Cyberscope

Audit Report

Quarashi Vesting

February 2022

Type	BEP20
Network	BSC
Address	0x3C7D0979cB9518F8050D516172FA4144384FC5b0
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	Vesting
Compiler Version	v0.8.7+commit.e28d00a7
Optimization	200 runs
Licence	GNU GPLv3
Explorer	https://bscscan.com/token/0x3C7D0979cB9518F8050D516172FA4144384FC5b0
Source	contract.sol
Domain	quarashi.network

Audit Updates

Initial Audit	25th February 2022
Corrected	

Contract Analysis

Vesting Functionality

The contract implements a vesting functionality. The contract owner schedules a vesting program for an investor.

The vesting schedule contains the amount of money that will be moved to the users and the vest policy.

The contract contains 3 different vest policies.

1. The investor will be able to claim 16.6% of the amount every month. The first claim will be available after 3 months.

2. The investor will be able to claim 16.6% of the amount every month. The first claim will be available after 6 months.

3. The investor will be able to claim 16.6% of the amount every month. The first claim will be available after 12 months.

The investor is able to claim his proportional amount even if the previous month has elapsed.

In this case, the investor should execute the "`claimForUser()`" function as many times as the elapsed months.

Validation Statement Required

Criticality	minor
Location	contract.sol#L742

Description

There is a case when the owner will pass the variable *_lockForUser* with a value higher than the value of *period* (6) and that will result in an exception and the code to fail.

```
uint256 public constant period = 6; // duration of token delivery

processForInvestor[_investor][newIndex] = Inv(
           _lockForUser / period,
           block.timestamp,
           _lockForUser,
           _typeVesting
       );
```

And the following lines will crash..

```
Inv memory userSign = processForInvestor[_investor][_index];

userSign.lockForUser / userSign.part >= period - (months - 3)
```

Recommendation

There should be a *require* statement to ensure *_lockForUser* variable is never higher than the value of *period*.

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
●	L09	Dead Code Elimination

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L227,235,599,612,630,742,783

Description

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

```
claimForUser
lockUpFor
renounceRole
revokeRole
grantRole
transferOwnership
renounceOwnership
```

Recommendation

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

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L688

Description

Constant state variables should be declared constant to save gas.

```
BACK_ROLE
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L705,711,724,743,744,745,783,844,688,690 and 3 more

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.

```
daysInSeconds
period
BACK_ROLE
_investor
_index
_typeVesting
_lockForUser
startVestingForUser
claimedTokenForUser
...
```

Recommendation

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

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L564,661,174,142,150,85,101,60

Description

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

