Cyberscope

Audit Report

Hubin Network

August 2022

Type	BEP20
Network	BSC
Address	0xe33012187aF219072DfF81f54060fEBEd2A91337
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	HUBINNETWORK
Compiler Version	v0.8.8+commit.dddeac2f
Optimization	200 runs
Licence	Unlicense
Explorer	https://bscscan.com/token/0xe33012187af219072dff81f54060febed2a91337
Symbol	HBN
Decimals	18
Total Supply	100,000,000

Source Files

Filename	**SHA256**
contract.sol	6ecf0334f07acab0edbc376c9a2d71522d5a51eb7d60bcf871bca6c28a3f9fbb

Audit Updates

Initial Audit	29th August 2022
Corrected	

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#L759
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 `rescueBNB` method.

```
function rescueBNB(uint256 weiAmount) external onlyOwner {
        payable(owner()).transfer(weiAmount);
    }
```

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
●	STC	Succeeded Transfer Check	Unresolved
●	FSA	Fixed Swap Address	Unresolved
●	CO	Code Optimization	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
●	L13	Divide before Multiply Operation	Unresolved
●	L14	Uninitialized Variables in Local Scope	Unresolved

STC - Succeeded Transfer Check

Criticality	minor / informative
Location	contract.sol#L763
Status	Unresolved

Description

According to the ERC20 specification, the transfer methods should be checked if the result is successful. Otherwise, the contract may wrongly assume that the transfer has been established.

```
function rescueBSC20(address tokenAdd, uint256 amount) external onlyOwner {
        require(tokenAdd != address(this), "Owner can't claim contract's balance of its own
tokens");
        IBEP20(tokenAdd).transfer(owner(), amount);
    }
```

Recommendation

The contract should check if the result of the transfer methods is successful.

FSA - Fixed Swap Address

Criticality	minor / informative
Location	contract.sol#L445
Status	Unresolved

Description

The swap address is assigned once in the constructor and it can not be changed. The decentralized swaps sometimes create a new swap version or abandon the current. A contract that cannot change the swap address may not be able to catch-up the upgrade.

```
constructor() BEP20("HUBIN NETWORK", "HBN") {
        _tokengeneration(msg.sender, 1e8 * 10**decimals());
        exemptFee[msg.sender] = true;

        IRouter _router = IRouter(0x10ED43C718714eb63d5aA57B78B54704E256024E);
        // Create a pancake pair for this new token
        address _pair = IFactory(_router.factory()).createPair(address(this), _router.WETH());
```

Recommendation

It could be better to allow the swap address mutation in case of future swap updates.

CO - Code Optimization

Criticality	minor / informative
Location	contract.sol#L554
Status	Unresolved

Description

1. The contract is repeating a predefined calculation in every transaction.
2. The contract is repeating the same value in two variables.

```
else if (recipient == pair && !useLaunchFee) {
    feeswap =
        sellTaxes.liquidity +
        sellTaxes.marketing +
        sellTaxes.ops +
        sellTaxes.dev;
    feesum = feeswap;
    currentTaxes = sellTaxes;
} else if (!useLaunchFee) {
    feeswap =
        taxes.liquidity +
        taxes.marketing +
        taxes.ops +
        taxes.dev ;
    feesum = feeswap;
    currentTaxes = taxes;
} else if (useLaunchFee) {
    feeswap = launchtax;
    feesum = launchtax;
}
```

Recommendation

1. The sell and the buy taxes sum could be calculated once in the setter segments of sellTaxes and taxes.
2. Since the `feesum` is always the `feeswap` value, it is redundant to use both variables.

L01 - Public Function could be Declared External

Criticality	minor / informative
Location	contract.sol#L90,207,336,230,181,82,340,133,146,163,114
Status	Unresolved

Description

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

```
symbol
increaseAllowance
renounceOwnership
decreaseAllowance
transferFrom
name
transferOwnership
transfer
allowance
...
```

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#L403
Status	Unresolved

Description

Constant state variables should be declared constant to save gas.

```
launchtax
```

Recommendation

Add the constant attribute to state variables that never change.

L03 - Redundant Statements

Criticality	minor / informative
Location	contract.sol#L5
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#L683,693,692,686,694,682,709,401,408,702,359,696,56,685,589,684,58,695,736,676
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.

```
_marketing
SetSellTaxes
_dev
_liquidity
SetBuyTaxes
_deadline
genesis_block
deadWallet
EnableTrading
...
```

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#L746,676,730,709
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.

```
maxBuyLimit = maxBuy * 10 ** decimals()
tokenLiquidityThreshold = new_amount * 10 ** decimals()
coolDownTime = time * 1
deadline = _deadline
```

Recommendation

Emit an event for critical parameter changes.

L13 - Divide before Multiply Operation

Criticality	minor / informative
Location	contract.sol#L589
Status	Unresolved

Description

Performing divisions before multiplications may cause lose of prediction.

```
unitBalance = deltaBalance / (denominator - swapTaxes.liquidity)
```

Recommendation

The multiplications should be prior to the divisions.

L14 - Uninitialized Variables in Local Scope

Criticality	minor / informative
Location	contract.sol#L537,539,536
Status	Unresolved

Description

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

```
feesum
currentTaxes
feeswap
```

Recommendation

All the local scoped variables should be initialized.

