Cyberscope

Audit Report

Quarashi

February 2022

Type	BEP20
Network	BSC
Address	0xfD0fD32A20532ad690731c2685d77c351015ebBa
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	Quarashi
Compiler Version	v0.4.23+commit.124ca40d
Optimization	200 runs
Licence	
Explorer	https://bscscan.com/token/0xfD0fD32A20532ad690731c2685d77c351015ebBa
Symbol	QUA
Decimals	18
Total Supply	200,000,000
Source	/contracts/Quarashi.sol
Domain	quarashi.network

Audit Updates

Initial Audit	25th February 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	medium
Location	contract.sol#L699

Description

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

```
    function transfer(address _to, uint256 _value) public returns (bool
_success) {
        require(!paused || msg.sender == owner);
        return super.transfer(_to, _value);
    }
```

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.

MT - Mint Tokens

Criticality	critical
Location	contract.sol#L362

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 _to,
    uint256 _amount
  )
    hasMintPermission
    canMint
    public
    returns (bool)
  {
    totalSupply_ = totalSupply_.add(_amount);
    balances[_to] = balances[_to].add(_amount);
    emit Mint(_to, _amount);
    emit Transfer(address(0), _to, _amount);
    return true;
  }
```

Recommendation

The contract has a *mintingFinished* mechanism to protect itself from inflation and abuse. When the team has minted enough tokens for their business logic, its highly advised that they call the above functionality.

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.

Freeze Functionality

The contract supports a feature called “Freeze”. Users can send an amount of tokens to an address and choose when these tokens will be available. Essentially, the recipient will have the tokens but will not be able to use them until the time period elapses.

According to the contract’s configuration, the *canMint* modifier locks during the contract initialization. That means that the *mintAndFreeze* function is redundant since it can never be called.

Contract Diagnostics

Critical Medium Minor

Severity	Code	Description
	L01	Public Function could be Declared External
	L04	Conformance to Solidity Naming Conventions
	L09	Dead Code Elimination

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L98,134,140,229,252,308,317,362,382,415 and 11 more

Description

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

```
decimals
symbol
name
mintAndFreeze
unpause
pause
burn
releaseAll
freezeTo
...
```

Recommendation

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

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L107,122,168,169,170,196,209,210,230,231 and 27 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.

```
_value
_to
_from
_until
_amount
_index
_addr
_owner
_newOwner
...
```

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#L57,41

Description

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

```
mul
div
```

Recommendation

Remove unused functions.

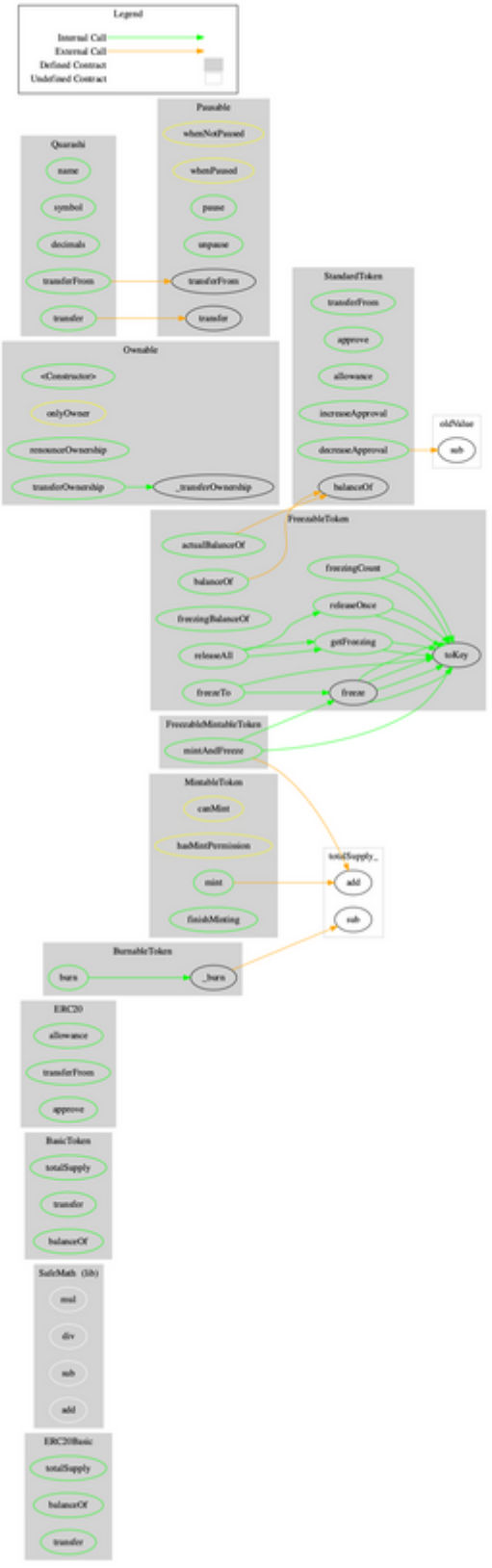
Contract Functions

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
ERC20Basic	Implementation			
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
SafeMath	Library			
	mul	Internal		
	div	Internal		
	sub	Internal		
	add	Internal		
BasicToken	Implementation	ERC20Basic		
	totalSupply	Public		-
	transfer	Public	✓	-
	balanceOf	Public		-
ERC20	Implementation	ERC20Basic		
	allowance	Public		-
	transferFrom	Public	✓	-
	approve	Public	✓	-
StandardToken	Implementation	ERC20, BasicToken		
	transferFrom	Public	✓	-
	approve	Public	✓	-
	allowance	Public		-
	increaseApproval	Public	✓	-

	decreaseApproval	Public	✓	-
Ownable	Implementation			
	<Constructor>	Public	✓	-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_transferOwnership	Internal	✓	
MintableToken	Implementation	StandardToken, Ownable		
	mint	Public	✓	hasMintPermission canMint
	finishMinting	Public	✓	onlyOwner canMint
FreezableToken	Implementation	StandardToken		
	balanceOf	Public		-
	actualBalanceOf	Public		-
	freezingBalanceOf	Public		-
	freezingCount	Public		-
	getFreezing	Public		-
	freezeTo	Public	✓	-
	releaseOnce	Public	✓	-
	releaseAll	Public	✓	-
	toKey	Internal		
	freeze	Internal	✓	
BurnableToken	Implementation	BasicToken		
	burn	Public	✓	-
	_burn	Internal	✓	
Pausable	Implementation	Ownable		
	pause	Public	✓	onlyOwner whenNotPaused

	unpause	Public	✓	onlyOwner whenPaused
FreezableMintableToken	Implementation	FreezableToken, MintableToken		
	mintAndFreeze	Public	✓	onlyOwner canMint
Consts	Implementation			
Quarashi	Implementation	Consts, FreezableMintableToken, BurnableToken, Pausable		
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	transferFrom	Public	✓	-
	transfer	Public	✓	-

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

The Smart Contract analysis reported no compiler errors, 1 medium and 1 critical threat issues. The contract Owner can Mint new tokens after initial deployment, and he can also pause transactions. Additionally, the contract contains some extra features like transfer and time-lock tokens. These functions can be called by the users. There are also some unused declared variables *CONTINUE_MINTING*, *START_TIME* and *TARGET_USER*.The audit describes these features and notes some minor recommendations. A multi-wallet signing pattern will provide security against potential hacks.

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