```
toString
toHexString
reset
decrement
_msgData
_setRoleAdmin
_checkRole
```

Recommendation

Remove unused functions.

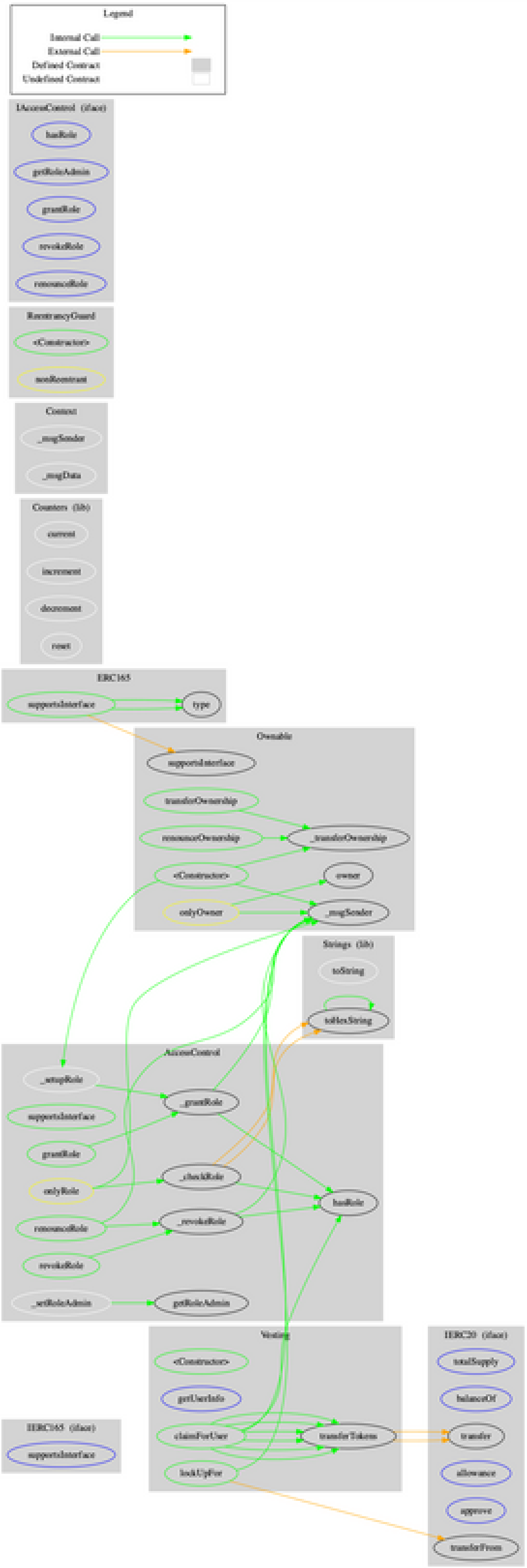
Contract Functions

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
IERC165	Interface			
	supportsInterface	External		-
ERC165	Implementation	IERC165		
	supportsInterface	Public		-
Strings	Library			
	toString	Internal		
	toHexString	Internal		
	toHexString	Internal		
Counters	Library			
	current	Internal		
	increment	Internal	✓	
	decrement	Internal	✓	
	reset	Internal	✓	
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_transferOwnership	Internal	✓	

IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
ReentrancyGuard	Implementation			
	<Constructor>	Public	✓	-
IAccessControl	Interface			
	hasRole	External		-
	getRoleAdmin	External		-
	grantRole	External	✓	-
	revokeRole	External	✓	-
	renounceRole	External	✓	-
AccessControl	Implementation	Context, IAccessControl, ERC165, Ownable		
	supportsInterface	Public		-
	hasRole	Public		-
	_checkRole	Internal		
	getRoleAdmin	Public		-
	grantRole	Public	✓	onlyOwner
	revokeRole	Public	✓	onlyOwner
	renounceRole	Public	✓	-
	_setupRole	Internal	✓	
	_setRoleAdmin	Internal	✓	
	_grantRole	Private	✓	
	_revokeRole	Private	✓	

Vesting	Implementation	AccessControl, ReentrancyGuard		
	<Constructor>	Public	✓	-
	getUserInfo	External		-
	lockUpFor	Public	✓	nonReentrant onlyOwner
	claimForUser	Public	✓	nonReentrant
	transferTokens	Internal	✓	

Contract Flow



Domain Info

Domain Name	quarashi.network
Registry Domain ID	74953879daf3467fb29bb3e7bb89fc11-DONUTS
Creation Date	2021-02-20T06:45:34Z
Updated Date	2021-07-23T19:03:45Z
Registry Expiry Date	2023-02-20T06:45:34Z
Registrar WHOIS Server	whois.namecheap.com
Registrar URL	https://www.namecheap.com/
Registrar	NameCheap, Inc.
Registrar IANA ID	1068

The domain has been created about 1 year before the creation of the audit. It will expire in 12 months.

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

Summary

This is the vesting contract of Quarashi Network. The contract contains 2 main functionalities, locking tokens for users so that they can unvest them in the future, and a function to manually claim the vested tokens if the required time has elapsed. There is also a role system mechanism to ensure who is eligible for manual claiming.

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