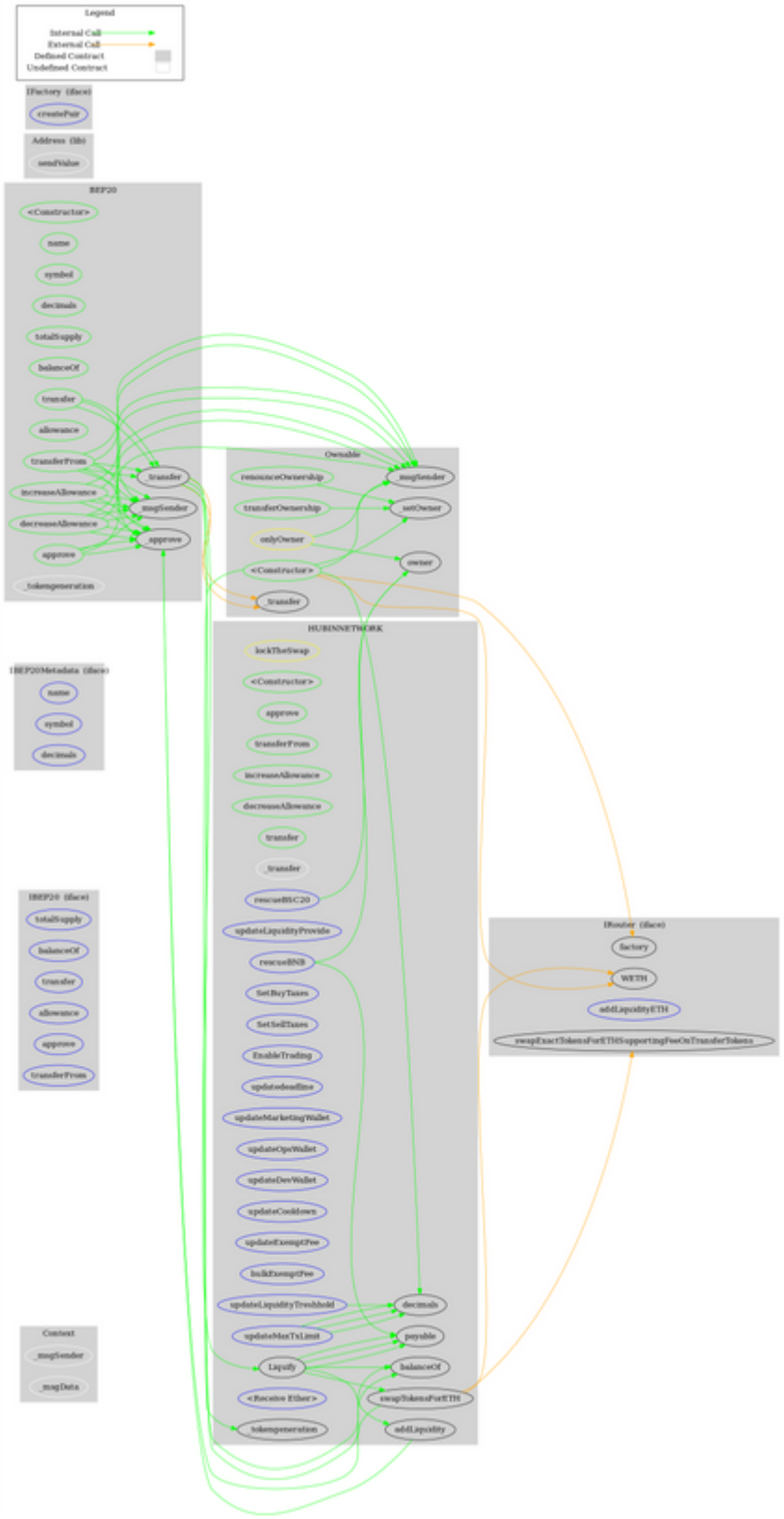
Contract Functions

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
IBEP20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
IBEP20Metadata	Interface	IBEP20		
	name	External		-
	symbol	External		-
	decimals	External		-
BEP20	Implementation	Context, IBEP20, IBEP20Metadata		
	<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	✓	
	_tokengeneration	Internal	✓	
	_approve	Internal	✓	
Address	Library			
	sendValue	Internal	✓	
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_setOwner	Private	✓	
IFactory	Interface			
	createPair	External	✓	-
IRouter	Interface			
	factory	External		-
	WETH	External		-
	addLiquidityETH	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
HUBINNETWORK	Implementation	BEP20, Ownable		
	<Constructor>	Public	✓	BEP20
	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-

	transfer	Public	✓	-
	_transfer	Internal	✓	
	Liquify	Private	✓	lockTheSwap
	swapTokensForETH	Private	✓	
	addLiquidity	Private	✓	
	updateLiquidityProvide	External	✓	onlyOwner
	updateLiquidityTreshhold	External	✓	onlyOwner
	SetBuyTaxes	External	✓	onlyOwner
	SetSellTaxes	External	✓	onlyOwner
	EnableTrading	External	✓	onlyOwner
	updatedeadline	External	✓	onlyOwner
	updateMarketingWallet	External	✓	onlyOwner
	updateOpsWallet	External	✓	onlyOwner
	updateDevWallet	External	✓	onlyOwner
	updateCooldown	External	✓	onlyOwner
	updateExemptFee	External	✓	onlyOwner
	bulkExemptFee	External	✓	onlyOwner
	updateMaxTxLimit	External	✓	onlyOwner
	rescueBNB	External	✓	onlyOwner
	rescueBSC20	External	✓	onlyOwner
	<Receive Ether>	External	Payable	-

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.

The contract will increase the transaction fees to 99 for a max of 5 blocks. There is also a limit of max 14% on sell fees and 10% buy fees.

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.

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