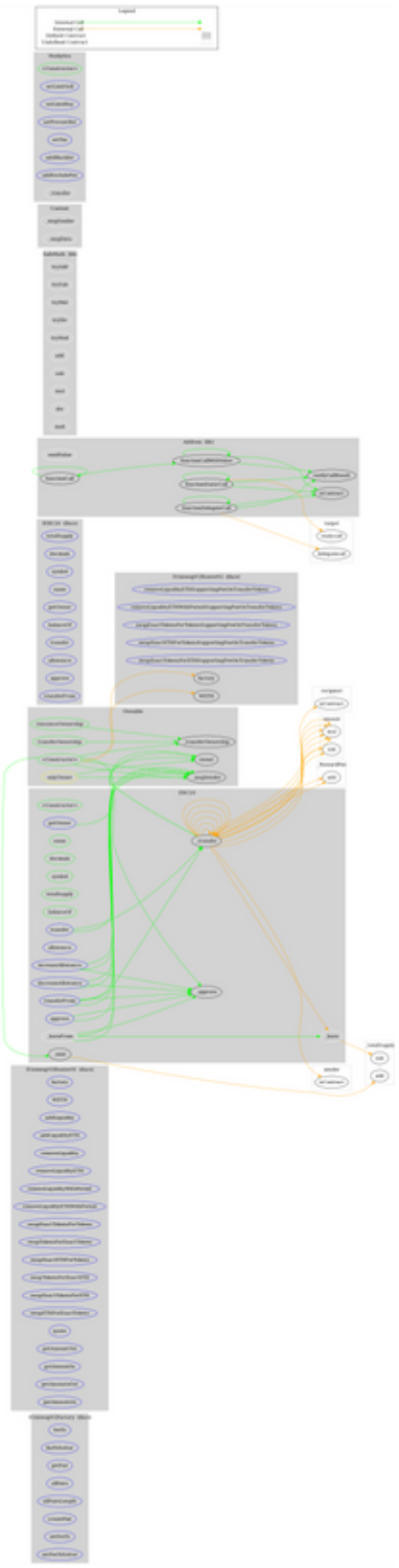
Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
IUniswapV2Factory	Interface			
	feeTo	External		-
	feeToSetter	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	createPair	External	✓	-
	setFeeTo	External	✓	-
	setFeeToSetter	External	✓	-
IUniswapV2Router01	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	✓	-
IERC20	Interface			
	totalSupply	External		-
	decimals	External		-
	symbol	External		-
	name	External		-
	getOwner	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
Address	Library			
	isContract	Internal		
	sendValue	Internal	✓	
	functionCall	Internal	✓	
	functionCall	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionStaticCall	Internal		

	functionStaticCall	Internal		
	functionDelegateCall	Internal	✓	
	functionDelegateCall	Internal	✓	
	verifyCallResult	Internal		
SafeMath	Library			
	tryAdd	Internal		
	trySub	Internal		
	tryMul	Internal		
	tryDiv	Internal		
	tryMod	Internal		
	add	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	mod	Internal		
	sub	Internal		
	div	Internal		
	mod	Internal		
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_transferOwnership	Internal	✓	
ERC20	Implementation	Context, IERC20, Ownable		
	<Constructor>	Public	✓	-
	getOwner	External		-

	name	Public		-
	decimals	Public		-
	symbol	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
	increaseAllowance	External	✓	-
	decreaseAllowance	External	✓	-
	_transfer	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_approve	Internal	✓	
	_burnFrom	Internal	✓	
Huskylnu	Implementation	ERC20		
	<Constructor>	Public	✓	ERC20
	setLimitSell	External	✓	onlyOwner
	setLimitBuy	External	✓	onlyOwner
	setPreventBot	External	✓	onlyOwner
	setTax	External	✓	onlyOwner
	addBlacklist	External	✓	onlyOwner
	addExcludeFee	External	✓	onlyOwner
	_transfer	Internal	✓	

Contract Flow



Domain Info

Domain Name	huskyinubsc.finance
Registry Domain ID	0fa5adfa927e417bbff491aac65745db-DONUTS
Creation Date	2022-03-29T16:04:05Z
Updated Date	2022-03-29T16:04:08Z
Registry Expiry Date	2023-03-29T16:04:05Z
Registrar WHOIS Server	
Registrar URL	
Registrar	
Registrar IANA ID	

The domain has been created about 2 hours 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 blacklisting wallets and stopping transactions. The contract can be converted to honeypot and prevent the 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.

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 provides all the essential tools to assist users draw their own conclusions.

The Cyberscope team

https://www.cyberscope.io