Cyberscope

Audit Report

Nevis Investments

June 2022

SHA256 ee5c8791cf9e29a4955e20890a266bdba300105aebc0d4c16d294af03ff90b5a

Audited by © cyberscope

Table of Contents

Contract Review

Contract Name	NEVISTRADEBANK
Testing Deploy	https://testnet.bscscan.com/token/0x1a984838C102D54DaB36e3fcEF8f2AA28F3c5cBD
Domain	https://nevis.investments/

Source Files

Filename	**SHA256**
contract.sol	ee5c8791cf9e29a4955e20890a266bdba300105aebc0d4c16d294af03ff90b5a

Audit Updates

Initial Audit	28th June 2022
Corrected	

Contract Analysis

The users have the ability to invest and withdraw tokens from the Nevis Trade Bank. In addition the users have the ability to become referrers to other users. Nevis Trade Bank has one plan which locks your tokens for 365 days and returns 12% interest.

- On every investment on the contract the owner keeps a 10% fee on the contract. If a referrer is set, 1% bonus goes to the referrer's bonus.
- The users can withdraw tokens when the plan time is passed.
- A referrer can withdraw his bonus if he has dividends in the Nevis Trade Bank.

Contract Diagnostics

Critical Medium Minor

Severity	Code	Description
	STC	Succeeded Transfer Check
	CO	Code Optimization
	DSM	Data Structure Misuse
	L01	Public Function could be Declared External
	L02	State Variables could be Declared Constant
	L04	Conformance to Solidity Naming Conventions
	L05	Unused State Variable
	L09	Dead Code Elimination
	L13	Divide before Multiply Operation
	L14	Uninitialized Variables in Local Scope

STC - Succeeded Transfer Check

Criticality	minor
Location	contract.sol#L446,L579

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.

```
token.transferFrom(address(msg.sender),address(this), _amount);

token.transfer(msg.sender,totalAmount);
```

Recommendation

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

CO - Code Optimization

Criticality	minor
Location	contract.sol#L456

Description

There are code segments that could be optimized. A segment may be optimized so that it becomes a smaller size, consumes less memory, executes more rapidly, or performs fewer operations.

Unnecessary variable check. This code segment can perform fewer operations.

```
if (user.referrer != address(0)) {
    address upline = user.referrer;
    if (upline != address(0)) {
```

The function `buy_token` swaps to the same address, so it can be called with the sum of `PROJECT_FEE` and `TOKEN_FEE` to avoid a function call.

```
uint256 fee = _amount.mul(PROJECT_FEE).div(PERCENTS_DIVIDER);
buy_token(fee);
uint256 purchase_amount= _amount.mul(TOKEN_FEE).div(PERCENTS_DIVIDER);
buy_token(purchase_amount);
```

Recommendation

Rewrite some code segments so the runtime will be more performant.

DSM - Data Structure Misuse

Criticality	minor
Location	contract.sol#L384

Description

The contract uses the valuable _confirmedSnipers as an array. The business logic of the contract does not require to iterate this structure sequentially. Thus, unnecessary loops are produced that increase the required gas.
Since the contract plan is always one, it could be a variable rather than an array.

```
Plan[] internal plans;
```

Recommendation

The contract could use a data structure that provides instant access. For instance, a Set or a Map would fit better to the business logic of the contract. This way the time complexity will be reduced from o(n) to o(1).

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L67,556,645,641,42,78,38,53,73,34,637,46,62,58

Description

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

```
allowance
approve
totalSupply
getContractBUSDBalance
name
increaseAllowance
transfer
symbol
decreaseAllowance
...
```

Recommendation

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

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L31,359,367,30,28,32,371,20,368,372,370

Description

Constant state variables should be declared constant to save gas.

```
NEVIS_TOKEN
Pair
totalRefBonus
busd
BUSD
_decimals
_limitSupply
_name
totalInvested
...
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L425,372,28,31,181,197,370,359,440,180,22,30,495,356,218,371,594,32

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
_Amount
BUSD
WETH
NevisSwapRouter
buy_token
_name
nevis_token
DOMAIN_SEPARATOR
...
```

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#L28,410

Description

There are segments that contain unused state variables.

```
referralAddress
_limitSupply
```

Recommendation

Remove unused state variables.

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L94

Description

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

```
_burn
```

Recommendation

Remove unused functions.

