



Cyberscope

Audit Report

Quarashi

February 2022

Type BEP20

Network BSC

Address 0xfD0fD32A20532ad690731c2685d77c351015ebBa

Audited by © cyberscope

Table of Contents

Table of Contents	1
Contract Review	2
Audit Updates	2
Contract Analysis	3
ST - Stop Transactions	4
Description	4
Recommendation	4
Freeze Functionality	5
Contract Diagnostics	6
L01 - Public Function could be Declared External	7
Description	7
Recommendation	7
L04 - Conformance to Solidity Naming Conventions	8
Description	8
Recommendation	8
L09 - Dead Code Elimination	9
Description	9
Recommendation	9
Contract Functions	10
Contract Flow	13
Domain Info	14
Summary	15
Disclaimer	16
About Cyberscope	17

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	25th February 2022

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.

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.

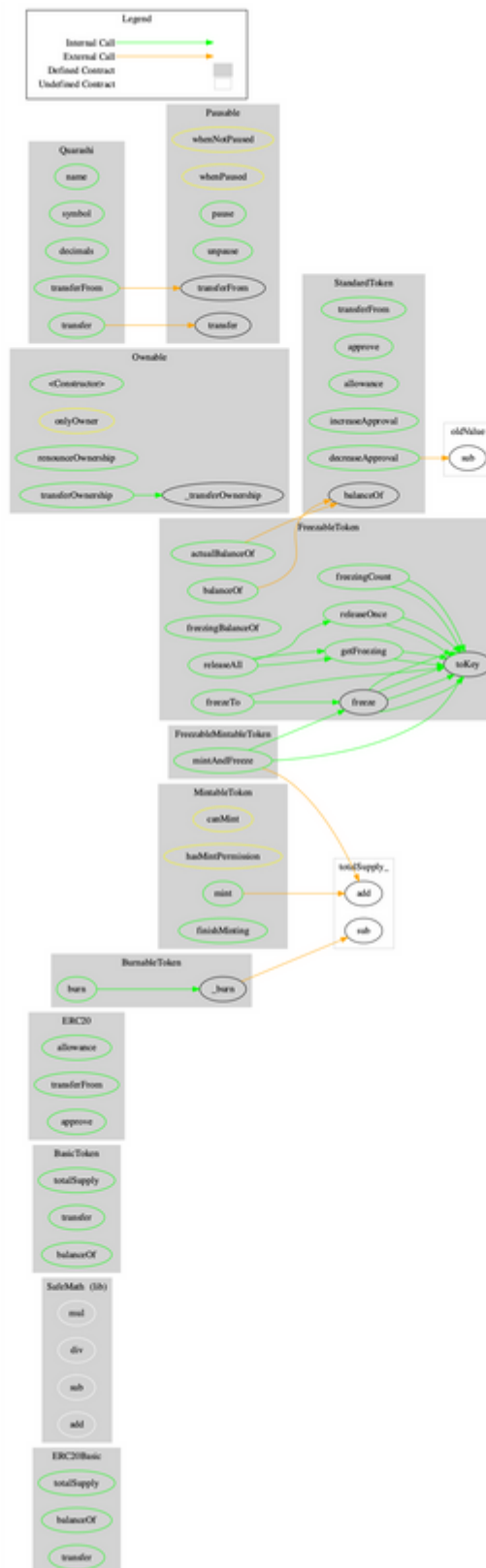
Contract Functions

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 and only 1 medium threat issue. The contract Owner can 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 contract contains mint functionality but it cannot be used since the mint permissions have been renounced. 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.

Cyberscope 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>



Audit Report

QUARASHI NETWORK

February 2022

Type	ERC-20
Network	ETH
Address	0x0aFf88B4Cf3015c9c17f1dA1FCCb88C632F3505E
Audited by	© coinscope

Table of Contents

Table of Contents	1
Contract Review	2
Audit Updates	2
Contract Analysis	2
Unused functionality	4
Freeze Functionality	4
Contract Diagnostics	5
L01 - Public Function could be Declared External	6
Description	6
Recommendation	6
L04 - Conformance to Solidity Naming Conventions	7
Description	7
Recommendation	7
L09 - Dead Code Elimination	8
Description	8
Recommendation	8
Contract Functions	9
Contract Flow	12
Domain Info	13
Summary	14
Disclaimer	15
About Coinscope	16

Contract Review

Contract Name	MainToken
Compiler Version	v0.4.24+commit.e67f0147
Optimization	200 runs
Licence	GNU GPLv3
Explorer	https://etherscan.io/token/0x0aff88b4cf3015c9c17f1da1fc8b88c632f3505e
Symbol	QUA
Decimals	18
Total Supply	799,176,692.33293494629

Audit Updates

Initial Audit	9th of 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

Unused functionality

The contract has the ability to mint new tokens. All the functions that are related to minting require the *canMint* modifier. The *canMint* modifier is a boolean flag that determines if the mint action can start. This flag initially was enabled but now it is disabled. The mint flag is a one-way flag, that means that it can not switch on again. Thus, the mint functionality cannot be used any more.

The transaction that switched off the mint feature

<https://etherscan.io/tx/0x1fe29e6c3af500a7b6f5e323992e30b0fece1af113f14d9d2831e448d8c88bdf>

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 she will not be able to use them until the time period elapses.

