



Smart Contract Audit

FOR

Elon Floki CEO

DATED : 8 May 23'



AUDIT SUMMARY

Project name - Elon Floki CEO

Date: 8 May, 2023

Scope of Audit- Audit Ace was consulted to conduct the smart contract audit of the solidity source codes.

Audit Status: Passed

Issues Found

Status	Critical	High	Medium	Low	Suggestion
Open	0	1	2	0	0
Acknowledged	0	0	0	0	0
Resolved	0	0	0	0	0



USED TOOLS

Tools:

1- Manual Review:

A line by line code review has been performed by audit ace team.

2- BSC Test Network: All tests were conducted on the BSC Test network, and each test has a corresponding transaction attached to it. These tests can be found in the "Functional Tests" section of the report.

3- Slither :

The code has undergone static analysis using Slither.

Testnet version:

The tests were performed using the contract deployed on the BSC Testnet, which can be found at the following address:

<https://testnet.bscscan.com/address/0xE4C789A7c82B0B912200F85962a87cee533C4F3>



Token Information

Token Name: Elon Floki CEO

Token Symbol: ElonFloki

Decimals: 9

Token Supply: 1,000,000,000

Token Address:

0xD020E5040085f86989c3f216Cc8258D36317ff11

Checksum:

80b4b14b6f2ec91de8765d5ba8fe52cf73411863

Owner:

0x655776fC999B3Ece67d6124952FEdB598481A35d

(at time of writing the audit)

Deployer:

0x655776fC999B3Ece67d6124952FEdB598481A35d



TOKEN OVERVIEW

Fees:

Buy Fees: 0% current and can change Upto 20%

Sell Fees: 10% current and change upto 20%

Transfer Fees: 10% current and change upto 20%

Fees Privilege: Owner

Ownership: Owned

Minting: No mint function

Max Tx Amount/ Max Wallet Amount: No

Blacklist: No

Other Privileges: changing buy/sell/transfer fees



AUDIT METHODOLOGY

The auditing process will follow a routine as special considerations by Auditace:

- Review of the specifications, sources, and instructions provided to Auditace to make sure the contract logic meets the intentions of the client without exposing the user's funds to risk.
- Manual review of the entire codebase by our experts, which is the process of reading source code line-by-line in an attempt to identify potential vulnerabilities.
- Specification comparison is the process of checking whether the code does what the specifications, sources, and instructions provided to Auditace describe.
- Test coverage analysis determines whether the test cases are covering the code and how much code is exercised when we run the test cases.
- Symbolic execution is analysing a program to determine what inputs cause each part of a program to execute.
- Reviewing the codebase to improve maintainability, security, and control based on the established industry and academic practices.

