



Cyberscope

Audit Report

Zonulet

May 2022

Type ERC20

Network Harmony

Address 0xa499ce35fc84c6ea6acb1f8864c4d9f975b306bd

Audited by © cyberscope

Table of Contents

Table of Contents	1
Contract Review	3
Source Files	3
Audit Updates	3
Contract Analysis	4
ST - Stop Transactions	5
Description	5
Recommendation	5
ELFM - Exceed Limit Fees Manipulation	6
Description	6
Recommendation	6
BC - Blacklisted Contracts	7
Description	7
Recommendation	7
Contract Diagnostics	8
L01 - Public Function could be Declared External	9
Description	9
Recommendation	9
L02 - State Variables could be Declared Constant	10
Description	10
Recommendation	10
L04 - Conformance to Solidity Naming Conventions	11
Description	11
Recommendation	11
L05 - Unused State Variable	12
Description	12

Recommendation	12
L07 - Missing Events Arithmetic	13
Description	13
Recommendation	13
L09 - Dead Code Elimination	14
Description	14
Recommendation	14
L12 - Using Variables before Declaration	15
Description	15
Recommendation	15
L14 - Uninitialized Variables in Local Scope	16
Description	16
Recommendation	16
L15 - Local Scope Variable Shadowing	17
Description	17
Recommendation	17
Contract Functions	18
Contract Flow	25
Summary	25
Disclaimer	27
About Cyberscope	28

Contract Review

Contract Name	Zonulet
Compiler Version	v0.6.12+commit.27d51765
Optimization	runs
Licence	none
Explorer	https://explorer.harmony.one/address/0xa499ce35fc84c6ea6acb1f8864c4d9f975b306bd
Symbol	ZONU
Decimals	18
Total Supply	1,000,000,000
Domain	zonulet.io

Source Files

Filename	SHA256
contract.sol	caf6dd0b56511285a7abe95fcc2fa97c977949dd1bf23cd74e8ce232da4f3dee

Audit Updates

Initial Audit	23rd May 2022
Corrected	27th May 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#L1549

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 `maxWalletTokens` to zero.

```
uint256 contractBalanceReceipient = balanceOf(to);
require(contractBalanceReceipient + amount <= maxWalletTokens,
"Exceeds maximum wallet token amount.");
}
```

Recommendation

The contract could embody a check for not allowing setting the `maxWalletTokens` less than a reasonable amount. A suggested implementation could check that the maximum amount should be more than a fixed percentage of the total supply.

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.

ELFM - Exceed Limit Fees Manipulation

Criticality	critical
Location	contract.sol#L1420

Description

The contract owner has the authority to increase over the allowed limit of 25%. The owner may take advantage of it by calling the `setMarketingFee` function with a high percentage value.

```
function setMarketingFee(uint256 value) external onlyOwner {  
    marketingFee = value;  
    totalFees = ZDEXRewardsFee.add(liquidityFee).add(marketingFee);  
}
```

Recommendation

The contract could embody a check for the maximum acceptable value.

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.

BC - Blacklisted Contracts

Criticality	medium
Location	contract.sol#L1541

Description

The contract owner has the authority to stop contracts from transactions. The owner may take advantage of it by calling the `blacklistAddress` function.

```
require(!_isBlacklisted[from] && !_isBlacklisted[to], 'Blacklisted address');
```

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.

Contract Diagnostics

● Critical ● Medium ● Minor ● Pass

Severity	Code	Description
●	L01	Public Function could be Declared External
●	L02	State Variables could be Declared Constant
●	L04	Conformance to Solidity Naming Conventions
●	L05	Unused State Variable
●	L07	Missing Events Arithmetic
●	L09	Dead Code Elimination
●	L12	Using Variables before Declaration
●	L14	Uninitialized Variables in Local Scope
●	L15	Local Scope Variable Shadowing

L01 - Public Function could be Declared External

Criticality

minor

Location`contract.sol#L404,413,665,669,676,682,774,782,799,825,833,844,862,884,903,1130,1145,1173,1187,1365,1382,1398,1427,1437,1455,1474,1478,1482,1823,1867`

Description

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

```
process
getAccountAtIndex
dividendTokenBalanceOf
withdrawableDividendOf
isExcludedFromFees
updateGasForProcessing
updateMaxWallet
setAutomatedMarketMakerPair
excludeMultipleAccountsFromFees
...
```

Recommendation

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

L02 - State Variables could be Declared Constant

Criticality

minor

Location

contract.sol#L1263,1267

Description

Constant state variables should be declared constant to save gas.

```
swapTokensAtAmount  
deadWallet
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality

minor

Location

contract.sol#L125,302,303,320,1173,1180,1187,1197,1099,1105,1265,1272,1275,1281,1778

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.

```
_account  
_marketingWalletAddress  
DEZURewardsFee  
_isBlacklisted  
DEZU  
magnitude  
_owner  
MINIMUM_LIQUIDITY  
PERMIT_TYPEHASH  
...
```

Recommendation

Follow the Solidity naming convention.

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

L05 - Unused State Variable

Criticality

minor

Location

contract.sol#L577

Description

There are segments that contain unused state variables.