The contract combines the mint and freeze functionality. That means that the contract owner had the ability to mint and time-lock tokens to an address. This feature is not available any more since it required the *canMint* modifier

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 and 15 more

Description

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

```
decimals  
symbol  
name  
...
```

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 and 31 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  
...
```

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.

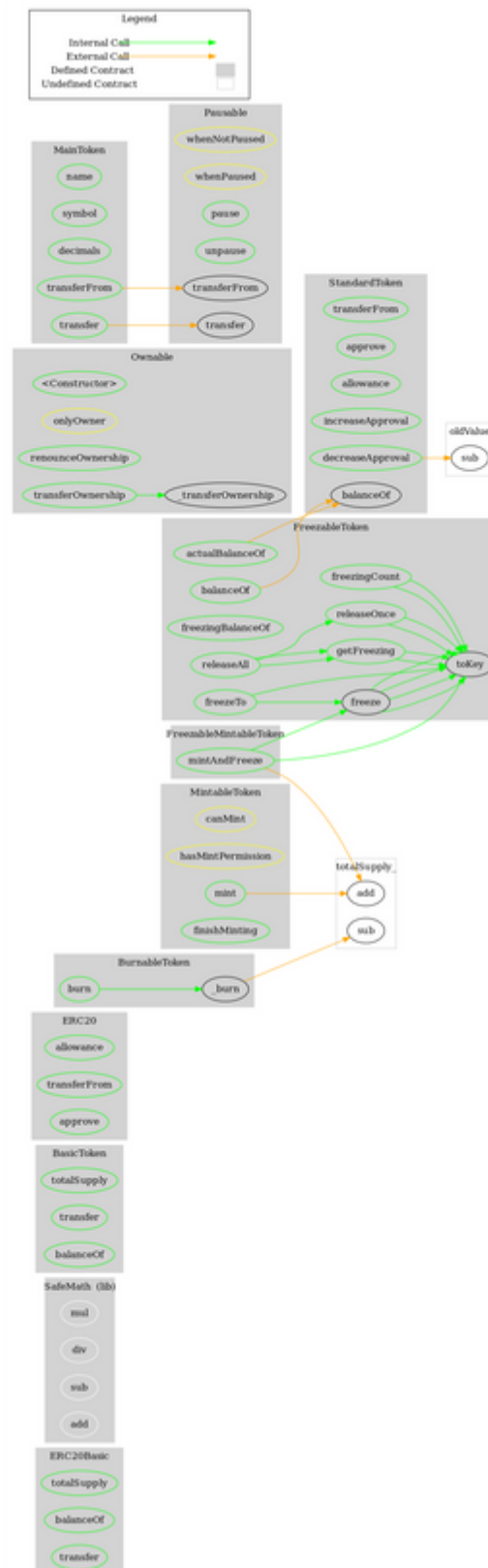
Contract Functions

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			
MainToken	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 12 months before the creation of the audit. It will expire in about 1 year.

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

Summary

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. Additionally, the contract contains some extra features like transfer and time-lock tokens. These features can be issued by the users. The audit describes these features and notes some minor recommendations.

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.

Coinscope 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 Coinscope 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 Coinscope 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 Coinscope team disclaims any liability for the resulting losses.

About Coinscope

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.

Coinscope is aiming to make crypto discoverable and efficient globally. It provides all the essential tools to assist users draw their own conclusions.



The Coinscope.co team

<https://www.coinscope.co>



Cyberscope

Audit Report

Quarashi Staking BSC

April 2022

Type BEP20

Network BSC

Address 0xee7b65e341de03621964c0f2cddee78690e2cee9

Audited by © cyberscope

Table of Contents

Table of Contents	1
Contract Review	3
Source Files	3
Audit Updates	3
Contract Analysis	4
Pools	4
Reward calculation	5
Contract Owner privileges	5
Deposit Info Id Event Emit	6
Description	6
Recommendation	6
Minimum Deposit Amount	7
Description	7
Recommendation	7
Contract Diagnostics	8
L01 - Public Function could be Declared External	9
Description	9
Recommendation	9
L04 - Conformance to Solidity Naming Conventions	10
Description	10
Recommendation	10
L09 - Dead Code Elimination	11
Description	11
Recommendation	11
L13 - Divide before Multiply Operation	12
Description	12

Recommendation	12
L14 - Uninitialized Variables in Local Scope	13
Description	13
Recommendation	13
Contract Functions	14
Contract Flow	16
Summary	17
Disclaimer	18
About Cyberscope	19

Contract Review

Contract Name	QuaStaking
Compiler Version	v0.8.11+commit.d7f03943
Optimization	200000 runs
Licence	MIT
Explorer	https://bscscan.com/token/0xee7b65e341de03621964c0f2cdaee78690e2cee9

Source Files

Filename	SHA256
contract.sol	44757d01b05df3de640fa3381690fec00ea926a303367b2d8d2ee182304f313c

Audit Updates

Initial Audit	13th April 2022
Corrected	

Contract Analysis

The contract implements a basic staking feature. The users have the ability to deposit tokens to three different pools. Each pool provides a different combination of A.P.Y. (Annual Percentage Yield), locking period and commission. The commission is only applied if the user withdraws the tokens earlier than the locking period.

Pools

The pool options are 3 and cannot be changed.

Pool Id	A.P.Y. (percentage)	Locking Period (months)	Commission (percentage)
0	0.0055	1	0.01
1	0.0125	6	0.03
3	0.028	12	0.08

Reward calculation

The APY percentage is added every month to the previous month's APY. So for instance, if a user stake 10000 tokens in the pool id 1, then the withdrawn amount after 6 months will be 10773.9. As a result the APY does not work as an annual percentage but as an accumulated monthly percentage.

Early Withdraw

The depositors have the ability to withdraw the tokens earlier than the locking period. As a result the depositor will receive the APY percentage proportional to the time that has been elapsed. Additionally, the depositor will be taxed with a commission amount. The commission amount is calculated based on the initial deposit, not in the awarded amount.

Contract Owner privileges

- The Admin role is renounced
- The Admin role has the ability to set the commission address
- The Admin role has the ability to withdraw the contract's excessed tokens.

Deposit Info Id Event Emit

Criticality	minor
Location	contract.sol#L591,598

Description

Since the `TokensStaked` and `Withdraw()` are based on the user's deposit info index, it would be more informative to emit the `depositInfoId` number in the event as well.

Recommendation

The `depositInfoId` could be emitted in the events.

Minimum Deposit Amount

Criticality	minor
Location	contract.sol#L666

Description

The calculation of award amount is a production of division. Hence, there is a minimum amount that the division will return zero.

The minimum amount are:

- if a user deposits 181 tokens in the pool id 1, then the awards amount will be zero.
- if a user deposits 79 tokens in the pool id 2, then the awards amount will be zero.
- if a user deposits 35 tokens in the pool id 3, then the awards amount will be zero.

