Cyberscope

Audit Report

Boa hancock inu

April 2022

SHA256 61e85f9ef29f38f23621c2aaa500bbbe99799f5e591999e0523f6411b6ebfb67

Audited by © cyberscope

Table of Contents

Source Files

Filename	SHA256
contract.sol	61e85f9ef29f38f23621c2aaa500bbbe99799f5e591999e0523f6411b6ebfb67

Audit Updates

Initial Audit	27th April 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#L3090,3101

Description

The contract owner has the authority to stop the sales for all users excluding the owner. The owner may take advantage of it by setting the `maxBnbUserCanSell` to zero.

```
canSwap(amount, maxBnbUserCanSell);
```

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 `isOpen` to false.

```
require(isOpen || _whiteList[from] || _whiteList[to], "Not Open");
```

Recommendation

The contract could embody a check for not allowing setting the _maxTxAmount less than a reasonable amount. 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#L2748,2759

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 `setSellOutTaxPercentage` function with a high percentage value.

```
function setSellOutTaxPercentage(uint8 percent)
    external
    onlyRole(OPERATOR_ROLE)
    whenPaused
{
    SELL_OUT_TAX_PERCENTAGE = percent;
}
```

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.

BC - Blacklisted Contracts

Criticality	critical
Location	contract.sol#L2561

Description

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

```
function includeToBlackList(address[] memory _users) external
onlyRole(OPERATOR_ROLE){
    for (uint8 i = 0; i < _users.length; i++) {
        _blackList[_users[i]] = true;
    }
}
```

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
	L01	Public Function could be Declared External
	L04	Conformance to Solidity Naming Conventions
	L05	Unused State Variable
	L09	Dead Code Elimination
	L11	Unnecessary Boolean equality
	L12	Using Variables before Declaration
	L14	Uninitialized Variables in Local Scope

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L78,86,348,374,382,1862,1875,1893,2141,2149,2173,2180,2215,2260,2280,2557,2567,2641,2673,2694,2704,2714,2724,2734,2740,2779,2783,2848,2862,3021,3027,3035,3050,3058

Description

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

```
closeAutoAddLP
openAutoAddLP
closeBuyBack
openBuyBack
setmaxBnbUserCanBuySell
transferOnlyDev
getWhiteList
unpause
pause
...
```

Recommendation

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

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L177,410,411,428,1339,1342,1529,1555,1558,1639,1561,1666,1669,1683,1705,1708,1723,1777,1780,1961,1998,2002,2066,2129,2133,2468,2109,2111,2477,2482,2512,2551,2561,2529,2530,2674,2675,2676,2677,2767,2773,2788,2794,2801,2806,2593,2594,2609

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.

```
deadWallet
SELL_OUT_TAX_PERCENTAGE
BUY_IN_TAX_PERCENTAGE
_maker
_value
_clearedUser
_evilUser
_manualBuyBackAddress
_autoBuyBackAddress
...
```

Recommendation

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

L05 - Unused State Variable

Criticality	minor
Location	contract.sol#L1961

Description

There are segments that contain unused state variables.

```
__gap
```

Recommendation

Remove unused state variables.

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L1780,1926,1917,1024,1034,1053,1067,1086,1096,999,1705,1666,1669,1339,1342,1457,1440,1476,1447,1483,1498,2504,2305,1224,1181,1205,1196,1157,1165,692,701,819,794,1558,1632

Description

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

```
_authorizeUpgrade
__UUPSUpgradeable_init_unchained
toString
toHexString
getUint256Slot
getBytes32Slot
safeTransferFrom
safeTransfer
safeIncreaseAllowance
...
```

Recommendation

Remove unused functions.

L11 - Unnecessary Boolean equality

Criticality	minor
Location	contract.sol#L3066

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.

```
address(lpAddress).balance > autoAddETHThrehold.mul(110).div(100) &&
isActiveAutoLP == true
```

Recommendation

Remove the equality to the boolean constant.

