Cyberscope

Audit Report

ARABIANSHINJA

April 2022

Type	BEP20
Network	BSC
Address	0x7A8EC8A189ef1371FCE9639C71BbB4393a54dEC1
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	ARABIANSHINJA
Compiler Version	v0.8.10+commit.fc410830
Optimization	5 runs
Licence	None
Explorer	https://bscscan.com/token/0x7A8EC8A189ef1371FCE9639C71BbB4393a54dEC1
Symbol	$ARABIANSHINJA
Decimals	18
Total Supply	1,000,000,000,000
Domain	

Source Files

Filename	**SHA256**
contract.sol	ad6a13d5d6df977517617baf24b94fea54362cf69e2fa275c334cd01ce23651e

Audit Updates

Initial Audit	27th April 2022
Corrected phase 1	30th April 2022
Corrected phase 2	2nd 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

Contract Diagnostics

Critical Medium Minor

Severity	Code	Description
	CO	Code Optimization
	L01	Public Function could be Declared External
	L02	State Variables could be Declared Constant
	L03	Redundant Statements
	L04	Conformance to Solidity Naming Conventions
	L07	Missing Events Arithmetic
	L08	Tautology or Contradiction
	L09	Dead Code Elimination
	L12	Using Variables before Declaration
	L13	Divide before Multiply Operation
	L14	Uninitialized Variables in Local Scope

CO - Code Optimization

Criticality	minor
Location	contract.sol#L676

Description

The `setFees` method could be removed since it cannot be called successfully.

- The sum of two unsigned integers cannot be less than zero, otherwise it will underflow.

- At an unsigned integer this expression is redundant since the only way to fulfil is the `v` to be zero: `0 <= v && v <= 5`

```
function setFees(uint256 _marketingFee, uint256 _devFee, uint256 _treasuryFee,
uint256 _liquidityFee, uint256 _rewardsFee, uint256 _sellFeeIncrease) public
onlyOwner{
    require(0 <= _rewardsFee && _rewardsFee <= 5, "Requested rewardsFee fee not
within acceptable range.");
    require(0 <= _liquidityFee && _liquidityFee <= 5 , "Requested liquidity fee
not within acceptable range.");
    require(0 <= _marketingFee && _marketingFee <= 5, "Requested marketing fee
not within acceptable range.");
    require(0 <= _devFee && _devFee <= 5, "Requested marketing fee not within
acceptable range.");
    require(0 <= _devFee && _treasuryFee <= 5, "Requested marketing fee not
within acceptable range.");
    require(0 <= _sellFeeIncrease && _sellFeeIncrease <= 6, "Requested sell fee
increase not within acceptable range.");
    require(0 < _marketingFee + _liquidityFee, "Total fee amount must be
strictly positive.");

    rewardsFee = _rewardsFee;
    liquidityFee = _liquidityFee;
    marketingFee = _marketingFee;
    devFee = _devFee;
    TreasuryFee = _treasuryFee;
    sellFeeIncrease = _sellFeeIncrease;
    totalFees = _rewardsFee + _liquidityFee + _marketingFee + _treasuryFee +
_devFee;
```

```
    emit SetFees( marketingFee, TreasuryFee, devFee);
}
```

Recommendation

Rewrite some code segments so the runtime will be more performant.

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L82,96,101,126,130,134,138,146,151,155,160,166,171,299,580,587,597,613,631,648,655,676,696,706,838,921

Description

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

```
buybackStuckBNB
planBurn
setSwapTokensAtAmount
setTradeRestrictions
setFees
setDistributionCriteria
updateGasForProcessing
setAutomatedMarketMakerPair
updateUniswapV2Router
...
```

Recommendation

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

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L437,247

Description

Constant state variables should be declared constant to save gas.

```
dividendsPerShareAccuracyFactor
deadAddress
```

Recommendation

Add the constant attribute to state variables that never change.

L03 - Redundant Statements

Criticality	minor
Location	contract.sol#L875,876,877

Description

Detect the usage of redundant statements that have no effect.

```
ARABIANSHINJA
```

Recommendation

Remove redundant statements if they congest code but offer no value.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L19,82,275,223,224,233,235,580,587,655,676,696,706,797,838,843,452

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.

```
TreasuryFee
_burnAmount
_burnDenominator
_burnNumerator
_minutes
_shouldBurn
_swapTokensAtAmount
_maxWallet
_maxTx
...
```

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	contract.sol#L275,280,580

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.

```
antiBotDuration = _antiBotDuration
totalShares = totalShares - shares[shareholder].amount + amount
minPeriod = _minPeriod
```

Recommendation

Emit an event for critical parameter changes.