VULNERABILITY CHECKLIST



Return values of low-level calls



Gasless Send



Private modifier



Using block.timestamp



Multiple Sends



Re-entrancy



Using Suicide



Tautology or contradiction



Gas Limit and Loops



Timestamp Dependence



Address hardcoded



Revert/require functions



Exception Disorder



Use of tx.origin



Using inline assembly



Integer overflow/underflow



Divide before multiply



Dangerous strict equalities



Missing Zero Address Validation



Using SHA3



Compiler version not fixed



Using throw



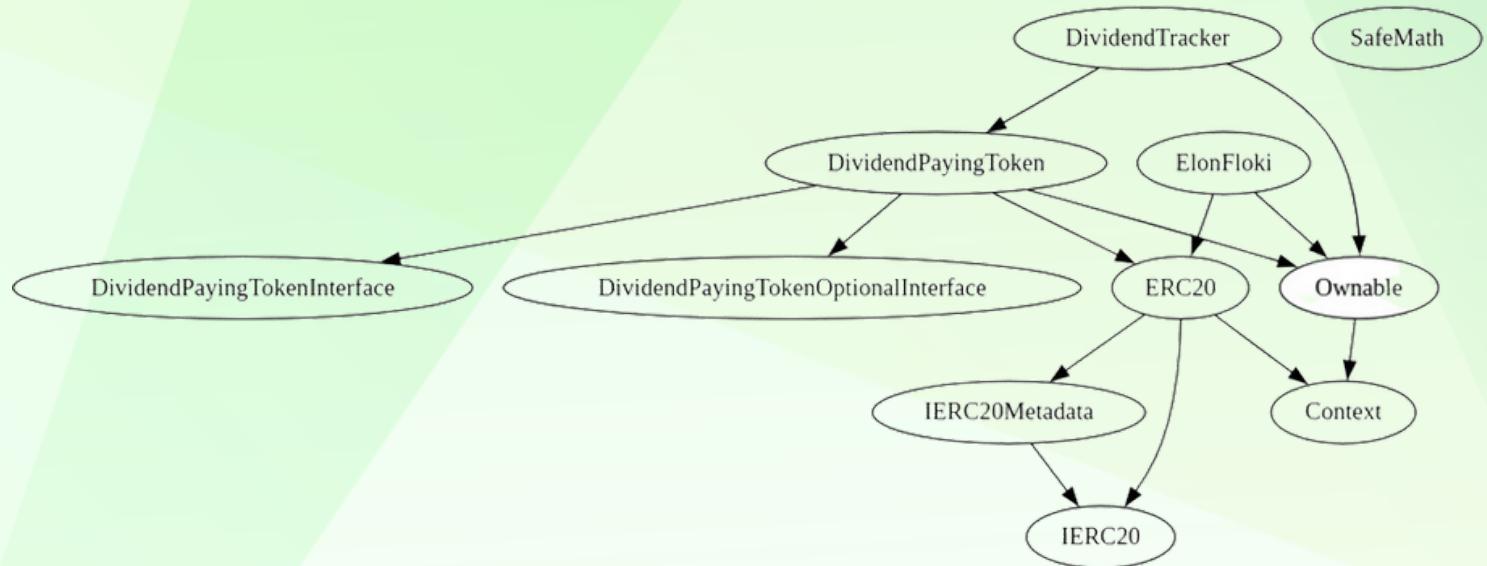
CLASSIFICATION OF RISK

Severity	Description
◆ Critical	These vulnerabilities could be exploited easily and can lead to asset loss, data loss, asset, or data manipulation. They should be fixed right away.
◆ High-Risk	A vulnerability that affects the desired outcome when using a contract, or provides the opportunity to use a contract in an unintended way.
◆ Medium-Risk	A vulnerability that could affect the desired outcome of executing the contract in a specific scenario.
◆ Low-Risk	A vulnerability that does not have a significant impact on possible scenarios for the use of the contract and is probably subjective.
◆ Gas Optimization / Suggestion	A vulnerability that has an informational character but is not affecting any of the code.

Findings

Severity	Found
◆ Critical	0
◆ High-Risk	1
◆ Medium-Risk	2
◆ Low-Risk	0
◆ Gas Optimization / Suggestions	0

INHERITANCE TREE





POINTS TO NOTE

- Owner is able to set buy/sell/transfer tax each one up to 20%
- Owner is not able to set a max buy/transfer/wallet/sell amount
- Owner is not able to blacklist an arbitrary wallet
- Owner is not able to disable trades
- Owner is not able to mint new tokens
- **Owner must enable trades for holders to be able to trade**



CONTRACT ASSESSMENT

Contract	Type	Bases			
			Function Name	**Visibility**	**Mutability**
					Modifiers
			Context	Implementation	
	L _msgSender	Internal	🔒		
	L _msgData	Internal	🔒		
			Ownable	Implementation	Context
	L <Constructor>	Public	!	NO	!
	L owner	Public	!	NO	!
	L renounceOwnership	Public	!	!	onlyOwner
	L transferOwnership	Public	!	!	onlyOwner
	L _transferOwnership	Internal	🔒	!	
			SafeMath	Library	
	L add	Internal	🔒		
	L sub	Internal	🔒		
	L sub	Internal	🔒		
	L mul	Internal	🔒		
	L div	Internal	🔒		
	L div	Internal	🔒		
	L mod	Internal	🔒		
	L mod	Internal	🔒		
			SafeMathInt	Library	
	L mul	Internal	🔒		
	L div	Internal	🔒		
	L sub	Internal	🔒		
	L add	Internal	🔒		
	L abs	Internal	🔒		
	L toUint256Safe	Internal	🔒		
			SafeMathUint	Library	
	L toInt256Safe	Internal	🔒		
			IterableMapping	Library	
	L get	Internal	🔒		
	L getIndexOfKey	Internal	🔒		
	L getKeyAtIndex	Internal	🔒		
	L size	Internal	🔒		
	L set	Internal	🔒	!	
	L remove	Internal	🔒	!	

