Cyberscope

Audit Report

Nevis Investments

July 2022

Type	BEP20
Network	BSC
Address	0x24DEF7C64B8a8C3Af2aE0a74A69999733502AB98
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	Nevis
Compiler Version	v0.6.12+commit.27d51765
Optimization	200 runs
Licence	None
Explorer	https://bscscan.com/token/0x24DEF7C64B8a8C3Af2aE0a74A69999733502AB98
Symbol	NEVIS
Decimals	9
Total Supply	100,000,000,000
Domain	https://nevis.investments/

Source Files

Filename	**SHA256**
contract.sol	3a36341d9bb0f50fc53472f1b1cfed3ff8a98e74766f3c612e738ae8529e279a

Audit Updates

Initial Audit	12th 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

Contract Diagnostics

Critical Medium Minor

Severity	Code	Description
	MC	Missing Check
	L01	Public Function could be Declared External
	L02	State Variables could be Declared Constant
	L04	Conformance to Solidity Naming Conventions
	L05	Unused State Variable
	L13	Divide before Multiply Operation

MC - Missing Check

Criticality	minor
Location	contract.sol#L786,L800

Description

The contract is processing variables that have not properly sanitized and checked that they form the proper shape. These variables may produce vulnerability issues.The SafeMath library must be used to avoid possible variable overflow. Due to the old solidity version.

```
totalHoldings+=balanceOf(investors[i]);
user.rewards+=userreward;
```

Recommendation

The contract should properly check the variables according to the required specifications.

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L687,637,651,301,684,956,806,676,655,680,310,633,671,316,629,750,625,660,646

Description

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

```
transfer
transferFrom
name
_getCurrentSupply
symbol
geUnlockTime
decreaseAllowance
decimals
transferOwnership
...
```

Recommendation

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

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L572,549,570,567,548,571

Description

Constant state variables should be declared constant to save gas.

```
_symbol
DEAD
_tTotal
_name
ZERO
_decimals
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L547,936,581,369,385,549,548,761,590,368,589,750,687,942,591,407,577,584,578,556,551,574

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.

```
_redistributionFee
_tOwned
_balances
_NevisbankWallet
_NevisRewardWallet
_NevisbankFee
WETH
setfees
NevisswapFee
...
```

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#L566,579,582,575

Description

There are segments that contain unused state variables.

```
_previousredistributionFee
_previousNevisswapsellFee
_previousNevisbankFee
MAX
```

Recommendation

Remove unused state variables.

L13 - Divide before Multiply Operation

Criticality	minor
Location	contract.sol#L775

Description

Performing divisions before multiplications may cause lose of prediction.

