Cyberscope

Audit Report

Vivus Token

April 2022

Type	BEP20
Network	BSC
Address	0x365813964f8b2c7CA3Fc3B6D6f01F82EB689210C
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	VIV
Compiler Version	v0.8.10+commit.fc410830
Optimization	200 runs
Licence	MIT
Explorer	https://testnet.bscscan.com/token/0x365813964f8b2c7CA3Fc3B6D6f01F82EB689210C
Symbol	VIV
Decimals	18
Total Supply	1,000,000,000
Domain	vivustoken.com

Source Files

Filename	**SHA256**
contract.sol	2b17b4f7f20feb3614aace1bf9c8fa891ba183f2c513bca8c098fc7ea0a76a4c

Audit Updates

Initial Audit	13th 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#L753

Description

The contract will only allow the owner to make transactions. The isTradingEnabled boolean is always set to false and there is no function to enable it, hence the contract will not work.

```
        require(isTradingEnabled || (sender == owner() || recipient ==
owner()), "VIV: trading is disabled");
```

Recommendation

Add a function to allow the owner to enable Trading, or remove the require statement

MT - Mint Tokens

Criticality	critical
Location	contract.sol#L833

Description

The contract owner has the authority to mint tokens. The owner may take advantage of it by calling the `mint` function. As a result the contract tokens will be highly inflated

```
function mint(address account, uint256 amount) public onlyOwner {
        _mint(account, amount);
    }
```

Recommendation

The owner should carefully manage the credentials of the owner's account. We advised considering an extra-strong security mechanism that the actions may be quarantined by many users instead of one. The owner could also renounce the contract ownership for a period of time or pass the access to the zero address.

Contract Diagnostics

🔴 Critical 🟠 Medium ● Minor

Severity	Code	Description
●	L01	Public Function could be Declared External
●	L02	State Variables could be Declared Constant
●	L04	Conformance to Solidity Naming Conventions
●	L08	Tautology or Contradiction
●	L09	Dead Code Elimination
●	L13	Divide before Multiply Operation

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L74,83,233,241,258,265,272,279,291,309,326,334,350,369,833

Description

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

```
mint
decreaseAllowance
increaseAllowance
allowance
approve
transferFrom
transfer
getOwner
balanceOf
...
```

Recommendation

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

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L709,705

Description

Constant state variables should be declared constant to save gas.

```
pancakeRouterAddress
isTradingEnabled
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L205,207,528,529,546,568,781,790,799,718

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.

```
maxFees
_newWallet
WETH
MINIMUM_LIQUIDITY
PERMIT_TYPEHASH
DOMAIN_SEPARATOR
_allowances
_balances
...
```

Recommendation

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

L08 - Tautology or Contradiction

Criticality	minor
Location	contract.sol#L808,816

Description

Detects expressions that are tautologies or contradictions. For instance, an uint variable will always be greater than or equal to zero.

```
require(bool,string)(newFee >= 0,VIV: Fee must be greater then or equal 0)
```

Recommendation

Fix the incorrect comparison by changing the value type or the comparison.

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L435,391

Description

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

```
_transfer
_burn
```

Recommendation

Remove unused functions.

L13 - Divide before Multiply Operation

Criticality	minor
Location	contract.sol#L749

Description

Performing divisions before multiplications may cause lose of prediction.

```
fees = (amount / 100)
```

Recommendation

The multiplications should be prior to the divisions.

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
IBEP20	Interface			
	name	External		-
	symbol	External		-
	decimals	External		-
	totalSupply	External		-
	balanceOf	External		-
	getOwner	External		-
	transfer	External	✓	-
	transferFrom	External	✓	-
	approve	External	✓	-
	allowance	External		-
BEP20	Implementation	Ownable, IBEP20		
	<Constructor>	Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-

	balanceOf	Public		-
	getOwner	Public		-
	transfer	Public	✓	-
	transferFrom	Public	✓	-
	approve	Public	✓	-
	allowance	Public		-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	_transfer	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_approve	Internal	✓	
	_setupDecimals	Internal	✓	
	_beforeTokenTransfer	Internal	✓	
IPancakeV2Factory	Interface			
	feeTo	External		-
	feeToSetter	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	createPair	External	✓	-
	setFeeTo	External	✓	-
	setFeeToSetter	External	✓	-
IPancakeV2Pair	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	✓	-
IPancakeV2Router01	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		-
IPancakeV2Router02	Interface	IPancakeV2Router01		
	removeLiquidityETHSupportingFeeOnTransferTokens	External	✓	-
	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External	✓	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
VIV	Implementation	BEP20		
	<Constructor>	Public	Payable	BEP20
	_transfer	Internal	✓	
	<Receive Ether>	External	Payable	-
	updateMarketingWallet	External	✓	onlyOwner
	updateRandomAIWallet	External	✓	onlyOwner
	updateLiquidityWallet	External	✓	onlyOwner
	updateliquidityPoolFee	External	✓	onlyOwner
	updateMarketingFee	External	✓	onlyOwner
	excludeFromFees	Public	✓	onlyOwner
	_setupRandomAIWallet	Private	✓	
	mint	Public	✓	onlyOwner

Contract Flow

Domain Info

Domain Name	vivustoken.com
Registry Domain ID	2663486327_DOMAIN_COM-VRSN
Creation Date	2021-12-23T00:00:00Z
Updated Date	2021-12-23T17:33:26Z
Registry Expiry Date	2022-12-23T17:23:49Z
Registrar WHOIS Server	whois.ascio.com
Registrar URL	http://www.ascio.com
Registrar	Ascio Technologies, Inc
Registrar IANA ID	106

The domain has been created 4 months before the creation of the audit. It will expire in 8 months.

There is no public billing information, the creator is protected by the privacy settings.

Summary

Vivus Token is an interesting project that has a friendly and growing community. There are some functions that can be abused by the owner, like minting new tokens after initial deployment. The maximum fee percentage that can be set is 20% and it only applies on sales. The contract does not allow any transactions to happen apart from the owner. A multi-wallet signing pattern will provide security against potential hacks. Temporarily locking the contract or renouncing ownership will eliminate some of 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