CONTRACT ASSESSMENT

```
| **IUniswapV2Factory** | Interface | ||| |
| L | feeTo | External ! | |NO ! | |
| L | feeToSetter | External ! | |NO ! | |
| L | getPair | External ! | |NO ! | |
| L | allPairs | External ! | |NO ! | |
| L | allPairsLength | External ! | |NO ! | |
| L | createPair | External ! | ● |NO ! | |
| L | setFeeTo | External ! | ● |NO ! | |
| L | setFeeToSetter | External ! | ● |NO ! | |
|||||
| **IUniswapV2Pair** | Interface | |||
| L | name | External ! | |NO ! | |
| L | symbol | External ! | |NO ! | |
| L | decimals | External ! | |NO ! | |
| L | totalSupply | External ! | |NO ! | |
| L | balanceOf | External ! | |NO ! | |
| L | allowance | External ! | |NO ! | |
| L | approve | External ! | ● |NO ! | |
| L | transfer | External ! | ● |NO ! | |
| L | transferFrom | External ! | ● |NO ! | |
| L | DOMAIN_SEPARATOR | External ! | |NO ! | |
| L | PERMIT_TYPEHASH | External ! | |NO ! | |
| L | nonces | External ! | |NO ! | |
| L | permit | External ! | ● |NO ! | |
| L | MINIMUM_LIQUIDITY | External ! | |NO ! | |
| L | factory | External ! | |NO ! | |
| L | token0 | External ! | |NO ! | |
| L | token1 | External ! | |NO ! | |
| L | getReserves | External ! | |NO ! | |
| L | price0CumulativeLast | External ! | |NO ! | |
| L | price1CumulativeLast | External ! | |NO ! | |
| L | kLast | External ! | |NO ! | |
| L | mint | External ! | ● |NO ! | |
| L | burn | External ! | ● |NO ! | |
| L | swap | External ! | ● |NO ! | |
| L | skim | External ! | ● |NO ! | |
| L | sync | External ! | ● |NO ! | |
| L | initialize | External ! | ● |NO ! | |
|||||
| **IUniswapV2Router01** | Interface | |||
| L | factory | External ! | |NO ! |
```



CONTRACT ASSESSMENT

L WETH External ! NO !
L addLiquidity External ! ● NO !
L addLiquidityETH External ! S NO !
L removeLiquidity External ! ● NO !
L removeLiquidityETH External ! ● NO !
L removeLiquidityWithPermit External ! ● NO !
L removeLiquidityETHWithPermit External ! ● NO !
L swapExactTokensForTokens External ! ● NO !
L swapTokensForExactTokens External ! ● NO !
L swapExactETHForTokens External ! S NO !
L swapTokensForExactETH External ! ● NO !
L swapExactTokensForETH External ! ● NO !
L swapETHForExactTokens External ! S NO !
L quote External ! NO !
L getAmountOut External ! NO !
L getAmountIn External ! NO !
L getAmountsOut External ! NO !
L getAmountsIn External ! NO !
IUniswapV2Router02 Interface IUniswapV2Router01
L removeLiquidityETHSupportingFeeOnTransferTokens External ! ● NO !
L removeLiquidityETHWithPermitSupportingFeeOnTransferTokens External ! ● NO !
L swapExactTokensForTokensSupportingFeeOnTransferTokens External ! ● NO !
L swapExactETHForTokensSupportingFeeOnTransferTokens External ! S NO !
L swapExactTokensForETHSupportingFeeOnTransferTokens External ! ● NO !
IERC20 Interface
L totalSupply External ! NO !
L balanceOf External ! NO !
L allowance External ! NO !
L transfer External ! ● NO !
L approve External ! ● NO !
L transferFrom External ! ● NO !
IERC20Metadata Interface IERC20
L name External ! NO !
L symbol External ! NO !
L decimals External ! NO !
ERC20 Implementation Context, IERC20, IERC20Metadata
L <Constructor> Public ! ● NO !
L name Public ! NO !
L symbol Public ! NO !