```
dividentShare = nevisinbusd.mul(100).div(holdingbusd)
```

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	✓	-
SafeMath	Library			
	add	Internal		
	sub	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
	mod	Internal		
	mod	Internal		
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
Address	Library			
Ownable	Implementation	Context		
	<Constructor>	Internal	✓	
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner

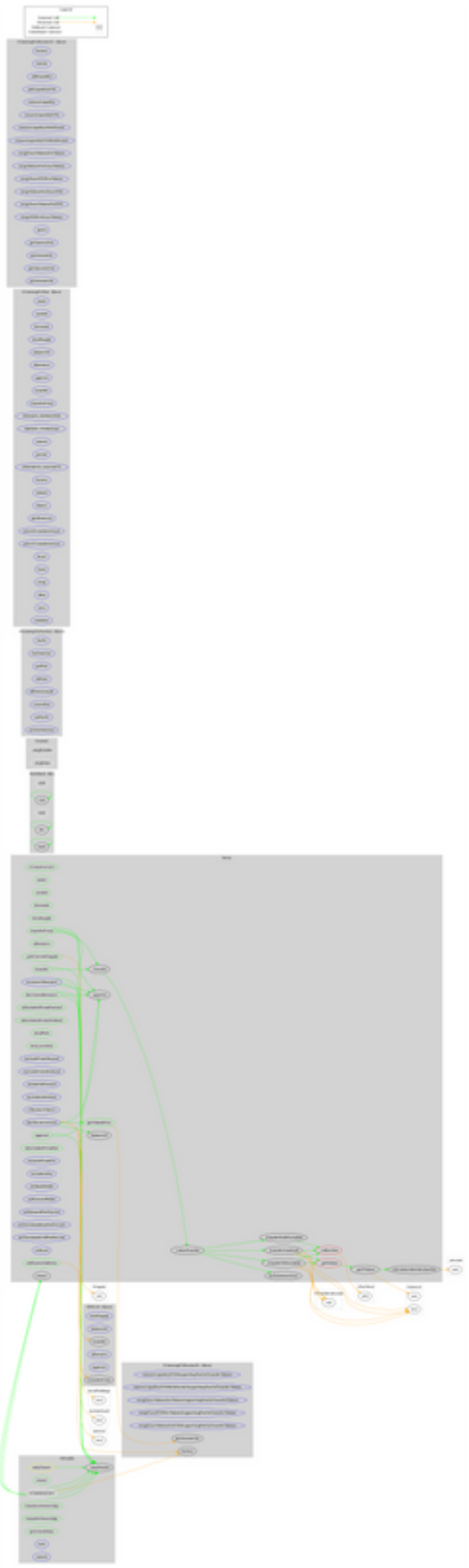
	geUnlockTime	Public		-
	lock	External	✓	onlyOwner
	unlock	External	✓	-
IUniswapV2Factory	Interface			
	feeTo	External		-
	feeToSetter	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	createPair	External	✓	-
	setFeeTo	External	✓	-
	setFeeToSetter	External	✓	-
IUniswapV2Pair	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		-
	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	✓	-
Nevis	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	External	✓	-
	decreaseAllowance	Public	✓	-
	isExcludedFromReward	Public		-
	isExcludedFromDividend	Public		-
	totalFees	Public		-
	total_investor	Public		-
	excludeFromReward	External	✓	onlyOwner
	excludeFromDividend	External	✓	onlyOwner
	includeInReward	External	✓	onlyOwner
	includeInDivident	External	✓	onlyOwner
	<Receive Ether>	External	Payable	-
	_reflectFee	Private	✓	
	_getValues	Private		
	_getTValues	Private		
	_getCurrentSupply	Public		-
	calculateredistributionFee	Private		
	getTokenPrice	Public		-

	distributerewards	External	Payable	onlyOwner
	isExcludedFromFee	Public		-
	_approve	Private	✓	
	_transfer	Private	✓	
	_tokenTransfer	Private	✓	
	includeIninvestor	Private		
	_transferStandard	Private	✓	
	_transferToExcluded	Private	✓	
	_transferBothExcluded	Private	✓	
	excludeFromFee	External	✓	onlyOwner
	includeInFee	External	✓	onlyOwner
	setBankWallet	External	✓	onlyOwner
	setRewardWallet	External	✓	onlyOwner
	setRewardFeePercent	External	✓	onlyOwner
	setNevisbankBuyFeePercent	External	✓	onlyOwner
	setNevisbankSellFeePercent	External	✓	onlyOwner
	setbusd	External	✓	onlyOwner
	setRouterAddress	Public	✓	onlyOwner

Contract Flow



Domain Info

Domain Name	nevis.investments
Registry Domain ID	96638f912b3b43dea5c1a1d6d10aee9e-DONUTS
Creation Date	2022-02-02T10:11:43Z
Updated Date	2022-05-05T11:44:18Z
Registry Expiry Date	2027-02-02T10:11:43Z
Registrar WHOIS Server	www.bigrock.com.in/whois-lookup.php
Registrar URL	http://bigrock.com
Registrar	BigRock Solutions Ltd.
Registrar IANA ID	1495

The domain has been created in over 4 years before the creation of the audit.

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

Summary

Nevis Investments Token is an interesting project that has a friendly and growing community. The Smart Contract analysis reported no compiler error or critical issues. The contract Owner can access some admin functions that can not be used in a malicious way to disturb the users' transactions. There is also a limit of max 10% on buy transactions and 25% on sell 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