Cyberscope

Audit Report

DemonHellboy

June 2022

Type	BEP20
Network	BSC
Address	0xa604E4F3037c0D4818854aDa78D3503Ea1500b07
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	DemonHellboy
Compiler Version	v0.8.4+commit.c7e474f2
Optimization	200 runs
Licence	Unlicense
Explorer	https://bscscan.com/token/0xa604E4F3037c0D4818854aDa78D3503Ea1500b07
Symbol	DHB
Decimals	9
Total Supply	99,999,999,999,999
Domain	demonhellboy.us

Source Files

Filename	**SHA256**
contract.sol	4e70041f7d9104bc5b4b3751bdbb54a9eee76f5ffff5ff6e2f12d688d088c90a

Audit Updates

Initial Audit	14th June 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#L1117, 1104

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 `_maxTxAmount` to zero.

```
 if(from != owner() && to != owner()) {
            require(amount <= _maxTxAmount, "Transfer amount exceeds the 
maxTxAmount.");
        }
```

The contract owner has also the authority to stop the sales for all users excluding the owner. He can take advantage of it by setting the sellfees to very high values. This will convert the contract into a **HONEYPOT**.

```
    function setSellFee(uint256 sellTaxFee, uint256 sellLiquidityFee) external 
onlyOwner {
        _sellTaxFee = sellTaxFee;
        _sellLiquidityFee = sellLiquidityFee;
    }
```

Recommendation

The contract could embody a check for not allowing setting the _maxTxAmount less than a reasonable amount or not allowing the sellfees to be very high values. A suggested implementation could check that the maximum 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.

ELFM - Exceed Limit Fees Manipulation

Criticality	critical
Location	contract.sol#L1104,1121

Description

The contract owner has the authority to increase over the allowed limit of 25%. The owner may take advantage of it by calling the `setLiquidityFeePercent` or `setMarketingDivisor` function with a high percentage value.

```
   function setLiquidityFeePercent(uint256 liquidityFee) external onlyOwner {
        _liquidityFee = liquidityFee;
    }
```

```
        transferToAddressETH(marketingAddress,
transferredBalance.mul(marketingDivisor).div(100));
```

Recommendation

The contract could embody a check for the maximum acceptable value.

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.

ULTW - Unlimited Liquidity to Team Wallet

Criticality	minor
Location	contract.sol#L1191

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

Severity	Code	Description
🟠	MC	Missing Check
⚪	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
⚪	L14	Uninitialized Variables in Local Scope

MC - Missing Check

Criticality	medium
Location	contract.sol#L1044,744

Description

The contract is processing variables that have not properly sanitized and checked that they form the proper shape. These variables may produce vulnerability issues. If _buyBackTimeInterval or _buyBackMaxTimeForHistories are greater than block.timestamp then the transaction will revert.

```
uint256 startTime = block.timestamp - _buyBackTimeInterval;

uint256 maxStartTimeForHistories = block.timestamp -
_buyBackMaxTimeForHistories;
```

Recommendation

The contract should properly check the variables according to the required specifications.

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L193,198,204,208,212,219,568,572,576,580,589,594,598,603,609,614,619,623,627,631,635,645,662,1018,1022,1026,1074,1087,1138,1143,1166,1185

Description

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

```
transferForeignToken
changeRouterVersion
setAutoBuyBackEnabled
setBuyBackEnabled
GetSwapMinutes
GetBuyBackTimeInterval
includeInFee
excludeFromFee
isExcludedFromFee
...
```

Recommendation

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

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L459,516,457,458,453

Description

Constant state variables should be declared constant to save gas.

```
_tTotal
_symbol
_name
_isEnabledBuyBackAndBurn
_decimals
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L264,265,281,302,991,997,1066,1070,1074,1078,1082,1087,1091,1125,1129,1133,1138,1143,1166,1185,1191,1196,1202,1208,476,479,482,483,485,486,488,489,491,494,498,503,504,505,506,507,516

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.

```
_isEnabledBuyBackAndBurn
_buyBackMaxTimeForHistories
_buyBackTimeInterval
_buyBackDivisor
_isAutoBuyBack
_sellHistories
_maxTxAmount
_addressFees
_buyBackRangeRate
...
```

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#L1066,1070,1078,1082,1091,1095,1099,1104,1109,1113,1117,1121,1125

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
marketingDivisor = divisor
_maxTxAmount = maxTxAmount
buyBackSellLimit = buyBackSellSetLimit
_liquidityFee = liquidityFee
_sellTaxFee = sellTaxFee
_buyTaxFee = buyTaxFee
_taxFee = taxFee
_intervalMinutesForSwap = newMinutes * 60
...
```

Recommendation

Emit an event for critical parameter changes.

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L151,134,138,142,146,114,125,867

Description

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

```
addLiquidity
sendValue
isContract
functionCallWithValue
functionCall
_functionCallWithValue
```

Recommendation

Remove unused functions.