```
MAX_INT256
```

Recommendation

Remove unused state variables.

L07 - Missing Events Arithmetic

Criticality

minor

Location

contract.sol#L1410,1415,1420,1437

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.

```
maxWalletTokens = maxWallet * (10 ** 18)
marketingFee = value
liquidityFee = value
DEZURewardsFee = value
```

Recommendation

Emit an event for critical parameter changes.

L09 - Dead Code Elimination

Criticality

minor

Location

contract.sol#L1207,623

Description

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

```
abs  
_transfer
```

Recommendation

Remove unused functions.

L12 - Using Variables before Declaration

Criticality

minor

Location

contract.sol#L1607

Description

The contract is using a variable before the declaration. This is usually happening either if it has not been declared yet or the variable has been declared in a different scope.

```
iterations
claims
lastProcessedIndex
```

Recommendation

The variables should be declared before any usage of them.

L14 - Uninitialized Variables in Local Scope

Criticality

minor

Location

contract.sol#L1607

Description

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

```
iterations  
lastProcessedIndex  
claims
```

Recommendation

All the local scoped variables should be initialized.

L15 - Local Scope Variable Shadowing

Criticality

minor

Location

contract.sol#L1125,1173,1180,1187,1197

Description

There are variables that are defined in the local scope containing the same name from an upper scope.

