Cyberscope

Audit Report

Vacuum

November 2022

SHA256 0ee2a1c6bc4655dc463cc159ba239e894de6193648cc9e13da7dbb2438277a81

Audited by © cyberscope

Table of Contents

Contract Review

Contract Name	Vacuum
Compiler Version	v0.8.9+commit.e5eed63a
Optimization	0 runs
Explorer	https://testnet.bscscan.com/token/0x36D330319a775cfc522c630e0126E8bbdFdC6829
Symbol	VC
Decimals	9
Total Supply	2,997,924,580
Domain	https://testnet.bscscan.com/address/0x36D330319a775cfc522c630e0126E8bbdFdC6829

Audit Updates

Initial Audit	7th November 2022
Corrected	

Source Files

Filename	SHA256
@openzeppelin/contracts/access/Ownable.sol	9353af89436556f7ba8abb3f37a6677249aa4df6024fbfaa94f79ab2f44f3231
@openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol	af5c8a77965cc82c33b7ff844deb9826166689e55dc037a7f2f790d057811990
@openzeppelin/contracts/token/ERC20/IERC20.sol	94f23e4af51a18c2269b355b8c7cf4db8003d075c9c541019eb8dcf4122864d5
@openzeppelin/contracts/utils/Address.sol	1e0922f6c0bf6b1b8b4d480dcabb691b1359195a297bde6dc5172e79f3a1f826
@openzeppelin/contracts/utils/Context.sol	1458c260d010a08e4c20a4a517882259a23a4baa0b5bd9add9fb6d6a1549814a
@openzeppelin/contracts/utils/math/SafeMath.sol	0dc33698a1661b22981abad8e5c6f5ebca0dfe5ec14916369a2935d888ff257a
contracts/vaccum(final).sol	0ee2a1c6bc4655dc463cc159ba239e894de6193648cc9e13da7dbb2438277a81

Contract Analysis

🔴 Critical 🟠 Medium ⚪ Minor / Informative 🔵 Pass

Severity	Code	Description	Status
🟠	ST	Stops Transactions	Unresolved
🔵	OCTD	Transfers Contract's Tokens	Passed
🔵	OTUT	Transfers User's Tokens	Passed
🔴	ELFM	Exceeds Fees Limit	Unresolved
🔵	ULTW	Transfers Liquidity to Team Wallet	Passed
🔵	MT	Mints Tokens	Passed
🔵	BT	Burns Tokens	Passed
🔴	BC	Blacklists Addresses	Unresolved

ST - Stops Transactions

Criticality	medium
Location	contract.sol#L821,L827
Status	Unresolved

Description

The contract owner has the authority to stop the transactions for all users excluding the owner. The owner may take advantage of it by setting the isLocked to true.

```
if (isLocked)
    require(
        _lockStart + _lockTime < block.timestamp,
       "transfer locked"
  );
```

The contract owner has the authority to stop the transactions for all users excluding the owner. The owner may take advantage of it by setting the _antiWhale to true and _whaleLimit to zero.

```
if (_antiWhale)
    require(
        amount <= _whaleLimit,
        "Transfer amount exceeds the whaleLimit."
    );
```

Recommendation

The contract could embody a check for not allowing setting the _whaleLimit 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 - Exceeds Fees Limit

Criticality	critical
Location	contract.sol#L634 - L647
Status	Unresolved

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 setMarketingFee, setLiquidityFee and setReflectionFee functions with a high percentage value.

```
function setMarketingFee(uint256 marketingFee) external onlyOwner {
    require(marketingFee < 100, "Fee cannot over 100");
    _marketingFee = marketingFee * 10**2;
}

function setLiquidityFee(uint256 liqSwapFee) external onlyOwner {
    require(liqSwapFee < 100, "Fee cannot over 100");
    _liqSwapFee = liqSwapFee * 10**2;
}

function setReflectionFee(uint256 refFee) external onlyOwner {
    require(refFee < 100, "Fee cannot over 100");
    _taxFee = refFee * 10**2;
}
```

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 - Blacklists Addresses

Criticality	critical
Location	contract.sol#L815,839
Status	Unresolved

Description

The contract owner has the authority to stop addresses from transactions. The owner may take advantage of it by calling the addToBlacklist function. The contract owner can also add the senders to the blacklist by setting the deadBlocks to zero.

```
require(
    !_isInBlacklist[from] && !_isInBlacklist[to],
    "you are in a blacklist"
);
//
if (
    tradingActiveBlock > 0 &&
    tradingActiveBlock + deadBlocks > block.number
) {
    addToBlacklist(to);
}
```