```
_maxUnstakeAmount * pools[_poolId].APY / PERSENT_BASE;
```

Recommendation

The contract could have a minimum amount check, so it is guaranteed that all the depositors will receive rewards.

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
●	L13	Divide before Multiply Operation
●	L14	Uninitialized Variables in Local Scope

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L463,476,494

Description

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

```
renounceRole  
revokeRole  
grantRole
```

Recommendation

Use the external attribute for functions never called from the contract

L04 - Conformance to Solidity Naming Conventions

Criticality

minor

Location

contract.sol#L642,657,697,753,784,785

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.

```
_depositInfoId  
_user  
_poolId  
_commissionAddress
```

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#L527,248,223

Description

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

```
toString  
toHexString  
_setRoleAdmin
```

Recommendation

Remove unused functions.

L13 - Divide before Multiply Operation

Criticality	minor
Location	contract.sol#L783

Description

Performing divisions before multiplications may cause lose of prediction.

```
stakingDays = (block.timestamp - deposit.start) % MONTH / DAY
```

Recommendation

The multiplications should be prior to the divisions.

L14 - Uninitialized Variables in Local Scope

Criticality

minor

Location

contract.sol#L665,764,800,633,806

Description

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

```
i  
commissionAmount
```

Recommendation

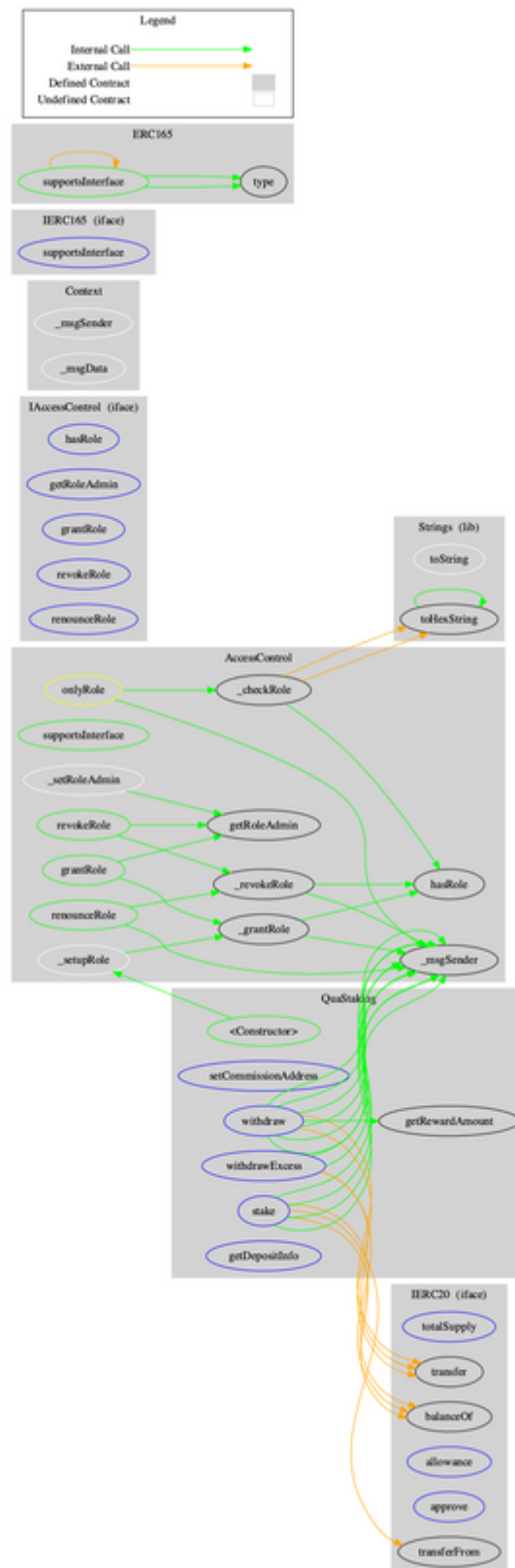
All the local scoped variables should be initialized.

Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
IAccessControl	Interface			
	hasRole	External		-
	getRoleAdmin	External		-
	grantRole	External	✓	-
	revokeRole	External	✓	-
	renounceRole	External	✓	-
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
Strings	Library			
	toString	Internal		
	toHexString	Internal		
	toHexString	Internal		
IERC165	Interface			
	supportsInterface	External		-
ERC165	Implementation	IERC165		

	supportsInterface	Public		-
AccessControl	Implementation	Context, IAccessControl, ERC165		
	supportsInterface	Public		-
	hasRole	Public		-
	_checkRole	Internal		
	getRoleAdmin	Public		-
	grantRole	Public	✓	onlyRole
	revokeRole	Public	✓	onlyRole
	renounceRole	Public	✓	-
	_setupRole	Internal	✓	
	_setRoleAdmin	Internal	✓	
	_grantRole	Internal	✓	
	_revokeRole	Internal	✓	
QuaStaking	Implementation	AccessControl		
	<Constructor>	Public	✓	-
	setCommissionAddress	External	✓	onlyRole
	stake	External	✓	-
	withdraw	External	✓	-
	withdrawExcess	External	✓	onlyRole
	getDepositInfo	External		-
	getRewardAmount	Public		-

Contract Flow



Summary

Quarashi Staking is a typical implementation of staking functionality. The users have the ability to stake tokens and get the rewards once the locked period has elapsed. This audit focuses on the business logic and potential optimizations.

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>



Cyberscope

Audit Report

Quarashi Staking ETH

April 2022

Type ERC20

Network ETH

Address 0xee7B65E341DE03621964c0f2CDAee78690e2cEe9

Audited by © cyberscope

Table of Contents

Table of Contents	1
Contract Review	2
Audit Updates	2
Source Files	3
Contract Analysis	4
Pools	4
Reward calculation	5
Contract Owner privileges	5
Deposit Info Id Event Emit	6
Description	6
Recommendation	6
Minimum Deposit Amount	7
Description	7
Recommendation	7
Contract Diagnostics	8
L04 - Conformance to Solidity Naming Conventions	9
Description	9
Recommendation	9
L14 - Uninitialized Variables in Local Scope	10
Description	10
Recommendation	10
Contract Functions	11
Contract Flow	13
Summary	14
Disclaimer	15
About Cyberscope	16

Contract Review

Contract Name	QuaStaking
Compiler Version	v0.8.11+commit.d7f03943
Optimization	200000 runs
Licence	MIT
