



Cyberscope

Audit Report

WakandaRouter

September 2022

Type BEP20

Network BSC TESTNET

Address 0xd97ECf01cd2C2f7F38999f2585483bb3A09139eF

Audited by © cyberscope

Table of Contents

Table of Contents	1
Contract Review	2
Audit Updates	2
Source Files	2
Introduction	3
Contract Diagnostics	4
L01 - Public Function could be Declared External	5
Description	5
Recommendation	5
L04 - Conformance to Solidity Naming Conventions	6
Description	6
Recommendation	6
L09 - Dead Code Elimination	7
Description	7
Recommendation	7
L14 - Uninitialized Variables in Local Scope	8
Description	8
Recommendation	8
Contract Functions	9
Contract Flow	14
Summary	15
Disclaimer	16
About Cyberscope	17

Contract Review

Contract Name	WakandaRouter
Compiler Version	v0.6.6+commit.6c089d02
Testing Deploy	https://testnet.bscscan.com/token/0xd97ECf01cd2C2f7F38999f2585483bb3A09139eF
Domain	-

Audit Updates

Initial Audit	21st September 2022
Corrected	

Source Files

Filename	SHA256
helpers/IERC20.sol	64469bf9aef280453f36bf73598078a603b95f1ef20bc0ef8256e76b216c44d8
swap/Router.sol	ecd7660cee1e38dcee45001cfc4aba1479c2b4fa3f5cd3b855f32d39aaf85b47

Introduction

The WakandaRouter contract implements a router contract. The router, which uses the library, fully supports all the basic requirements of a front-end offering trading and liquidity management functionality. Notably, it natively supports multi-pair trades (e.g. x to y to z), treats native currency as a first-class citizen, and offers meta-transactions for removing liquidity.

Contract Diagnostics

● Critical ● Medium ● Minor / Informative

Severity	Code	Description	Status
●	L01	Public Function could be Declared External	Unresolved
●	L04	Conformance to Solidity Naming Conventions	Unresolved
●	L09	Dead Code Elimination	Unresolved
●	L14	Uninitialized Variables in Local Scope	Unresolved

L01 - Public Function could be Declared External

Criticality	minor / informative
Location	src/swap/Router.sol#L1216,1226,1200,1192,1208
Status	Unresolved

Description

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

```
getAmountsOut  
getAmountsIn  
getAmountOut  
quote  
getAmountIn
```

Recommendation

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

L04 - Conformance to Solidity Naming Conventions

Criticality	minor / informative
Location	src/swap/Router.sol#L325,91,563,323,356
Status	Unresolved

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.

```
PERMIT_TYPEHASH  
WETH  
DOMAIN_SEPARATOR  
MINIMUM_LIQUIDITY
```

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 / informative
Location	src/swap/Router.sol#L36
Status	Unresolved

Description

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

```
safeApprove
```

Recommendation

Remove unused functions.

L14 - Uninitialized Variables in Local Scope

Criticality	minor / informative
Location	src/swap/Router.sol#L521,1088,888
Status	Unresolved

Description

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

```
i
```

Recommendation

All the local scoped variables should be initialized.

Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
IERC20	Interface			
	name	External		-
	symbol	External		-
	decimals	External		-
	totalSupply	External		-
	balanceOf	External		-
	allowance	External		-
	approve	External	✓	-
	transfer	External	✓	-
	transferFrom	External	✓	-
IWakandaFactory	Interface			
	feeTo	External		-
	feeToSetter	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	createPair	External	✓	-
	setFeeTo	External	✓	-
	setFeeToSetter	External	✓	-
TransferHelper	Library			
	safeApprove	Internal	✓	
	safeTransfer	Internal	✓	
	safeTransferFrom	Internal	✓	
	safeTransferETH	Internal	✓	