L12 - Using Variables before Declaration

Criticality	minor
Location	contract.sol#L1416

Description

The contract is using a variable before the declaration. This is usually happening either if it has not been declared yet or the variable has been declared in a different scope.

```
slot
```

Recommendation

The variables should be declared before any usage of them.

L14 - Uninitialized Variables in Local Scope

Criticality	minor
Location	contract.sol#L1416

Description

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

```
slot
```

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		
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_transferOwnership	Internal	✓	
IBEP20	Interface			
	totalSupply	External		-
	decimals	External		-
	symbol	External		-
	name	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
IPancakeFactory	Interface			
	feeTo	External		-
	feeToSetter	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-

	createPair	External	✓	-
	setFeeTo	External	✓	-
	setFeeToSetter	External	✓	-
IPancakeRouter01	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		-
IPancakeRouter02	Interface	IPancakeRouter01		
	removeLiquidityETHSupportingFeeOnTransferTokens	External	✓	-
	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External	✓	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-

ILiquidPool	Interface			
	addLiquidity	External	✓	-
LiquidPool	Implementation	Ownable		
	<Receive Ether>	External	Payable	-
	<Constructor>	Public	✓	-
	setTokenAddress	Public	✓	onlyOwner
	addLiquidity	External	✓	-
	balances	Public		-
	withdrawBNB	Public	✓	onlyOwner
	withdrawToken	Public	✓	onlyOwner
IPancakePair	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	✓	-
SafeMath	Library			
	add	Internal		
	sub	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
IERC20Upgradeable	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
IERC20MetadataUpgradeable	Interface	IERC20Upgradeable		
	name	External		-
	symbol	External		-
	decimals	External		-
StorageSlotUpgradeable	Library			
	getAddressSlot	Internal		
	getBooleanSlot	Internal		
	getBytes32Slot	Internal		
	getUint256Slot	Internal		

IBeaconUpgradeable	Interface			
	implementation	External		-
IERC1822ProxiableUpgradeable	Interface			
	proxiableUUID	External		-
IERC165Upgradeable	Interface			
	supportsInterface	External		-
StringsUpgradeable	Library			
	toString	Internal		
	toHexString	Internal		
	toHexString	Internal		
IAccessControlUpgradeable	Interface			
	hasRole	External		-
	getRoleAdmin	External		-
	grantRole	External	✓	-
	revokeRole	External	✓	-
	renounceRole	External	✓	-
AddressUpgradeable	Library			
	isContract	Internal		
	sendValue	Internal	✓	
	functionCall	Internal	✓	
	functionCall	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionStaticCall	Internal		
	functionStaticCall	Internal		

	verifyCallResult	Internal		
SafeERC20Upgradeable	Library			
	safeTransfer	Internal	✓	
	safeTransferFrom	Internal	✓	
	safeApprove	Internal	✓	
	safeIncreaseAllowance	Internal	✓	
	safeDecreaseAllowance	Internal	✓	
	_callOptionalReturn	Private	✓	
Initializable	Implementation			
	_isConstructor	Private		
ERC1967UpgradeUpgradeable	Implementation	Initializable		
	__ERC1967Upgrade_init	Internal	✓	onlyInitializing
	__ERC1967Upgrade_init_unchained	Internal	✓	onlyInitializing
	_getImplementation	Internal		
	_setImplementation	Private	✓	
	_upgradeTo	Internal	✓	
	_upgradeToAndCall	Internal	✓	
	_upgradeToAndCallUUPS	Internal	✓	
	_getAdmin	Internal		
	_setAdmin	Private	✓	
	_changeAdmin	Internal	✓	
	_getBeacon	Internal		
	_setBeacon	Private	✓	
	_upgradeBeaconToAndCall	Internal	✓	
	_functionDelegateCall	Private	✓	
UUPSUpgradeable	Implementation	Initializable, IERC1822ProxiableUpgradeable, ERC1967UpgradeUpgra		