Explorer	https://etherscan.io/address/0xee7B65E341DE03621964c0f2CDAee78690e2cEe9

Audit Updates

Initial Audit	13th April 2022
Corrected	

Source Files

Filename	SHA256
@openzeppelin/contracts/access/AccessControl.sol	0b280a0fe505b5b8bcb700e0b1f6242acf73e0b509372ef3acc46db051512e32
@openzeppelin/contracts/access/IAccessControl.sol	d03c1257f2094da6c86efa7aa09c1c07ebd33dd31046480c5097bc2542140e45
@openzeppelin/contracts/interfaces/IERC20.sol	1e78c90db4e4838c0a603bfbfd2bafa2c38ba997769043e2a6045ad9e73764b60
@openzeppelin/contracts/token/ERC20/IERC20.sol	c2b06bb4572bb4f84bfc5477dadcfcc497cb66c3a1bd53480e68bedc2e154a6
@openzeppelin/contracts/utils/Context.sol	1458c260d010a08e4c20a4a517882259a23a4baa0b5bd9add9fb6d6a1549814a
@openzeppelin/contracts/utils/introspection/ERC165.sol	8806a632d7b656cadb8133ff8f2acae4405b3a64d8709d93b0fa6a216a8a6154
@openzeppelin/contracts/utils/introspection/IERC165.sol	701e025d13ec6be09ae892eb029cd83b3064325801d73654847a5fb11c58b1e5
@openzeppelin/contracts/utils/Strings.sol	8597c62818dc6c6cf85c21179b90b714fb4f70a4347ca2eed23e88c87b08b8a1
contracts/QuaStaking.sol	b7a836810301a9c0aa8a3daa59618d2c01dc91992c79ada47201dfbb7644985e

Contract Analysis

The contract implements a basic staking feature. The users have the ability to deposit tokens to three different pools. Each pool provides a different combination of A.P.Y. (Annual Percentage Yield), locking period and commission. The commission is only applied if the user withdraws the tokens earlier than the locking period.

Pools

The pool options are 3 and cannot be changed.

Pool Id	A.P.Y. (percentage)	Locking Period (months)	Commission (percentage)
0	0.0055	1	0.01
1	0.0125	6	0.03
3	0.028	12	0.08

Reward calculation

The APY percentage is added every month to the previous month's APY. So for instance, if a user stake 10000 tokens in the pool id 1, then the withdrawn amount after 6 months will be 10773.9. As a result the APY does not work as an annual percentage but as an accumulated monthly percentage.

Early Withdraw

The depositors have the ability to withdraw the tokens earlier than the locking period. As a result the depositor will receive the APY percentage proportional to the time that has been elapsed. Additionally, the depositor will be taxed with a commission amount. The commission amount is calculated based on the initial deposit, not in the awarded amount.

Contract Owner privileges

- The Admin role is renounced
- The Admin role has the ability to set the commission address
- The Admin role has the ability to withdraw the contract's excessed tokens.

Deposit Info Id Event Emit

Criticality	minor
Location	contract.sol#L34,41

Description

Since the `TokensStaked` and `Withdraw()` are based on the user's deposit info index, it would be more informative to emit the `depositInfoId` number in the event as well.

Recommendation

The `depositInfoId` could be emitted in the events.

Minimum Deposit Amount

Criticality	minor
Location	contract.sol#L109

Description

The calculation of award amount is a production of division. Hence, there is a minimum amount that the division will return zero.

The minimum amount are:

- if a user deposits 181 tokens in the pool id 1, then the awards amount will be zero.
- if a user deposits 79 tokens in the pool id 2, then the awards amount will be zero.
- if a user deposits 35 tokens in the pool id 3, then the awards amount will be zero.

```
_maxUnstakeAmount * pools[_poolId].APY / PERSENT_BASE;
```

Recommendation

The contract could have a minimum amount check, so it is guaranteed that all the depositors will receive rewards.

Contract Diagnostics

● Critical ● Medium ● Minor

Severity	Code	Description
●	L04	Conformance to Solidity Naming Conventions
●	L14	Uninitialized Variables in Local Scope

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contracts/QuaStaking.sol#L85,100,140,196,227,228

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.