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 / Informative

Severity	Code	Description	Status
🔴	ZD	Zero Division	Unresolved
⚪	L02	State Variables could be Declared Constant	Unresolved
⚪	L04	Conformance to Solidity Naming Conventions	Unresolved
⚪	L07	Missing Events Arithmetic	Unresolved
⚪	L09	Dead Code Elimination	Unresolved
⚪	L13	Divide before Multiply Operation	Unresolved

ZD - Zero Division

Criticality	critical
Location	contracts/vaccum(final).sol#L868,871
Status	Unresolved

Description

The contract is using variables that may be set to zero as denominators. As a result, the transactions will revert.

```
uint256 liqAmount = contractTokenBalance.mul(_liqSwapFee).div(
    _liqSwapFee + _marketingFee
);
uint256 marketingAmount = contractTokenBalance.mul(_marketingFee).div(
    _liqSwapFee + _marketingFee
);
```

Recommendation

The contract should prevent those variables to be set to zero or should not allow to execute the corresponding statements.

L02 - State Variables could be Declared Constant

Criticality	minor / informative
Location	contracts/vaccum(final).sol#L411,366,371,372,373
Status	Unresolved

Description

Constant state variables should be declared constant to save gas.

```
numTokensSellToAddToLiquidity
_tTotal
_name
_symbol
_decimals
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor / informative
Location	contracts/vaccum(final).sol#L375,1101,382,381,404,768,72,103,378,70,385,397,384,1094,654,764,147
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.

```
_taxFee
_marketingAddress
_marketingFee
_liqSwapFee
_antiWhale
_amount
PERMIT_TYPEHASH
MINIMUM_LIQUIDITY
_liquidityFee
...
```

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	contracts/vaccum(final).sol#L1039,634,1094,649,1081,644,1044,639
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.

```
_feeBuy = feeBuy
_marketingFee = marketingFee * 10 ** 2
deadBlocks = _deadBlocks
_whaleLimit = whaleLimit * 10 ** 9
_lockTime = lockTime
_taxFee = refFee * 10 ** 2
_feeSell = feeSell
_liqSwapFee = liqSwapFee * 10 ** 2
```

Recommendation

Emit an event for critical parameter changes.

L13 - Divide before Multiply Operation

Criticality	minor / informative
Location	contracts/vaccum(final).sol#L928
Status	Unresolved

Description

Performing divisions before multiplications may cause lose of prediction.

