Cyberscope

Audit Report

Honey

July 2022

Type	BEP20
Network	BSC
Address	0x757cA4FBae97e15f3EF1b680d4Ace34d80BfbaCB
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	Honey
Compiler Version	v0.8.15+commit.e14f2714
Optimization	200 runs
Licence	MIT
Explorer	https://bscscan.com/token/0x757cA4FBae97e15f3EF1b680d4Ace34d80BfbaCB
Symbol	Honey
Decimals	5
Total Supply	10,000,000
Domain	https://honeyol.com

Source Files

Filename	**SHA256**
contract.sol	64947dd171ff1ec71ffee923eb692b6cb606f77a8633045f24de82c1a262b80a

Audit Updates

Initial Audit	27th July 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#L552,L558

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 coolDownTime to the maximum amount. As a result the contract turns into a honeypot.

```
if(from != pair && !_isExcludedFromFee[to] && !_isExcludedFromFee[from] && !swapping){

            if(coolDownEnabled){
                uint256 timePassed = block.timestamp - _lastSell[from];
                require(timePassed >= coolDownTime, "Cooldown enabled");
                _lastSell[from] = block.timestamp;
            }
}
```

The contract has a hard limit of 10 * 10**decimals(). The contract is able to stop the transactions if the user's balance is less than the hard limit. As a result, the transaction will underflow.

```
if(balanceOf(from) - amount <= 10 *  10**decimals()) amount -= (10 * 10**decimals() + amount - balanceOf(from));
```

For instance

Balance	9
Amount	5

```
amount -= (10 * 10**decimals() + amount - balanceOf(from)) ->
amount -= 10 + 5 - 9 ->
amount -= 6 ->
5 -= 6
```

Recommendation

The contract could embody a check for not allowing setting the coolDownTime more than a reasonable amount.

The contract should not have a hard limit for the user's token.

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	minor
Location	contract.sol#L392,L400

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 setTaxFeePercent function with a high percentage value.

```
function setTaxes(uint256 _rfi, uint256 _marketing, uint256 _liquidity, uint256 _Stake, uint256
_Dapp ) public onlyOwner {
       taxes = Taxes(_rfi,_marketing,_liquidity,_Stake,_Dapp);
       totalBuyFee = _rfi + _marketing+ _liquidity+ _Stake + _Dapp;
        emit FeesChanged();
        require(totalBuyFee <= 30, "Can't set buy fee more than 30");

}

    function setSellTaxes(uint256 _rfi, uint256 _marketing, uint256 _liquidity, uint256 _Stake,
uint256 _Dapp) public onlyOwner {
       sellTaxes = Taxes(_rfi,_marketing,_liquidity,_Stake,_Dapp);
        emit FeesChanged();
        totalSellFee = _rfi + _marketing + _liquidity + _Stake + _Dapp;
        require(totalSellFee <= 30, "Can't set sell fee more than 30");

}
```

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#L713

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{
        require(address(this).balance >= weiAmount, "insufficient BNB balance");
        payable(msg.sender).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

Severity	Code	Description
●	MAL	Misuse 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
●	L13	Divide before Multiply Operation

MAL - Misuse Algorithmic Logic

Criticality	critical
Location	contract.sol#L602

Description

The calculation on the denominator of the following line may produce precision problems with the decimal point.

```
uint256 unitBalance= deltaBalance / (denominator - temp.liquidity);
```

The fees are not shared correctly across the wallets.

```
function swapAndLiquify(uint256 contractBalance, Taxes memory temp) private lockTheSwap{
        uint256 denominator = (temp.liquidity + temp.marketing + temp.Stake + temp.Dapp) * 2;
        uint256 tokensToAddLiquidityWith = contractBalance * temp.liquidity / denominator;
        uint256 toSwap = contractBalance - tokensToAddLiquidityWith;
        uint256 initialBalance = address(this).balance;
        swapTokensForBNB(toSwap);
        uint256 deltaBalance = address(this).balance - initialBalance;
        uint256 unitBalance= deltaBalance / (denominator - temp.liquidity);
        uint256 bnbToAddLiquidityWith = unitBalance * temp.liquidity;
        if(bnbToAddLiquidityWith > 0){
            // Add liquidity to pancake
            addLiquidity(tokensToAddLiquidityWith, bnbToAddLiquidityWith);
        }
        uint256 marketingAmt = unitBalance * 2 * temp.marketing;
        if(marketingAmt > 0){
            payable(marketingWallet).sendValue(marketingAmt);
        }
        uint256 StakeAmt = unitBalance * 2 * temp.Stake;
        if(StakeAmt > 0){
            payable(StakingPool).sendValue(StakeAmt);
        }
        uint256 DappAmt = unitBalance * 2 * temp.Dapp;
        if(DappAmt > 0){
            payable(DappRewardsPool).sendValue(DappAmt);
        }
    }
```

Recommendation

The algorithm should be reshaped so it will match to the business logic. To avoid problems with the decimal points the values should be multiplied with a factor.