CONTRACT ASSESSMENT

```
| L | decimals | Public ! | NO ! | | |
| L | totalSupply | Public ! | NO ! |
| L | balanceOf | Public ! | NO ! |
| L | transfer | Public ! | ● | NO ! |
| L | allowance | Public ! | NO ! |
| L | approve | Public ! | ● | NO ! |
| L | transferFrom | Public ! | ● | NO ! |
| L | increaseAllowance | Public ! | ● | NO ! |
| L | decreaseAllowance | Public ! | ● | NO ! |
| L | _transfer | Internal 🔒 | ● |||
| L | _mint | Internal 🔒 | ● |||
| L | _burn | Internal 🔒 | ● |||
| L | _approve | Internal 🔒 | ● |||
| L | _beforeTokenTransfer | Internal 🔒 | ● |||
||||
| **DividendPayingTokenInterface** | Interface | ||
| L | dividendOf | External ! | NO ! |
| L | withdrawDividend | External ! | ● | NO ! |
||||
| **DividendPayingTokenOptionalInterface** | Interface | ||
| L | withdrawableDividendOf | External ! | NO ! |
| L | withdrawnDividendOf | External ! | NO ! |
| L | accumulativeDividendOf | External ! | NO ! |
||||
| **DividendPayingToken** | Implementation | ERC20, Ownable, DividendPayingTokenInterface, DividendPayingTokenOptionalInterface ||
| L | <Constructor> | Public ! | ● | ERC20 |
| L | distributeDividends | Public ! | ● | onlyOwner |
| L | withdrawDividend | Public ! | ● | NO ! |
| L | _withdrawDividendOfUser | Internal 🔒 | ● |||
| L | dividendOf | Public ! | NO ! |
| L | withdrawableDividendOf | Public ! | NO ! |
| L | withdrawnDividendOf | Public ! | NO ! |
| L | accumulativeDividendOf | Public ! | NO ! |
| L | _transfer | Internal 🔒 | ● |||
| L | _mint | Internal 🔒 | ● |||
| L | _burn | Internal 🔒 | ● |||
| L | _setBalance | Internal 🔒 | ● |||
||||
| **DividendTracker** | Implementation | Ownable, DividendPayingToken ||
| L | <Constructor> | Public ! | ● | DividendPayingToken |
| L | _transfer | Internal 🔒 | |||
| L | withdrawDividend | Public ! | NO ! |
```



CONTRACT ASSESSMENT

```
| L | updateMinimumTokenBalanceForDividends | External ! | ● | onlyOwner | |
| L | excludeFromDividends | External ! | ● | onlyOwner |
| L | updateClaimWait | External ! | ● | onlyOwner |
| L | setLastProcessedIndex | External ! | ● | onlyOwner |
| L | getLastProcessedIndex | External ! | | NO ! |
| L | getNumberOfTokenHolders | External ! | | NO ! |
| L | getAccount | Public ! | | NO ! |
| L | getAccountAtIndex | Public ! | | NO ! |
| L | canAutoClaim | Private 🔒 | |||
| L | setBalance | External ! | ● | onlyOwner |
| L | process | Public ! | ● | NO ! |
| L | processAccount | Public ! | ● | onlyOwner |
|||||
| **ElonFloki** | Implementation | ERC20, Ownable ||
| L | <Constructor> | Public ! | 💸 | ERC20 |
| L | <Receive Ether> | External ! | 💸 | NO ! |
| L | claimStuckTokens | External ! | ● | onlyOwner |
| L | isContract | Internal 🔒 | |||
| L | sendBNB | Internal 🔒 | ● | |
| L | _setAutomatedMarketMakerPair | Private 🔒 | ● | |
| L | excludeFromFees | External ! | ● | onlyOwner |
| L | isExcludedFromFees | Public ! | | NO ! |
| L | updateBuyFees | External ! | ● | onlyOwner |
| L | updateSellFees | External ! | ● | onlyOwner |
| L | changeMarketingWallet | External ! | ● | onlyOwner |
| L | enableTrading | External ! | ● | onlyOwner |
| L | _transfer | Internal 🔒 | ● | |
| L | swapAndLiquify | Private 🔒 | ● | |
| L | swapAndSendDividends | Private 🔒 | ● | |
| L | setSwapTokensAtAmount | External ! | ● | onlyOwner |
| L | setSwapEnabled | External ! | ● | onlyOwner |
| L | updateGasForProcessing | Public ! | ● | onlyOwner |
| L | updateMinimumBalanceForDividends | External ! | ● | onlyOwner |
| L | updateClaimWait | External ! | ● | onlyOwner |
| L | getClaimWait | External ! | | NO ! |
| L | getTotalDividendsDistributed | External ! | | NO ! |
| L | withdrawableDividendOf | Public ! | | NO ! |
| L | dividendTokenBalanceOf | Public ! | | NO ! |
| L | totalRewardsEarned | Public ! | | NO ! |
| L | excludeFromDividends | External ! | ● | onlyOwner |
| L | getAccountDividendsInfo | External ! | | NO ! |
| L | getAccountDividendsInfoAtIndex | External ! | | NO ! |
```

CONTRACT ASSESSMENT

	L	processDividendTracker	External	!		●	NO	!	
	L	claim	External	!		●	NO	!	
	L	claimAddress	External	!		●	onlyOwner		
	L	getLastProcessedIndex	External	!		!	NO	!	
	L	setLastProcessedIndex	External	!		●	onlyOwner		
	L	getNumberOfDividendTokenHolders	External	!		!	NO	!	