```
_owner  
_symbol  
_name
```

Recommendation

The local variables should have different names from the upper scoped variables.

Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
IUniswapV2Router01	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		-
IUniswapV2Router02	Interface	IUniswapV2Router01		
	removeLiquidityETHSupportingFeeOnTransferTokens	External	✓	-
	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External	✓	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
IUniswapV2Factory	Interface			
	feeTo	External		-
	feeToSetter	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	createPair	External	✓	-
	setFeeTo	External	✓	-
	setFeeToSetter	External	✓	-
IUniswapV2Pair	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	✓	-
IERC20Metadata	Interface	IERC20		
	name	External		-
	symbol	External		-
	decimals	External		-
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
SafeMath	Library			

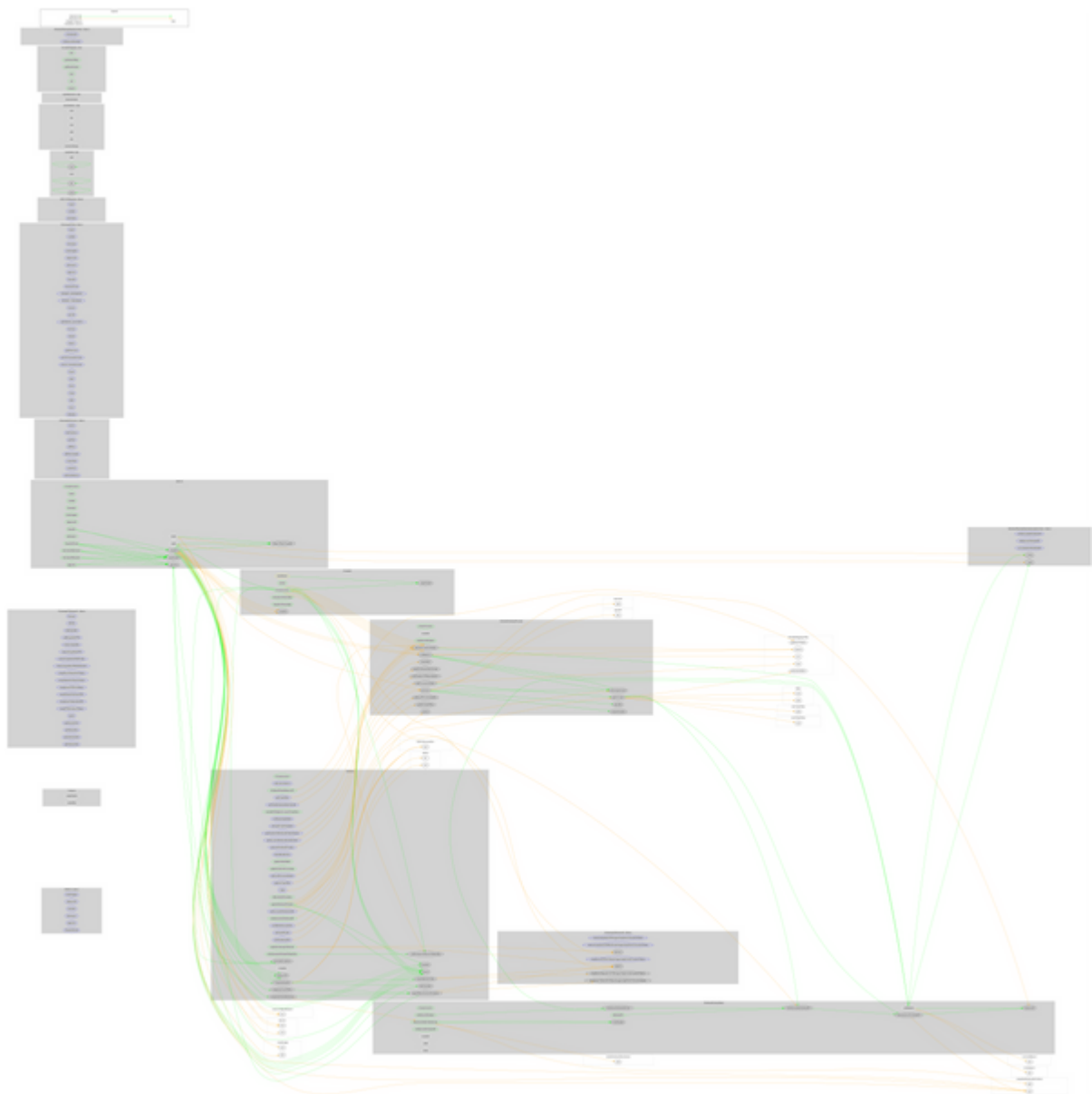
	add	Internal		
	sub	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
	mod	Internal		
	mod	Internal		
SafeMathInt	Library			
	mul	Internal		
	div	Internal		
	sub	Internal		
	add	Internal		
	abs	Internal		
	toUint256Safe	Internal		
SafeMathUint	Library			
	toInt256Safe	Internal		
IterableMapping	Library			
	get	Public		-
	getIndexOfKey	Public		-
	getKeyAtIndex	Public		-
	size	Public		-
	set	Public	✓	-
	remove	Public	✓	-
ERC20	Implementation	Context, IERC20, IERC20Metadata		
	<Constructor>	Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-

	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	_transfer	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_approve	Internal	✓	
	_beforeTokenTransfer	Internal	✓	
DividendPayingTokenInterface	Interface			
	dividendOf	External		-
	withdrawDividend	External	✓	-
DividendPayingTokenOptionalInterface	Interface			
	withdrawableDividendOf	External		-
	withdrawnDividendOf	External		-
	accumulativeDividendOf	External		-
DividendPayingToken	Implementation	ERC20, Ownable, DividendPayingTokenInterface, DividendPayingTokenOptionalInterface		
	<Constructor>	Public	✓	ERC20
	distributeDEZUDividends	Public	✓	onlyOwner
	withdrawDividend	Public	✓	-

	_withdrawDividendOfUser	Internal	✓	
	dividendOf	Public		-
	withdrawableDividendOf	Public		-
	withdrawnDividendOf	Public		-
	accumulativeDividendOf	Public		-
	_transfer	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_setBalance	Internal	✓	
ZoNulet	Implementation	ERC20, Ownable		
	<Constructor>	Public	✓	ERC20
	<Receive Ether>	External	Payable	-
	updateDividendTracker	Public	✓	onlyOwner
	updateUniswapV2Router	Public	✓	onlyOwner
	excludeFromFees	Public	✓	onlyOwner
	excludeMultipleAccountsFromFees	Public	✓	onlyOwner
	setMarketingWallet	External	✓	onlyOwner
	setDEZURewardsFee	External	✓	onlyOwner
	setLiquidityFee	External	✓	onlyOwner
	setMarketingFee	External	✓	onlyOwner
	setAutomatedMarketMakerPair	Public	✓	onlyOwner
	blacklistAddress	External	✓	onlyOwner
	updateMaxWallet	Public	✓	onlyOwner
	_setAutomatedMarketMakerPair	Private	✓	
	updateGasForProcessing	Public	✓	onlyOwner
	updateClaimWait	External	✓	onlyOwner
	getClaimWait	External		-
	getTotalDividendsDistributed	External		-
	isExcludedFromFees	Public		-
	withdrawableDividendOf	Public		-
	dividendTokenBalanceOf	Public		-
	excludeFromDividends	External	✓	onlyOwner
	getAccountDividendsInfo	External		-

	getAccountDividendsInfoAtIndex	External		-
	processDividendTracker	External	✓	-
	claim	External	✓	-
	getLastProcessedIndex	External		-
	getNumberOfDividendTokenHolders	External		-
	_transfer	Internal	✓	
	swapAndSendToFee	Private	✓	
	swapAndLiquify	Private	✓	
	swapTokensForEth	Private	✓	
	swapTokensForDexSwapZoo	Private	✓	
	addLiquidity	Private	✓	
	swapAndSendDividends	Private	✓	
ZoNuletDivide ndTracker	Implementation	Ownable, DividendPay ingToken		
	<Constructor>	Public	✓	DividendPayin gToken
	_transfer	Internal	✓	
	withdrawDividend	Public	✓	-
	excludeFromDividends	External	✓	onlyOwner
	updateClaimWait	External	✓	onlyOwner
	getLastProcessedIndex	External		-
	getNumberOfTokenHolders	External		-
	getAccount	Public		-
	getAccountAtIndex	Public		-
	canAutoClaim	Private		
	setBalance	External	✓	onlyOwner
	process	Public	✓	-
	processAccount	Public	✓	onlyOwner

Contract Flow



Summary

There are some functions that can be abused by the owner like stopping transactions for everyone except the owner, manipulating fees up to 100% and blacklisting addresses. A multi-wallet signing pattern will provide security against potential hacks. Temporarily locking the contract or renouncing ownership will eliminate all the contract threats.

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>