```
_depositInfoId  
_user  
_poolId  
_commissionAddress
```

Recommendation

Follow the Solidity naming convention.

<https://docs.soliditylang.org/en/v0.4.25/style-guide.html#naming-conventions>

L14 - Uninitialized Variables in Local Scope

Criticality

minor

Location

contracts/QuaStaking.sol#L108,207,243,76,249

Description

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

```
i  
commissionAmount
```

Recommendation

All the local scoped variables should be initialized.

Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
AccessControl	Implementation	Context, IAccessControl, ERC165		
	supportsInterface	Public		-
	hasRole	Public		-
	_checkRole	Internal		
	getRoleAdmin	Public		-
	grantRole	Public	✓	onlyRole
	revokeRole	Public	✓	onlyRole
	renounceRole	Public	✓	-
	_setupRole	Internal	✓	
	_setRoleAdmin	Internal	✓	
	_grantRole	Internal	✓	
	_revokeRole	Internal	✓	
IAccessControl	Interface			
	hasRole	External		-
	getRoleAdmin	External		-
	grantRole	External	✓	-
	revokeRole	External	✓	-
	renounceRole	External	✓	-
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-

	transferFrom	External	✓	-
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
ERC165	Implementation	IERC165		
	supportsInterface	Public		-
IERC165	Interface			
	supportsInterface	External		-
Strings	Library			
	toString	Internal		
	toHexString	Internal		
	toHexString	Internal		
QuaStaking	Implementation	AccessControl		
	<Constructor>	Public	✓	-
	setCommissionAddress	External	✓	onlyRole
	stake	External	✓	-
	withdraw	External	✓	-
	withdrawExcess	External	✓	onlyRole
	getDepositInfo	External		-
	getRewardAmount	Public		-

Contract Flow



Summary

Quarashi Staking is a typical implementation of staking functionality. The users have the ability to stake tokens and get the rewards once the locked period has elapsed. This audit focuses on the business logic and potential optimizations.

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>



Cyberscope

Audit Report

Quarashi Bridge

April 2022

Network BSC/ETH

Address 0x1460e300fa13F6B1f8Fd07bA1A9772A68778c97

Audited by © cyberscope

Table of Contents

Table of Contents	1
Contract Review	3
Binance Smart Chain	3
Ethereum Chain	3
Source Files	4
Audit Updates	5
Contract Analysis	6
Audit Scope	6
Blockchain Network Id	7
Signatures Validators	7
Network Fees	7
Contract Diagnostics	8
MC - Missing Check	9
Description	9
Recommendation	9
L01 - Public Function could be Declared External	10
Description	10
Recommendation	10
L04 - Conformance to Solidity Naming Conventions	11
Description	11
Recommendation	11
L07 - Missing Events Arithmetic	12
Description	12
Recommendation	12
L09 - Dead Code Elimination	13
Description	13

Recommendation	13
Contract Functions	14
Contract Flow	19
Summary	20
Disclaimer	21
About Cyberscope	22

Contract Review

Binance Smart Chain

Contract Name	SwapContract
Compiler Version	v0.8.4+commit.c7e474f2
Optimization	999999 runs
Explorer	https://bscscan.com/address/0x1460e300fa13F6B1f8Fd07bA1A9772A68778c97

Ethereum Chain

Contract Name	SwapContract
Compiler Version	v0.8.4+commit.c7e474f2
Optimization	999999 runs
Explorer	https://etherscan.io/address/0x1460e300fa13f6b1f8fd07ba1a9772a68778c97a

Source Files

Filename	SHA256
@openzeppelin/contracts/access/AccessControl.sol	6877e8ac1c1d0febb541b95080314677794f9f3fa1062bfc5b3dc875ba1e3d4b
@openzeppelin/contracts/access/AccessControlEnumerable.sol	80f821aef438c0e45949bfd9b18b57c547f8df0b0e376d8ecdf8605c6481316d
@openzeppelin/contracts/security/Pausable.sol	18e9c654be32b98232273f21457785e5ca5209ee978e84035634cf947e4a10bb
@openzeppelin/contracts/token/ERC20/IERC20.sol	b2565dec975f684ef0edfa505e212d0d0b602e1311afab782ea06ea8d3f49bb6
@openzeppelin/contracts/utils/Context.sol	810e94131350fcdbbed52a2a10523565ef98f4c9dda441495262923f251b0d12
@openzeppelin/contracts/utils/introspection/ERC165.sol	381b0589da0e1a32242d7314905d2cc6edd8dce8193ddb6bfacc5b685e311422
@openzeppelin/contracts/utils/introspection/IERC165.sol	072805b211a653c333b232a3199b9e65fa7b82fc7a40ee5a3bc8a2dadd1cba01
@openzeppelin/contracts/utils/Strings.sol	3b2b0d75c7e5688950d3b6e63e46473054395dad6e390431f73febb2199913c5
@openzeppelin/contracts/utils/structs/EnumerableSet.sol	f7b69cd8e46762273807f9b8282690baaae7a5458c99373dabea5882ffd6493c

contracts/ECDSAOffsetRecovery.sol	c69ef361fff9628e4093a4e2c7f7eaba8032f3eb2948208998ca221cda1af8ee
contracts/swapContract.sol	ef5f990aaf7ec1424149b2315fdc2c62109853cb67877db057bf4bc72b4d4779
contracts/TransferHelper.sol	4ec1019b6694d21e9b2a32fbf86842cbc44fe672d4e0eb60c174bf321b49da05

Audit Updates

Initial Audit	1st April 2022
Corrected	

Contract Analysis

The Quarashi Swap contracts is a mechanism for converting tokens from one network to another and vice versa. The mechanism is highly based on an off-chain mechanism. The mechanism watches the events from the contract in each network and emits the necessary events. For instance, when a user converts tokens from ETH to BSC network the following steps are taking place:

1. The “transferToOtherBlockchain” is getting called in the ETH network.
2. The tokens are transferred from the user to the contract’s address in the ETH.
3. An event is emitted.
4. The off-chain server caches the event.
5. The off-chain server prepares the call to the BSC bridge contract by signing the transaction. The signing is issued by the validator role. The validator role should be at least one.
6. The off-chain server executes the “transferToUserWithFee” in the BSC network.
7. The “transferToUserWithFee” validates that the transaction has been correctly signed by the validators.
8. The tokens are transferred to the user’s address from the contract in the BSC network.

That means that the bridge contract from each network should have a sufficient amount of tokens in order to respond to the demand.

Audit Scope

This audit focuses on the on-chain bridge contracts. It assumes that the off-chain source is safe and cannot be manipulated.

Blockchain Network Id

The contract could use the chain id instead of an arbitrary number for the blockchain enumeration. As a result, it will be easier tracing and more difficult to be manipulated in a malicious way. The chain id could either be configured by the contract owner or the contract could use an oracle pattern for setting the network automatically.

Read more about potential manipulations in the EIP-155 proposal and the “chainID” parameter in the chain-specification documentation.

<https://eips.ethereum.org/EIPS/eip-155>

<https://openethereum.github.io/Chain-specification>

Signatures Validators

The contract is using off-chain signatures in order to validate the transaction. The signatures are represented by a concatenated string. It could be more readable and easier for processing if the signature structure was an array of strings.

Network Fees

The tokens that are received from another network are taxed by a fee. This fee is different per network. If the fees are taking into account the transaction fees, then the fees for each “transferToUserWithFee” could be provided off-chain. That way each transaction would better estimate the fees that should be kept.

Contract Diagnostics

● Critical ● Medium ● Minor

Severity	Code	Description
●	MC	Missing Check
●	L01	Public Function could be Declared External
●	L04	Conformance to Solidity Naming Conventions
●	L07	Missing Events Arithmetic
●	L09	Dead Code Elimination

MC - Missing Check

Criticality	minor
Location	contract.sol#L152,203

Description

The contract is based on two functions. The `transferToOtherBlockchain` sends funds to other networks and the `transferToUserWithFee` receives funds from other networks. Both functions send tokens to/from the user's account. This operation is safely accomplished by the `safeTransfer` method. The contract could embody an extra precondition that checks if the user's funds are sufficient in the `transferToOtherBlockchain` function. Respectively, it could check if the contract's balance is sufficient in the `transferToUserWithFee` function.

The contract could also check if the issued amount is more than the corresponding network's fees.