		deable		
	__UUPSUpgradeable_init	Internal	✓	onlyInitializing
	__UUPSUpgradeable_init_unchained	Internal	✓	onlyInitializing
	proxiableUUID	External		notDelegated
	upgradeTo	External	✓	onlyProxy
	upgradeToAndCall	External	Payable	onlyProxy
	_authorizeUpgrade	Internal	✓	
ERC165Upgradeable	Implementation	Initializable, IERC165Upgradeable		
	__ERC165_init	Internal	✓	onlyInitializing
	__ERC165_init_unchained	Internal	✓	onlyInitializing
	supportsInterface	Public		-
ContextUpgradeable	Implementation	Initializable		
	__Context_init	Internal	✓	onlyInitializing
	__Context_init_unchained	Internal	✓	onlyInitializing
	_msgSender	Internal		
	_msgData	Internal		
AccessControlUpgradeable	Implementation	Initializable, ContextUpgradeable, IAccessControlUpgradeable, ERC165Upgradeable		
	__AccessControl_init	Internal	✓	onlyInitializing
	__AccessControl_init_unchained	Internal	✓	onlyInitializing
	supportsInterface	Public		-
	hasRole	Public		-
	_checkRole	Internal		
	getRoleAdmin	Public		-
	grantRole	Public	✓	onlyRole
	revokeRole	Public	✓	onlyRole
	renounceRole	Public	✓	-

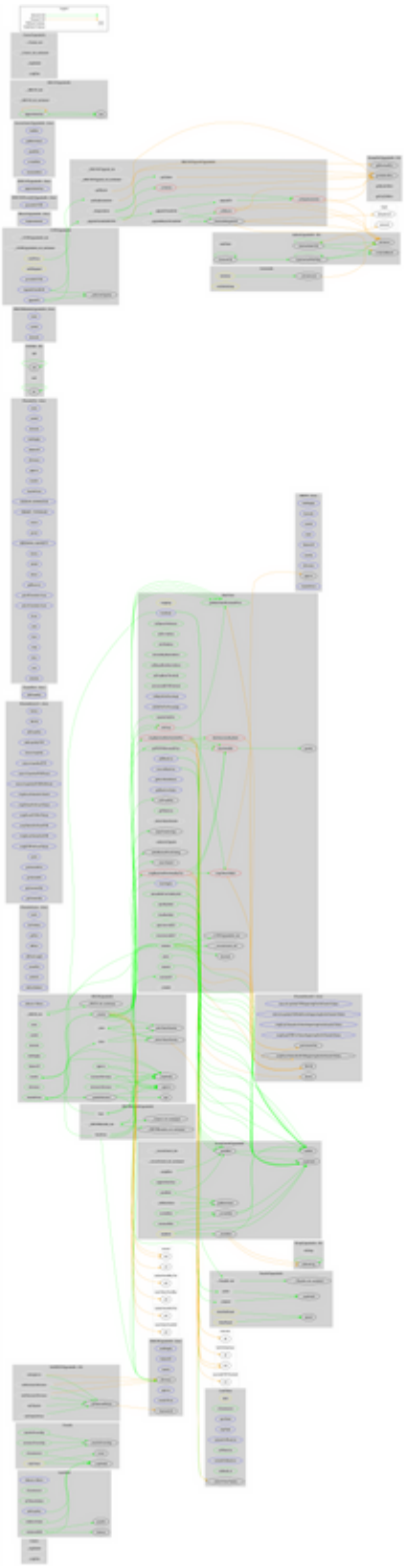
	_setupRole	Internal	✓	
	_setRoleAdmin	Internal	✓	
	_grantRole	Internal	✓	
	_revokeRole	Internal	✓	
PausableUpgradeable	Implementation	Initializable, ContextUpgradeable		
	__Pausable_init	Internal	✓	onlyInitializing
	__Pausable_init_unchained	Internal	✓	onlyInitializing
	paused	Public		-
	_pause	Internal	✓	whenNotPaused
	_unpause	Internal	✓	whenPaused
ERC20Upgradeable	Implementation	Initializable, ContextUpgradeable, IERC20Upgradeable, IERC20MetadataUpgradeable		
	<Receive Ether>	External	Payable	-
	__ERC20_init	Internal	✓	onlyInitializing
	__ERC20_init_unchained	Internal	✓	onlyInitializing
	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	✓	
	_spendAllowance	Internal	✓	
	_beforeTokenTransfer	Internal	✓	
	_afterTokenTransfer	Internal	✓	
ERC20BurnableUpgradeable	Implementation	Initializable, ContextUpgradeable, ERC20Upgradeable		
	__ERC20Burnable_init	Internal	✓	onlyInitializing
	__ERC20Burnable_init_unchained	Internal	✓	onlyInitializing
	burn	Internal	✓	
	burnFrom	Internal	✓	
LockToken	Implementation	AccessControlUpgradeable		
	<Constructor>	Public	✓	-
	openTrade	External	✓	onlyRole
	stopTrade	External	✓	onlyRole
	includeToWhiteList	External	✓	onlyRole
	setWhiteList	Public	✓	onlyRole
	includeToBlackList	External	✓	onlyRole
	setBlackList	Public	✓	onlyRole
MainToken	Implementation	Initializable, ERC20Upgradeable, ERC20BurnableUpgradeable, PausableUpgradeable, AccessControlUpgradeable, UUPSUpgradeable, LockToken		
	initialize	Public	✓	initializer