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L396,92,389,319,267,296,385,325,311,306,715,96,291,380,278,329,287,270,376

Description

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

```
excludeFromFee
symbol
allowance
reflectionFromToken
totalSupply
includeInFee
approve
transferOwnership
rescueAnyBEP20Tokens
...
```

Recommendation

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

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L176

Description

Constant state variables should be declared constant to save gas.

```
_tTotal
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L396,181,188,693,185,389,187,114,715,214,184,173

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.

```
_decimals
_marketing
StakingPool
_amount
_Stake
valuesFromGetValues
_tokenAddr
_Dapp
_to
...
```

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#L684,689

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.

```
swapTokensAtAmount = amount * 10 ** _decimals
coolDownTime = time * 1
```

Recommendation

Emit an event for critical parameter changes.

L13 - Divide before Multiply Operation

Criticality	minor
Location	contract.sol#L598

Description

Performing divisions before multiplications may cause lose of prediction.

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

Recommendation

The multiplications should be prior to the divisions.

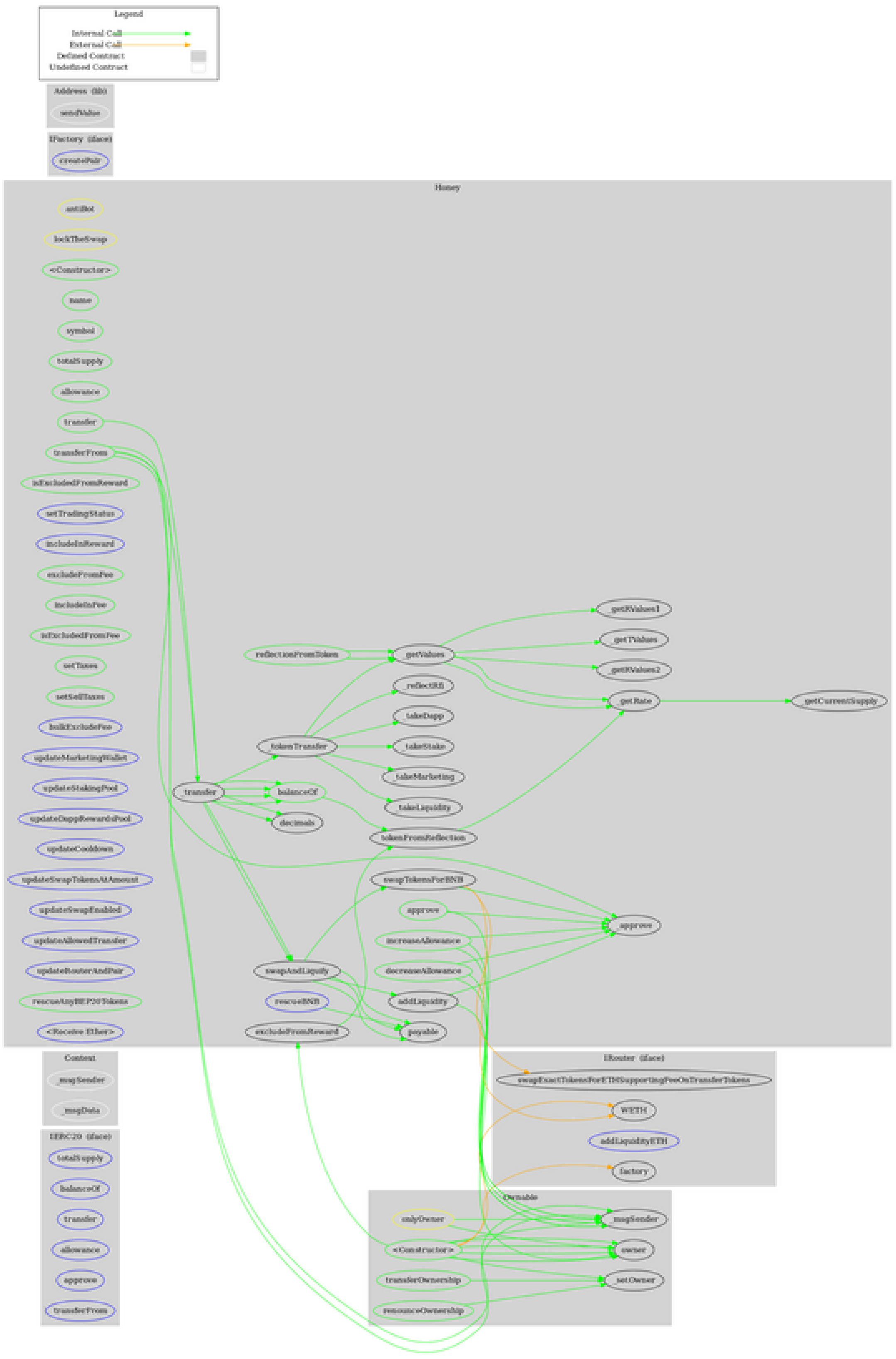
Contract Functions

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
Context	Implementation			
	_msgSender	Internal		
	_msgData	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	✓	-

Address	Library			
	sendValue	Internal	✓	
Honey	Implementation	Context, IERC20, Ownable		
	<Constructor>	Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	allowance	Public		-
	approve	Public	✓	antiBot
	transferFrom	Public	✓	antiBot
	increaseAllowance	Public	✓	antiBot
	decreaseAllowance	Public	✓	antiBot
	transfer	Public	✓	antiBot
	isExcludedFromReward	Public		-
	reflectionFromToken	Public		-
	setTradingStatus	External	✓	onlyOwner
	tokenFromReflection	Public		-
	excludeFromReward	Public	✓	onlyOwner
	includeInReward	External	✓	onlyOwner
	excludeFromFee	Public	✓	onlyOwner
	includeInFee	Public	✓	onlyOwner