```
_taxFee = _taxFee.mul(_feeBuy).div(100)
```

Recommendation

The multiplications should be prior to the divisions.

Contract Functions

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	_checkOwner	Internal		
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_transferOwnership	Internal	✓	
IERC20Metadata	Interface	IERC20		
	name	External		-
	symbol	External		-
	decimals	External		-
IERC20	Interface			
	totalSupply	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		
Context	Implementation			
	_msgSender	Internal		
	_msgData	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		
IUniswapV2Factory	Interface			
	feeTo	External		-
	feeToSetter	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	createPair	External	✓	-
	setFeeTo	External	✓	-
	setFeeToSetter	External	✓	-

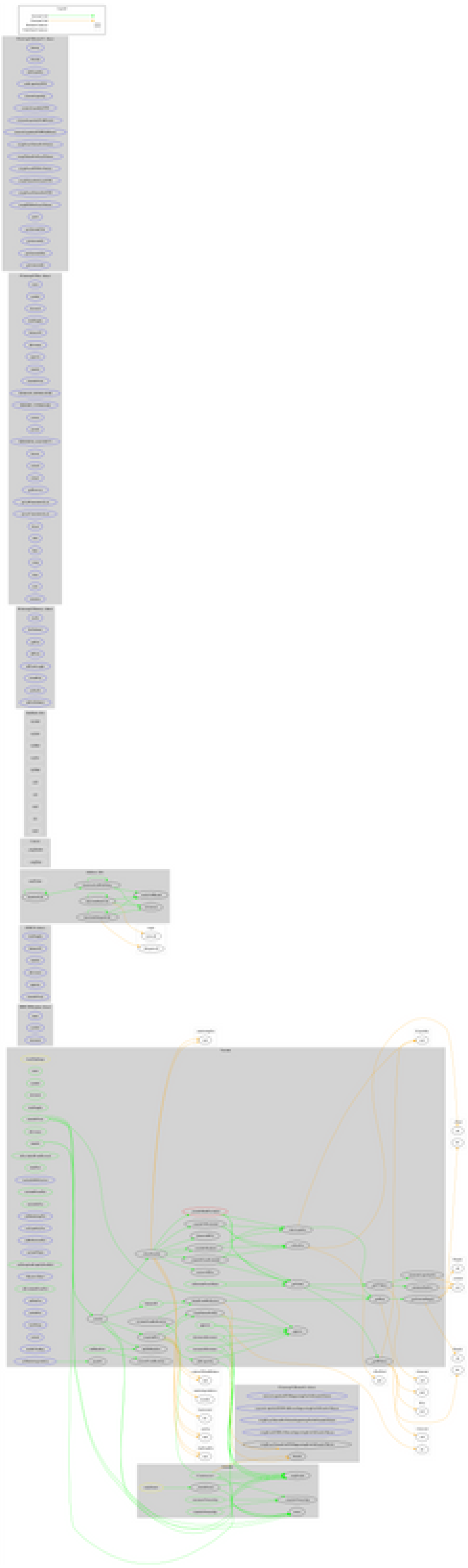
IUniswapV2Pair	Interface			
	name	External		-
	symbol	External		-
	decimals	External		-
	totalSupply	External		-
	balanceOf	External		-
	allowance	External		-
	approve	External	✓	-
	transfer	External	✓	-
	transferFrom	External	✓	-
	DOMAIN_SEPARATOR	External		-
	PERMIT_TYPEHASH	External		-
	nonces	External		-
	permit	External	✓	-
	MINIMUM_LIQUIDITY	External		-
	factory	External		-
	token0	External		-
	token1	External		-
	getReserves	External		-
	price0CumulativeLast	External		-
	price1CumulativeLast	External		-
	kLast	External		-
	mint	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	✓	-
Vacuum	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		-
	reflectionFromToken	Public		-
	tokenFromReflection	Public		-
	excludeFromReflection	Public	✓	onlyOwner
	includeInReflection	External	✓	onlyOwner
	_transferBothExcluded	Private	✓	
	excludeFromFee	Public	✓	onlyOwner
	includeInFee	Public	✓	onlyOwner
	setMarketingFee	External	✓	onlyOwner
	setLiquidityFee	External	✓	onlyOwner
	setReflectionFee	External	✓	onlyOwner
	setAntiWhale	External	✓	onlyOwner
	setSwapAndLiquifyEnabled	Public	✓	onlyOwner
	<Receive Ether>	External	Payable	-
	_reflectFee	Private	✓	
	_getValues	Private		
	_getTValues	Private		
	_getRValues	Private		
	_getRate	Private		
	_getCurrentSupply	Private		
	_takeLiquidity	Private	✓	
	calculateTaxFee	Private		
	calculateLiquidityFee	Private		
	removeAllFee	Private	✓	
	restoreAllFee	Private	✓	
	isExcludedFromFee	Public		-
	_approve	Private	✓	
	_transfer	Private	✓	

	swapAndFee	Private	✓	lockTheSwap
	swapTokensForEth	Private	✓	
	addLiquidity	Private	✓	
	_tokenTransfer	Private	✓	
	_transferStandard	Private	✓	
	_transferToExcluded	Private	✓	
	_transferFromExcluded	Private	✓	
	setBuyFee	External	✓	onlyOwner
	setSellFee	External	✓	onlyOwner
	setBlacklists	Public	✓	onlyOwner
	addToBlacklist	Public	✓	onlyOwner
	removeFromBlacklist	Public	✓	onlyOwner
	lockToken	External	✓	onlyOwner
	unlock	External	✓	onlyOwner
	enableTrading	External	✓	onlyOwner
	setMarketingAddress	External	✓	onlyOwner

Contract Flow



Domain Info

Domain Name	vacuum.ltd
Registry Domain ID	2515712477_DOMAIN_COM-VRSN
Creation Date	2020-04-17T15:07:05.00Z
Updated Date	2022-03-18T06:00:20.01Z
Registry Expiry Date	2023-04-17T15:07:05.00Z
Registrar WHOIS Server	whois.namecheap.com
Registrar URL	http://www.namecheap.com
Registrar	NAMECHEAP INC
Registrar IANA ID	1068

The domain was created over 2 years before the creation of the audit. It will expire in 5 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, manipulating fees and massively blacklisting addresses. 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