	initSpecialAddresses	Public	✓	onlyRole whenPaused
	setDevAddress	Public	✓	onlyRole whenPaused
	setLPAddress	Public	✓	onlyRole whenPaused
	setAutoBuyBackAddress	Public	✓	onlyRole whenPaused
	setManualBuyBackAddress	Public	✓	onlyRole whenPaused
	setSwapBackThreshold	Public	✓	onlyRole
	setAutoAddETHThrehold	Public	✓	onlyRole
	setBuyInTaxPercentage	External	✓	onlyRole whenPaused
	setSellOutTaxPercentage	External	✓	onlyRole whenPaused
	addBlackList	External	✓	-
	removeBlackList	External	✓	-
	pause	Public	✓	onlyRole
	unpause	Public	✓	onlyRole
	burnDead	External	✓	-
	burnSupply	External	✓	-
	getDevWalletStatus	External		-
	getBlackListStatus	External		-
	getMainTokenEstimatedPrice	Public		-
	getWETHEstimatedPrice	Public		-
	getWhiteList	Public		-
	_beforeTokenTransfer	Internal	✓	whenNotPaused
	transferOnlyDev	Public	✓	-
	_authorizeUpgrade	Internal	✓	onlyRole
	checkTransferType	Private		
	isNotBlockedFromTrading	Private		
	distributeAutoBuyBack	Private	✓	
	swapTokenToBnb	Private	✓	
	distributeBnb	Private	✓	
	swapBackAndDistributeBuyTax	Private	✓	swapping
	swapBackAndDistributeSellTax	Private	✓	swapping

	_basicTransfer	Internal	✓	
	canSwap	Private		
	setmaxBnbUserCanBuySell	Public	✓	whenPaused
	openBuyBack	Public	✓	whenPaused
	closeBuyBack	Public	✓	whenPaused
	canSwapBack	Public		-
	openAutoAddLP	Public	✓	whenPaused
	closeAutoAddLP	Public	✓	whenPaused
	canAutoLP	Public		-
	_transfer	Internal	✓	open whenNotPaused

Contract Flow



Domain Info

Domain Name	boainu.com
Registry Domain ID	2689646246_DOMAIN_COM-VRSN
Creation Date	2022-04-16T10:26:49Z
Updated Date	2022-04-16T10:26:52Z
Registry Expiry Date	2023-04-16T10:26:49Z
Registrar WHOIS Server	whois.google.com
Registrar URL	https://domains.google.com
Registrar	Google LLC
Registrar IANA ID	895

The domain has been created 11 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 manipulating fees, stopping transactions and blacklisting addresses. The contract can be converted into a honeypot and prevent 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