```
TransferHelper.safeTransferFrom(address(tokenAddress), sender, address(this),
amount);
///
uint256 fee = feeAmountOfBlockchain[numOfThisBlockchain];
uint256 amountWithoutFee = amountWithFee - fee;
TransferHelper.safeTransfer(address(tokenAddress), user, amountWithoutFee);
TransferHelper.safeTransfer(address(tokenAddress), feeAddress, fee);
```

Recommendation

The contract could check if the sender's balance has at least the amount that is required.

```
require(
    tokenAddress.balanceOf(sender) >= amount,
    "swapContract: Not enough balance"
);
```

L01 - Public Function could be Declared External

Criticality	minor
Location	contracts/swapContract.sol#L379,387,395

Description

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

```
isRelayer  
isManager  
isOwner
```

Recommendation

Use the external attribute for functions never called from the contract

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contracts/swapContract.sol#L310,319,327,337

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.

```
_minConfirmationBlocks  
_maxGasPrice  
_minTokenAmount  
_minConfirmationSignatures
```

Recommendation

Follow the Solidity naming convention.

<https://docs.soliditylang.org/en/v0.4.25/style-guide.html#naming-conventions>

L07 - Missing Events Arithmetic

Criticality

minor

Location

contracts/swapContract.sol#L310

Description

Detected missing events for critical arithmetic parameters. There are functions that have no event emitted, so it is difficult to track off-chain changes.

```
minConfirmationSignatures = _minConfirmationSignatures
```

Recommendation

Emit an event for critical parameter changes.

L09 - Dead Code Elimination

Criticality

minor

Location

contracts/TransferHelper.sol#L7,50

Description

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