L08 - Tautology or Contradiction

Criticality	minor
Location	contract.sol#L676

Description

Detects expressions that are tautologies or contradictions. For instance, an uint variable will always be greater than or equal to zero.

```
require(bool,string)(0 <= _liquidityFee && _liquidityFee <= 5,Requested
liquidity fee not within acceptable range.)
require(bool,string)(0 <= _devFee && _treasuryFee <= 5,Requested marketing fee
not within acceptable range.)
require(bool,string)(0 <= _rewardsFee && _rewardsFee <= 5,Requested rewardsFee
fee not within acceptable range.)
require(bool,string)(0 <= _devFee && _devFee <= 5,Requested marketing fee not
within acceptable range.)
require(bool,string)(0 <= _marketingFee && _marketingFee <= 5,Requested
marketing fee not within acceptable range.)
require(bool,string)(0 <= _sellFeeIncrease && _sellFeeIncrease <= 6,Requested
sell fee increase not within acceptable range.)
```

Recommendation

Fix the incorrect comparison by changing the value type or the comparison.

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L195,213

Description

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

```
_setupDecimals
_burn
```

Recommendation

Remove unused functions.

L12 - Using Variables before Declaration

Criticality	minor
Location	contract.sol#L760

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.

```
fee
burnFees
```

Recommendation

The variables should be declared before any usage of them.

L13 - Divide before Multiply Operation

Criticality	minor
Location	contract.sol#L856,428

Description

Performing divisions before multiplications may cause lose of prediction.

```
maxWallet = 20 * TOTAL_SUPPLY / 1000 * (DECIMALS)
swapTokensAtAmount = 0 * TOTAL_SUPPLY / 1e4 * (DECIMALS)
maxTx = 5 * TOTAL_SUPPLY / 1000 * (DECIMALS)
treasurytokens = tokens * TreasuryFee / totalFees
devtokens = tokens * devFee / totalFees
marketingtokens = tokens * marketingFee / totalFees
halfLPTokens = LPtokens / 2
```

Recommendation

The multiplications should be prior to the divisions.

L14 - Uninitialized Variables in Local Scope

Criticality	minor
Location	contract.sol#L766

Description

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

```
fee_scope_0
burnFees_scope_1
```

Recommendation

All the local scoped variables should be initialized.

Contract Functions

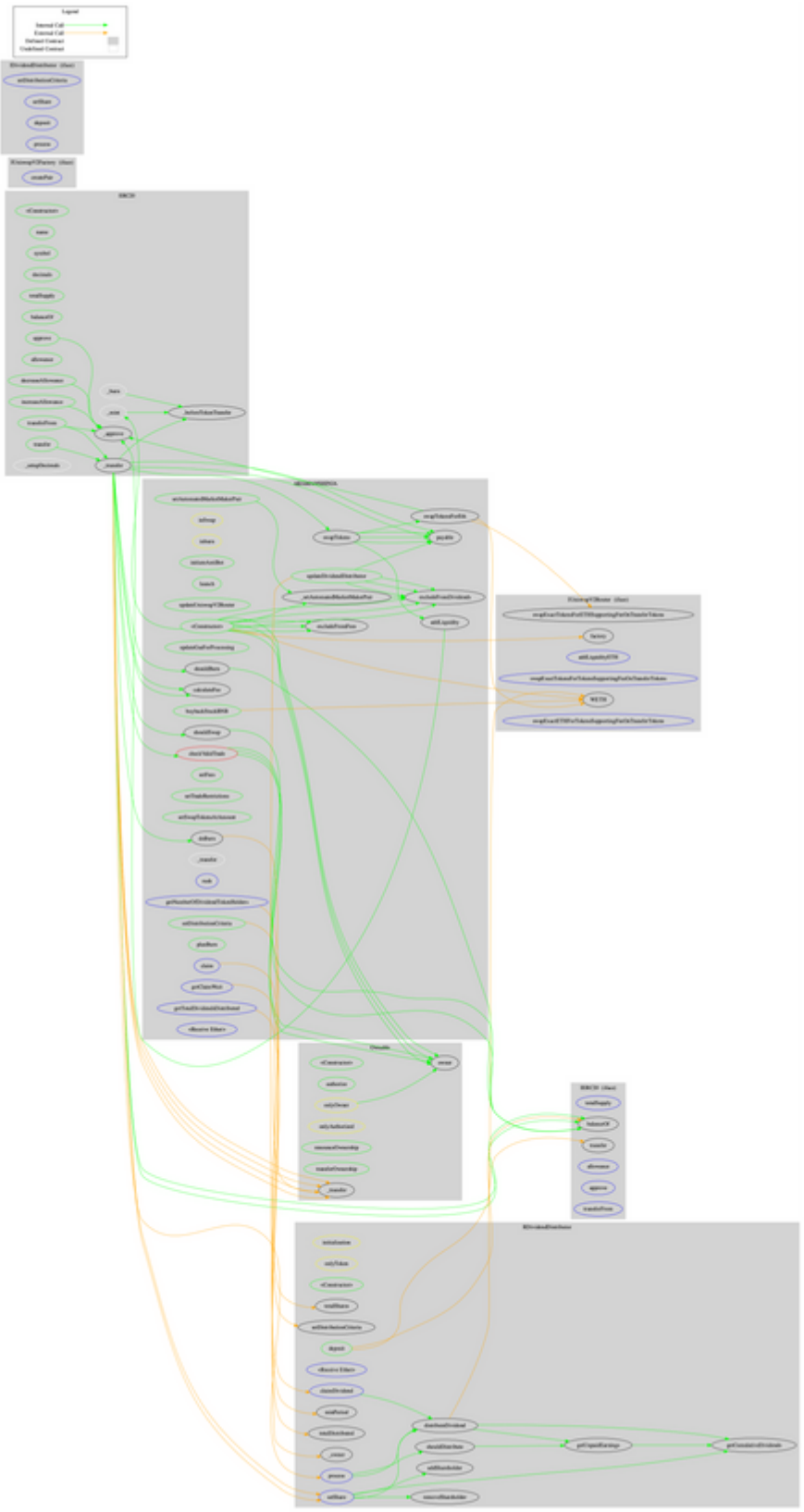
Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
IUniswapV2Router	Interface			
	factory	External		-
	WETH	External		-
	addLiquidityETH	External	Payable	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
IUniswapV2Factory	Interface			
	createPair	External	✓	-
IDividendDistributor	Interface			
	setDistributionCriteria	External	✓	-
	setShare	External	✓	-
	deposit	External	Payable	-
	process	External	✓	-