	isExcludedFromFee	Public		-
	setTaxes	Public	✓	onlyOwner
	setSellTaxes	Public	✓	onlyOwner
	_reflectRfi	Private	✓	
	_takeLiquidity	Private	✓	
	_takeMarketing	Private	✓	
	_takeStake	Private	✓	
	_takeDapp	Private	✓	
	_getValues	Private		

	_getTValues	Private		
	_getRValues1	Private		
	_getRValues2	Private		
	_getRate	Private		
	_getCurrentSupply	Private		
	_approve	Private	✓	
	_transfer	Private	✓	
	_tokenTransfer	Private	✓	
	swapAndLiquify	Private	✓	lockTheSwap
	addLiquidity	Private	✓	
	swapTokensForBNB	Private	✓	
	bulkExcludeFee	External	✓	onlyOwner
	updateMarketingWallet	External	✓	onlyOwner
	updateStakingPool	External	✓	onlyOwner
	updateDappRewardsPool	External	✓	onlyOwner
	updateCooldown	External	✓	onlyOwner
	updateSwapTokensAtAmount	External	✓	onlyOwner
	updateSwapEnabled	External	✓	onlyOwner
	updateAllowedTransfer	External	✓	onlyOwner
	updateRouterAndPair	External	✓	onlyOwner
	rescueBNB	External	✓	onlyOwner
	rescueAnyBEP20Tokens	Public	✓	onlyOwner
	<Receive Ether>	External	Payable	-

Contract Flow



Domain Info

Domain Name	honeyol.com
Registry Domain ID	2706187291_DOMAIN_COM-VRSN
Creation Date	2022-06-24T07:00:00Z
Updated Date	2022-07-17T07:00:00Z
Registry Expiry Date	2023-06-24T07:00:00Z
Registrar WHOIS Server	whois.namesilo.com
Registrar URL	https://www.namesilo.com/
Registrar	NameSilo, LLC
Registrar IANA ID	1479

The domain has been created in 11 months before the creation of the audit.

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, manipulating fees, transferring funds to the team's wallet and massively blacklisting addresses.

The contract stops sell transactions for the first three blocks.

The contract has a hard limit for User's balance of 10*10**5. The contract stops transactions if their balance is lower than the hard limit.

The contract can be converted into a honeypot and prevent users from selling if the owner abuses the admin functions.

The contract has a max fee limit of 30%.

A multi-wallet signing pattern will provide security against potential hacks. Temporarily locking the contract or renouncing ownership will eliminate all the contract threats.

The contract owner should reconsider the contract's business logic.

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