L13 - Divide before Multiply Operation

Criticality	minor
Location	contract.sol#L613

Description

Performing divisions before multiplications may cause lose of prediction.

```
share = user.deposits[i].amount.mul(plans[user.deposits[i].plan].percent).div(PLANPER_DIVIDER)
```

Recommendation

The multiplications should be prior to the divisions.

L14 - Uninitialized Variables in Local Scope

Criticality	minor
Location	contract.sol#L616

Description

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

```
totalAmount
```

Recommendation

All the local scoped variables should be initialized.

Contract Functions

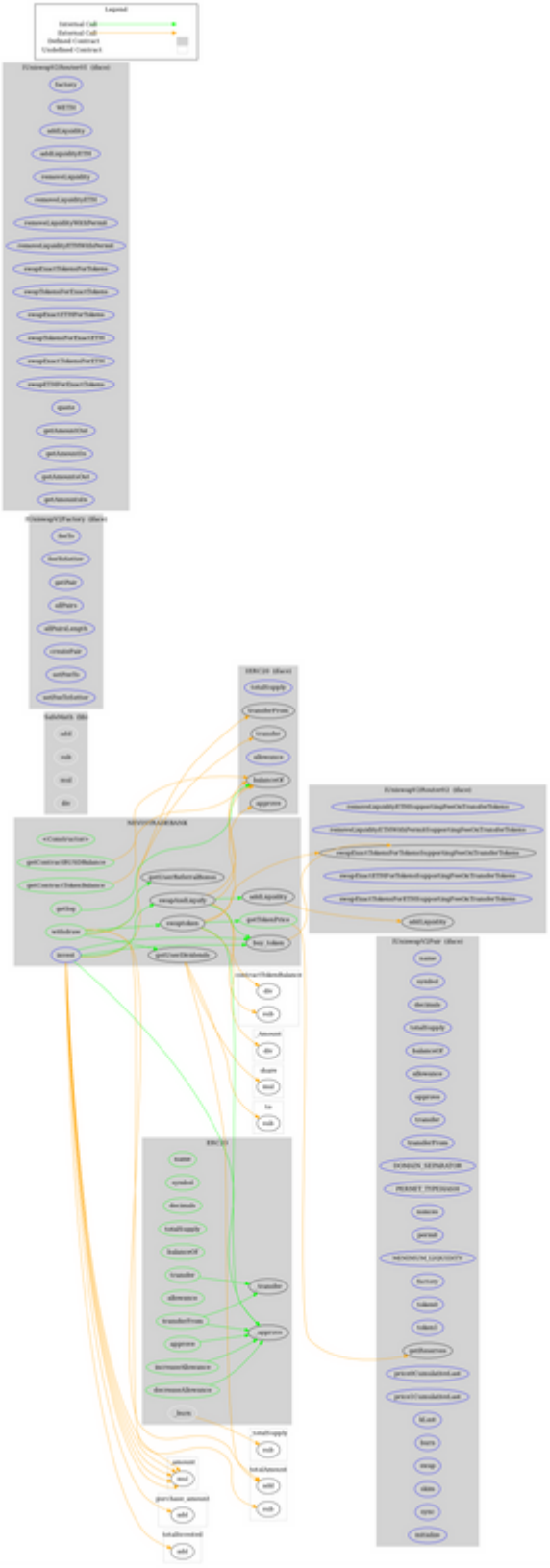
Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
ERC20	Implementation	IERC20		
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	_transfer	Internal	✓	
	_burn	Internal	✓	
	_approve	Internal	✓	
SafeMath	Library			
	add	Internal		
	sub	Internal		
	mul	Internal		

	div	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		-

	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	✓	-
NEVISTRADEBANK	Implementation	ERC20		
	<Constructor>	Public	✓	-
	invest	External	Payable	-
	buy_token	Private	✓	
	swapAndLiquify	Private	✓	
	addLiquidity	Private	✓	
	withdraw	Public	Payable	-
	getTokenPrice	Public		-
	swaptoken	Private	✓	
	getUserDividends	Public		-
	getUserReferralBonus	Public		-
	getContractBUSDBalance	Public		-
	getContractTokenBalance	Public		-
	getlog	Public		-

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 5 months before the creation of the audit. It will expire in over 4 years.

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

Summary

Nevis Investment 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.

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