Ownable	Implementation			
	<Constructor>	Public	✓	-
	owner	Public		-
	authorize	Public	✓	onlyOwner
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
ERC20	Implementation	IERC20		
	<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	✓	
	_setupDecimals	Internal	✓	
	_beforeTokenTransfer	Internal	✓	
RDividendDistributor	Implementation	IDividendDistributor		
	<Constructor>	Public	✓	-
	setDistributionCriteria	External	✓	onlyToken
	setShare	External	✓	onlyToken
	deposit	Public	Payable	-
	<Receive Ether>	External	Payable	-
	process	External	✓	-
	shouldDistribute	Internal		

	distributeDividend	Internal	✓	
	claimDividend	External	✓	-
	getUnpaidEarnings	Public		-
	getCumulativeDividends	Internal		
	addShareholder	Internal	✓	
	removeShareholder	Internal	✓	
ARABIANSHINJA	Implementation	ERC20, Ownable		
	<Constructor>	Public	✓	ERC20
	initiateAntiBot	Public	✓	onlyOwner
	launch	Public	✓	onlyOwner
	updateDividendDistributor	Public	✓	onlyOwner
	updateUniswapV2Router	Public	✓	onlyOwner
	excludeFromFees	Public	✓	onlyOwner
	excludeFromDividends	Public	✓	onlyOwner
	setAutomatedMarketMakerPair	Public	✓	onlyOwner
	_setAutomatedMarketMakerPair	Private	✓	
	updateGasForProcessing	Public	✓	onlyOwner
	setDistributionCriteria	Public	✓	onlyOwner
	getClaimWait	External		-
	getTotalDividendsDistributed	External		-
	claim	External	✓	-
	getNumberOfDividendTokenHolders	External		-
	setFees	Public	✓	onlyOwner
	setTradeRestrictions	Public	✓	onlyOwner
	setSwapTokensAtAmount	Public	✓	onlyOwner
	checkValidTrade	Private		
	_transfer	Internal	✓	
	rush	External	✓	onlyAuthorized
	calculateFee	Private	✓	
	shouldBurn	Private		
	planBurn	Public	✓	onlyAuthorized
	doBurn	Private	✓	inburn
	shouldSwap	Private		
	swapTokens	Private	✓	inSwap

	swapTokensForEth	Private	✓	
	addLiquidity	Private	✓	
	buybackStuckBNB	Public	✓	onlyAuthorized
	<Receive Ether>	External	Payable	-

Contract Flow



Domain Info

Domain Name	arabianshibnobi.com
Registry Domain ID	2691286184_DOMAIN_COM-VRSN
Creation Date	2022-04-23T07:15:44Z
Updated Date	2022-04-23T07:15:46Z
Registry Expiry Date	2023-04-23T07:15:44Z
Registrar WHOIS Server	whois.publicdomainregistry.com
Registrar URL	www.publicdomainregistry.com
Registrar	PDR Ltd. d/b/a PublicDomainRegistry.com
Registrar IANA ID	303

The domain has been created 4 days 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

ARABIANSHINJA is an interesting project that has a friendly and growing community. 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. The fees are fixed to 6% for buys and 10% for sales.

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