Cyberscope

Audit Report

Quarashi Bridge

April 2022

Network	BSC/ETH
Address	0x1460e300fa13F6B1f8Fd07bA1A9772A68778c97
Audited by	© cyberscope

Table of Contents

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 orale 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

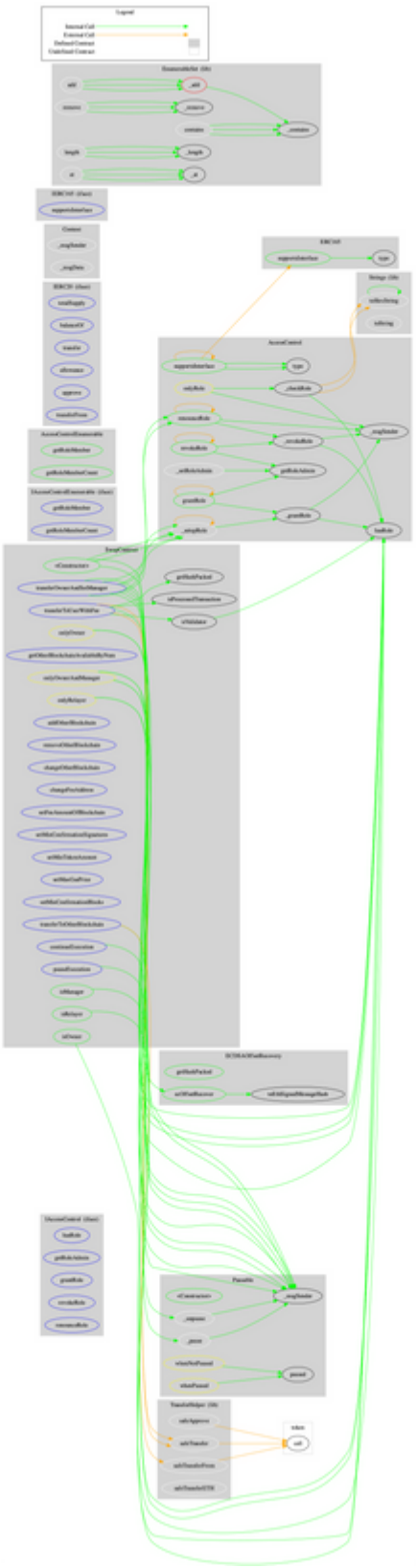
Remove unused functions.

Contract Functions

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
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.

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