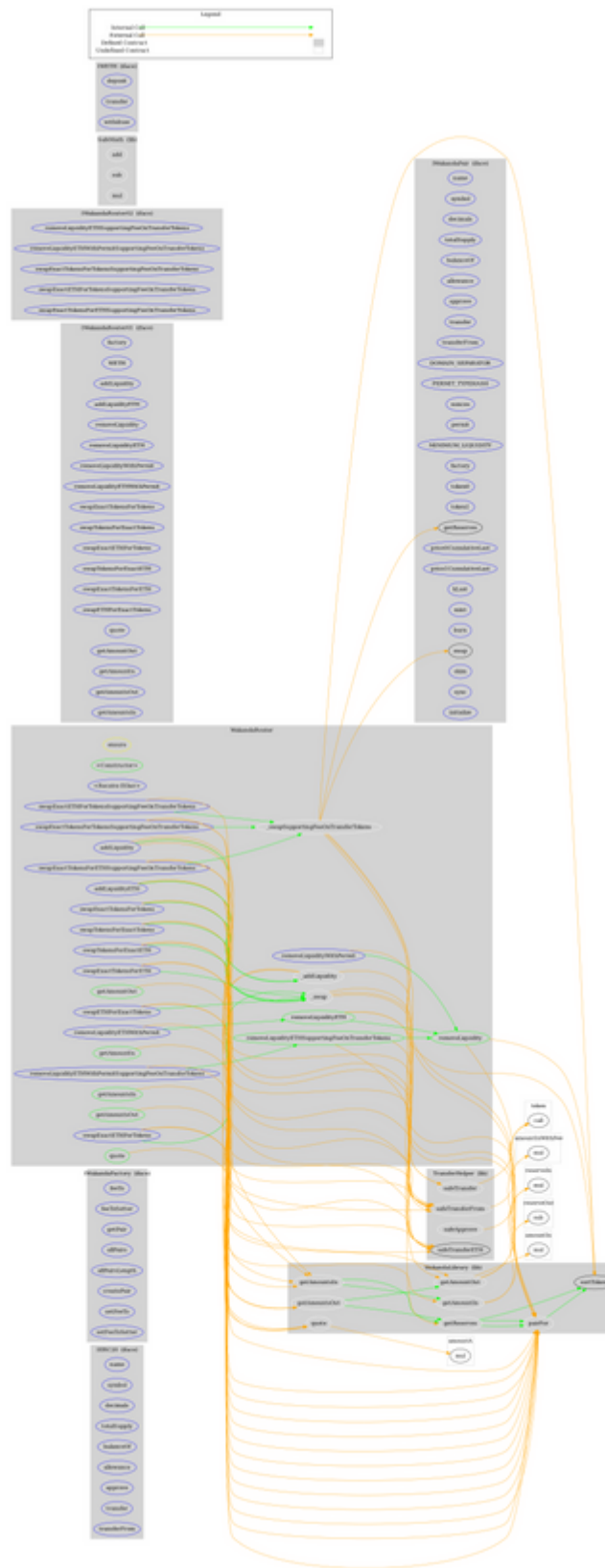
IWakandaRouter01	Interface			
	factory	External		-
	WETH	External		-
	addLiquidity	External	✓	-
	addLiquidityETH	External	Payable	-
	removeLiquidity	External	✓	-
	removeLiquidityETH	External	✓	-
	removeLiquidityWithPermit	External	✓	-
	removeLiquidityETHWithPermit	External	✓	-
	swapExactTokensForTokens	External	✓	-
	swapTokensForExactTokens	External	✓	-
	swapExactETHForTokens	External	Payable	-
	swapTokensForExactETH	External	✓	-
	swapExactTokensForETH	External	✓	-
	swapETHForExactTokens	External	Payable	-
	quote	External		-
	getAmountOut	External		-
	getAmountIn	External		-
	getAmountsOut	External		-
	getAmountsIn	External		-
IWakandaRouter02	Interface	IWakandaRouter01		
	removeLiquidityETHSupportingFeeOnTransferTokens	External	✓	-
	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External	✓	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
IWakandaPair	Interface			
	name	External		-