L14 - Uninitialized Variables in Local Scope

Criticality	minor
Location	contract.sol#L709

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		
	sendValue	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		-

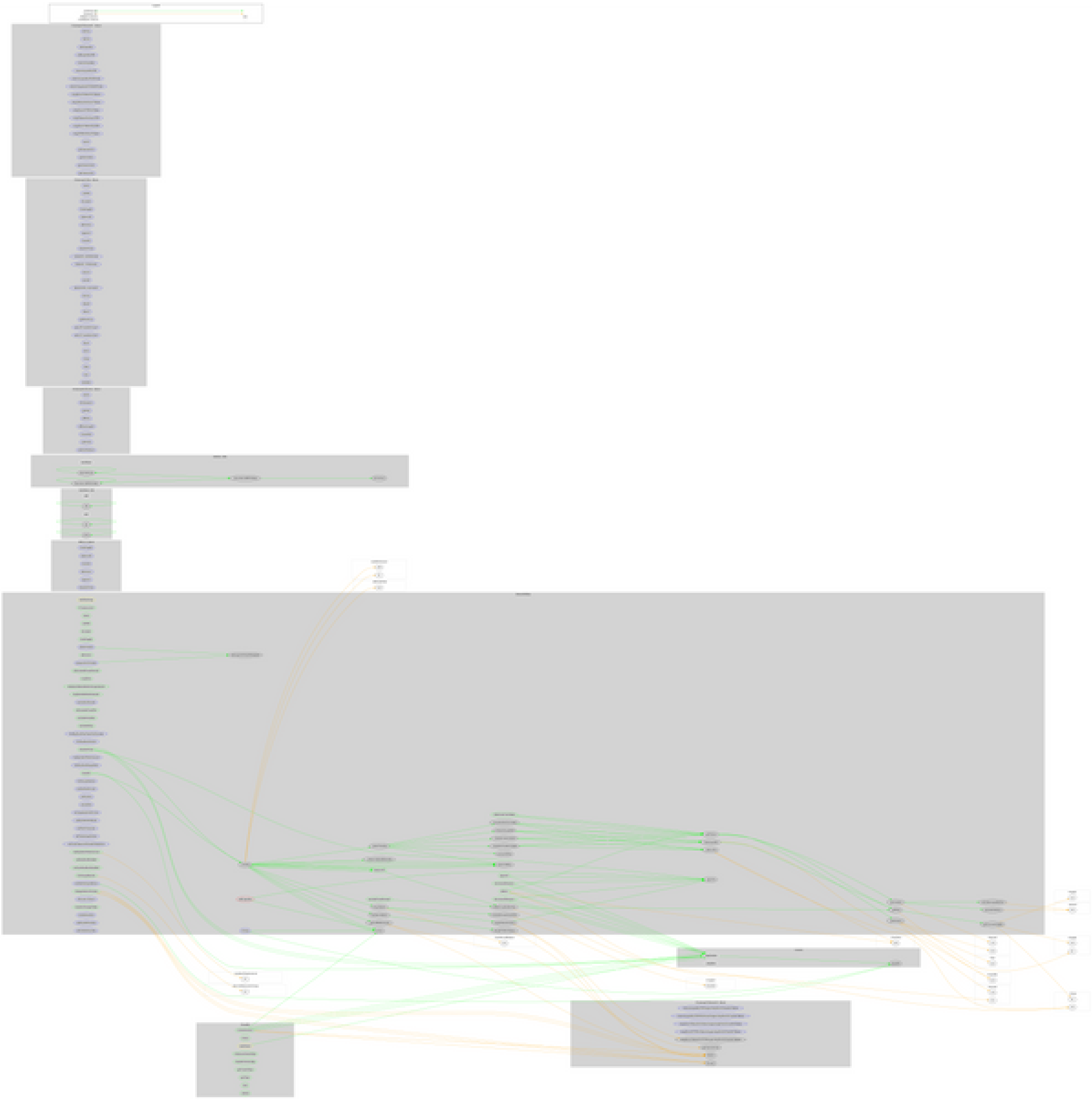
	nonces	External		-
	permit	External	✓	-
	MINIMUM_LIQUIDITY	External		-
	factory	External		-
	token0	External		-
	token1	External		-
	getReserves	External		-
	price0CumulativeLast	External		-
	price1CumulativeLast	External		-
	kLast	External		-
	burn	External	✓	-
	swap	External	✓	-
	skim	External	✓	-
	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	✓	-
DemonHellboy	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	✓	
	addLiquidity	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	✓	
	SetBuyBackMaxTimeForHistories	External	✓	onlyOwner
	SetBuyBackDivisor	External	✓	onlyOwner
	GetBuyBackTimeInterval	Public		-
	SetBuyBackTimeInterval	External	✓	onlyOwner
	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
	setMarketingDivisor	External	✓	onlyOwner
	setNumTokensSellToAddToBuyBack	External	✓	onlyOwner
	setMarketingAddress	External	✓	onlyOwner
	setSwapAndLiquifyEnabled	Public	✓	onlyOwner
	setBuyBackEnabled	Public	✓	onlyOwner
	setAutoBuyBackEnabled	Public	✓	onlyOwner
	prepareForPreSale	External	✓	onlyOwner
	afterPreSale	External	✓	onlyOwner
	transferToAddressETH	Private	✓	
	changeRouterVersion	Public	✓	onlyOwner
	<Receive Ether>	External	Payable	-
	transferForeignToken	Public	✓	onlyOwner
	Sweep	External	✓	onlyOwner
	setAddressFee	External	✓	onlyOwner
	setBuyAddressFee	External	✓	onlyOwner
	setSellAddressFee	External	✓	onlyOwner

Contract Flow



Domain Info

Domain Name	demonhellboy.us
Registry Domain ID	D3291C20469B347EEB3735A6C9F5304BB-GDREG
Creation Date	2022-06-10T15:18:39Z
Updated Date	2022-06-12T04:10:42Z
Registry Expiry Date	2023-06-10T15:18:39Z
Registrar WHOIS Server	whois.namecheap.com
Registrar URL	http://www.namecheap.com
Registrar	NameCheap, Inc.
Registrar IANA ID	1068

The domain has been created 4 days 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 stopping transactions for everyone except the owner, transferring funds from accumulated fees into the team wallet and manipulating fees up to 100%. The contract can also be converted into a **honeypot** and prevent users from selling if the configuration is abused. 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 provide all the essential tools to assist users draw their own conclusions.

The Cyberscope team

https://www.cyberscope.io