```
safeTransferETH  
safeApprove
```

Recommendation

Remove unused functions.

Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
IAccessControl	Interface			
	hasRole	External		-
	getRoleAdmin	External		-
	grantRole	External	✓	-
	revokeRole	External	✓	-
	renounceRole	External	✓	-
AccessControl	Implementation	Context, IAccessControl, ERC165		
	supportsInterface	Public		-
	hasRole	Public		-
	_checkRole	Internal		
	getRoleAdmin	Public		-
	grantRole	Public	✓	onlyRole
	revokeRole	Public	✓	onlyRole
	renounceRole	Public	✓	-
	_setupRole	Internal	✓	
	_setRoleAdmin	Internal	✓	
	_grantRole	Private	✓	
	_revokeRole	Private	✓	
IAccessControlEnumerable	Interface			
	getRoleMember	External		-
	getRoleMemberCount	External		-

AccessControl Enumerable	Implementation	IAccessControlEnumerable, AccessControl		
	supportsInterface	Public		-
	getRoleMember	Public		-
	getRoleMemberCount	Public		-
	grantRole	Public	✓	-
	revokeRole	Public	✓	-
	renounceRole	Public	✓	-
	_setupRole	Internal	✓	
Pausable	Implementation	Context		
	<Constructor>	Public	✓	-
	paused	Public		-
	_pause	Internal	✓	whenNotPaused
	_unpause	Internal	✓	whenPaused
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
ERC165	Implementation	IERC165		
	supportsInterface	Public		-
IERC165	Interface			
	supportsInterface	External		-

Strings	Library			
	toString	Internal		
	toHexString	Internal		
	toHexString	Internal		
EnumerableSet	Library			
	_add	Private	✓	
	_remove	Private	✓	
	_contains	Private		
	_length	Private		
	_at	Private		
	add	Internal	✓	
	remove	Internal	✓	
	contains	Internal		
	length	Internal		
	at	Internal		
	add	Internal	✓	
	remove	Internal	✓	
	contains	Internal		
	length	Internal		
	at	Internal		
	add	Internal	✓	
	remove	Internal	✓	
	contains	Internal		
	length	Internal		
	at	Internal		
ECDSAOffsetRecovery	Implementation			
	getHashPacked	Public		-
	toEthSignedMessageHash	Public		-
	ecOffsetRecover	Public		-

SwapContract	Implementation	AccessControlEnumerable, Pausable, ECDSAOffsetRecovery		
	<Constructor>	Public	✓	-
	getOtherBlockchainAvailableByNum	External		-
	transferToOtherBlockchain	External	✓	whenNotPaused
	transferToUserWithFee	External	✓	onlyRelayer whenNotPaused
	addOtherBlockchain	External	✓	onlyOwner
	removeOtherBlockchain	External	✓	onlyOwner
	changeOtherBlockchain	External	✓	onlyOwner
	changeFeeAddress	External	✓	onlyOwnerAndManager
	setFeeAmountOfBlockchain	External	✓	onlyOwnerAndManager
	setMinConfirmationSignatures	External	✓	onlyOwner
	setMinTokenAmount	External	✓	onlyOwnerAndManager
	setMaxGasPrice	External	✓	onlyOwnerAndManager
	setMinConfirmationBlocks	External	✓	onlyOwnerAndManager
	transferOwnerAndSetManager	External	✓	onlyOwner
	pauseExecution	External	✓	onlyOwner
	continueExecution	External	✓	onlyOwner
	isOwner	Public		-
	isManager	Public		-
	isRelayer	Public		-
	isValidator	Public		-
	isProcessedTransaction	Public		-
TransferHelper	Library			
	safeApprove	Internal	✓	
	safeTransfer	Internal	✓	
	safeTransferFrom	Internal	✓	

	safeTransferETH	Internal	✓	
--	-----------------	----------	---	--

Contract Flow



Summary

Quarashi Bridge is a mechanism for converting tokens between different networks. This functionality is accomplished by combining an off-chain server and one on-chain contract for each network. This audit focuses on the on-chain network. It mentions some security concerns, potential optimizations and comments regarding the business logic.

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.

Cyberscope 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>



Audit Report

QUARASHI NETWORK

February 2022

Type	ERC-20
Network	ETH
Address	0xd35c06a2781f648c75290976ecF71E71582188B7
Audited by	© coinscope

Table of Contents

Table of Contents	1
Contract Review	2
Audit Updates	2
Contract Analysis	2
Freeze Functionality	4
Contract Diagnostics	5
L01 - Public Function could be Declared External	6
Description	6
Recommendation	6
L04 - Conformance to Solidity Naming Conventions	7
Description	7
Recommendation	7
L09 - Dead Code Elimination	8
Description	8
Recommendation	8
Contract Functions	9
Contract Flow	12
Domain Info	13
Summary	14
Disclaimer	15
About Coinscope	16

Contract Review

Contract Name	MainToken
Compiler Version	v0.4.24+commit.e67f0147
Optimization	200 runs
Licence	GNU GPLv3
Explorer	https://etherscan.io/token/0xd35c06a2781f648c75290976ecF71E71582188B7
Symbol	QUA
Decimals	18
Total Supply	1,000,000,000

Audit Updates

Initial Audit	19th of 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

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#L117,153,159,248,271,327,434,438,446,477 and 6 more

Description

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

```
decimals
symbol
name
unpause
burn
releaseAll
freezeTo
freezingCount
freezingBalanceOf
...
```

Recommendation

Use the external attribute for functions never called from the contract

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L126,141,187,188,189,215,228,229,249,250 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#L76,60

Description

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