	symbol	External		-
	decimals	External		-
	totalSupply	External		-
	balanceOf	External		-
	allowance	External		-
	approve	External	✓	-
	transfer	External	✓	-
	transferFrom	External	✓	-
	DOMAIN_SEPARATOR	External		-
	PERMIT_TYPEHASH	External		-
	nonces	External		-
	permit	External	✓	-
	MINIMUM_LIQUIDITY	External		-
	factory	External		-
	token0	External		-
	token1	External		-
	getReserves	External		-
	price0CumulativeLast	External		-
	price1CumulativeLast	External		-
	kLast	External		-
	mint	External	✓	-
	burn	External	✓	-
	swap	External	✓	-
	skim	External	✓	-
	sync	External	✓	-
	initialize	External	✓	-
SafeMath	Library			
	add	Internal		
	sub	Internal		
	mul	Internal		
WakandaLibrary	Library			
	sortTokens	Internal		

	pairFor	Internal		
	getReserves	Internal		
	quote	Internal		
	getAmountOut	Internal		
	getAmountIn	Internal		
	getAmountsOut	Internal		
	getAmountsIn	Internal		
IWETH	Interface			
	deposit	External	Payable	-
	transfer	External	✓	-
	withdraw	External	✓	-
WakandaRouter	Implementation	IWakandaRouter02		
	<Constructor>	Public	✓	-
	<Receive Ether>	External	Payable	-
	_addLiquidity	Internal	✓	
	addLiquidity	External	✓	ensure
	addLiquidityETH	External	Payable	ensure
	removeLiquidity	Public	✓	ensure
	removeLiquidityETH	Public	✓	ensure
	removeLiquidityWithPermit	External	✓	-
	removeLiquidityETHWithPermit	External	✓	-
	removeLiquidityETHSupportingFeeOnTransferTokens	Public	✓	ensure
	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External	✓	-
	_swap	Internal	✓	
	swapExactTokensForTokens	External	✓	ensure
	swapTokensForExactTokens	External	✓	ensure
	swapExactETHForTokens	External	Payable	ensure
	swapTokensForExactETH	External	✓	ensure
	swapExactTokensForETH	External	✓	ensure
	swapETHForExactTokens	External	Payable	ensure
	_swapSupportingFeeOnTransferToken	Internal	✓	

	ns			
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	ensure
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	ensure
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	ensure
	quote	Public		-
	getAmountOut	Public		-
	getAmountIn	Public		-
	getAmountsOut	Public		-
	getAmountsIn	Public		-

Contract Flow



Summary

The smart contract analysis reported no critical or compiler issues. This audit focused on investigating security issues and potential improvements.

Disclaimer

All the content provided in this document is for general information only and should not be used as financial advice or a reason to buy any investment.

Cyberscope team provides no guarantees against the sale of team tokens or the removal of liquidity by the project audited in this document. Always Do your own research and protect yourselves from being scammed.

The Cyberscope team has audited this project for general information and only expresses their opinion based on similar projects and checks from popular diagnostic tools. Under no circumstances did Cyberscope receive a payment to manipulate those results or change the awarding badge that we will be adding in our website.

Always Do your own research and protect yourselves from scams. This document should not be presented as a reason to buy or not buy any particular token.

The Cyberscope team disclaims any liability for the resulting losses.

About Cyberscope

Coinscope audit and K.Y.C. service has been rebranded to Cyberscope.

Coinscope is the leading early coin listing, voting and auditing authority firm. The audit process is analyzing and monitoring many aspects of the project. That way, it gives the community a good sense of security using an informative report and a generic score.

Cyberscope and Coinscope are aiming to make crypto discoverable and efficient globally. They provide all the essential tools to assist users draw their own conclusions.



The Cyberscope team

<https://www.cyberscope.io>