```
mul  
div
```

Recommendation

Remove unused functions.

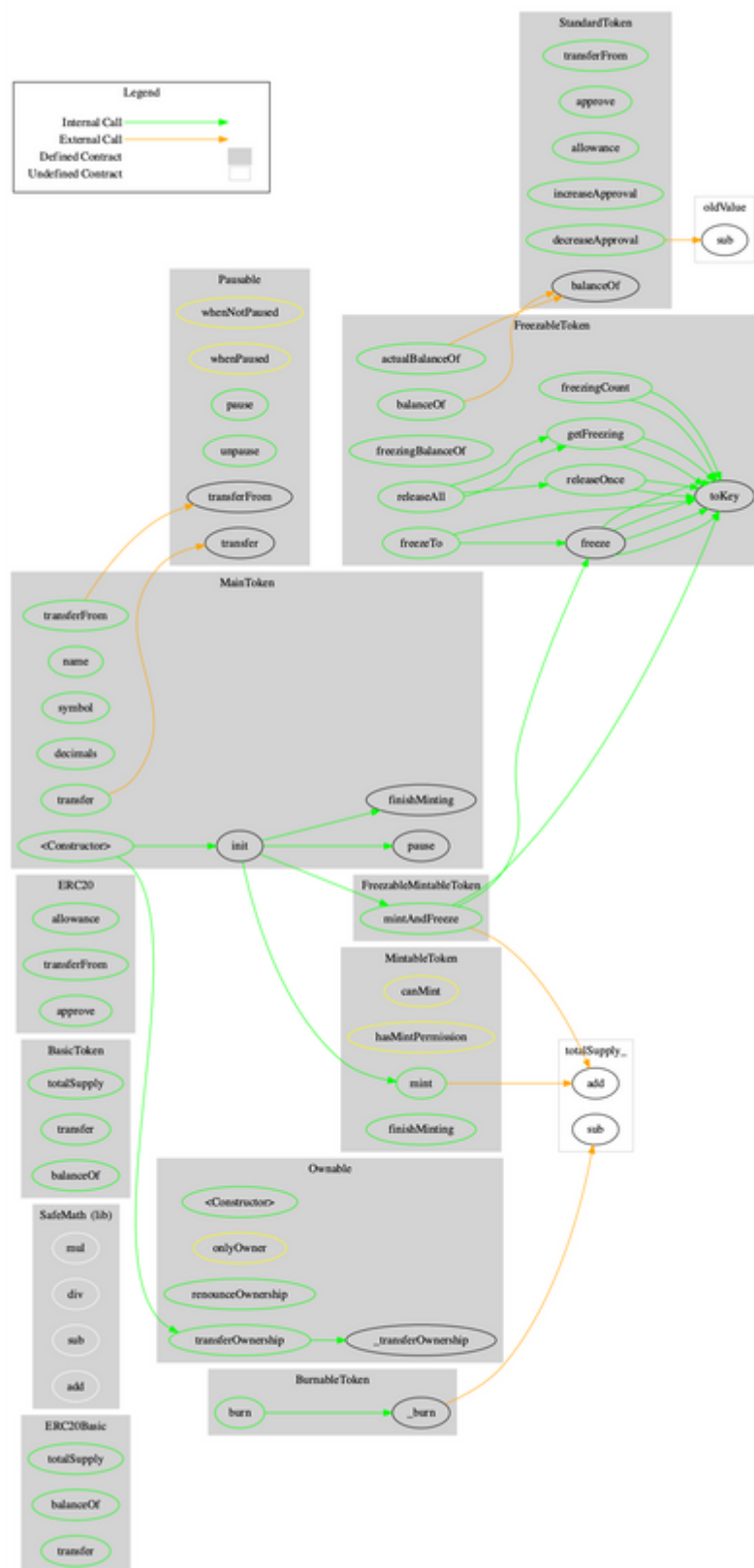
Contract Functions

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			
MainToken	Implementation	Consts, FreezableMintableToken, BurnableToken, Pausable		
	<Constructor>	Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	transferFrom	Public	✓	-
	transfer	Public	✓	-
	init	Private	✓	

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 12 months before the creation of the audit. It will expire in about 1 year.

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

Summary

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. Additionally, the contract contains some extra features like transfer and time-lock tokens. These functions can be called by the users. The audit describes these features and notes some minor recommendations.

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.

Coinscope 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 Coinscope 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 Coinscope 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 Coinscope team disclaims any liability for the resulting losses.

About Coinscope

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.

Coinscope is aiming to make crypto discoverable and efficient globally. It provides all the essential tools to assist users draw their own conclusions.



The Coinscope.co team

<https://www.coinscope.co>



Cyberscope

Audit Report

Quarashi Vesting

February 2022

Type BEP20

Network BSC

Address 0x3C7D0979cB9518F8050D516172FA4144384FC5b0

Audited by © cyberscope

Table of Contents

Table of Contents	1
Contract Review	2
Audit Updates	2
Contract Analysis	3
Vesting Functionality	3
Validation Statement Required	4
Description	4
Recommendation	4
Contract Diagnostics	5
L01 - Public Function could be Declared External	6
Description	6
Recommendation	6
L02 - State Variables could be Declared Constant	7
Description	7
Recommendation	7
L04 - Conformance to Solidity Naming Conventions	8
Description	8
Recommendation	8
L09 - Dead Code Elimination	9
Description	9
Recommendation	9
Contract Functions	10
Contract Flow	13
Domain Info	14
Summary	15
Disclaimer	16

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.

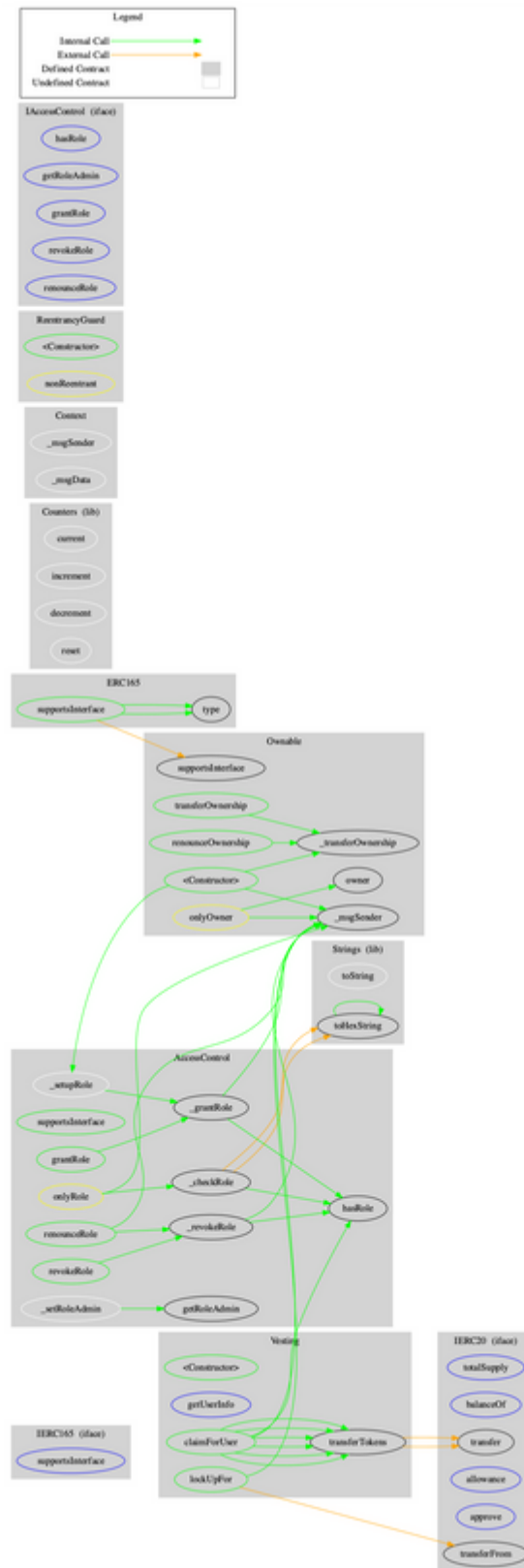
Contract Functions

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, Reentrancy Guard		
	<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>