Legend

	Symbol	Meaning	
:	----- -----		
	●	Function can modify state	
	\$	Function is payable	



STATIC ANALYSIS

```
Pragma version^0.8.17 (contracts/Token.sol#27) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.16
solc-0.8.19 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in ElonFloki.sendBNB(address,uint256) (contracts/Token.sol#1348-1359):
- (success) = recipient.call{value: amount}() (contracts/Token.sol#1357)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function IUniswapV2Pair.DOMAIN_SEPARATOR() (contracts/Token.sol#319) is not in mixedCase
Function IUniswapV2Pair.PERMIT_TYPEHASH() (contracts/Token.sol#321) is not in mixedCase
Function IUniswapV2Pair.MINIMUM_LIQUIDITY() (contracts/Token.sol#352) is not in mixedCase
Function IUniswapV2Router01.WETH() (contracts/Token.sol#392) is not in mixedCase
Parameter DividendPayingToken.dividendOf(address).owner (contracts/Token.sol#871) is not in mixedCase
Parameter DividendPayingToken.withdrawableDividendOf(address).owner (contracts/Token.sol#876) is not in mixedCase
Parameter DividendPayingToken.withdrawnDividendOf(address).owner (contracts/Token.sol#882) is not in mixedCase
Parameter DividendPayingToken.accumulativeDividendOf(address).owner (contracts/Token.sol#888) is not in mixedCase
Constant DividendPayingToken.magnitude (contracts/Token.sol#811) is not in UPPER_CASE_WITH_UNDERSCORES
Parameter DividendTracker.updateMinimumTokenBalanceForDividends(uint256).newMinimumBalance (contracts/Token.sol#987) is not in mixedCase
Parameter DividendTracker.getAccount(address).account (contracts/Token.sol#1032) is not in mixedCase
Parameter ElonFloki.updateBuyFees(uint256,uint256,uint256,uint256).liquidityFeeOnBuy (contracts/Token.sol#1393) is not in mixedCase
Parameter ElonFloki.updateBuyFees(uint256,uint256,uint256,uint256).marketingFeeOnBuy (contracts/Token.sol#1394) is not in mixedCase
Parameter ElonFloki.updateBuyFees(uint256,uint256,uint256,uint256).rewardsFeeOnBuy (contracts/Token.sol#1395) is not in mixedCase
Parameter ElonFloki.updateBuyFees(uint256,uint256,uint256,uint256).trueBurnFeeOnBuy (contracts/Token.sol#1396) is not in mixedCase
Parameter ElonFloki.updateSellFees(uint256,uint256,uint256,uint256).liquidityFeeOnSell (contracts/Token.sol#1415) is not in mixedCase
Parameter ElonFloki.updateSellFees(uint256,uint256,uint256,uint256).marketingFeeOnSell (contracts/Token.sol#1416) is not in mixedCase
Parameter ElonFloki.updateSellFees(uint256,uint256,uint256,uint256).rewardsFeeOnSell (contracts/Token.sol#1417) is not in mixedCase
Parameter ElonFloki.updateSellFees(uint256,uint256,uint256,uint256).trueBurnFeeOnSell (contracts/Token.sol#1418) is not in mixedCase
Parameter ElonFloki.changeMarketingWallet(address).marketingWallet (contracts/Token.sol#1437) is not in mixedCase
Parameter ElonFloki.setSwapEnabled(bool).enabled (contracts/Token.sol#1668) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Variable IUniswapV2Router01.addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256).amountADesired (contracts/Token.sol#397) is too similar to IUniswapV2Router01.addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256).amountBDesired (contracts/Token.sol#398)
Variable DividendPayingToken._withdrawDividendOfUser(address)._withdrawableDividend (contracts/Token.sol#848) is too similar to DividendTracker.getAccount(address).withdrawableDividends (contracts/Token.sol#1040)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-too-similar

SafeMathInt.MAX_INT256 (contracts/Token.sol#145) is never used in SafeMathInt (contracts/Token.sol#143-185)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

ElonFloki.dividendTracker (contracts/Token.sol#1234) should be immutable
ElonFloki.uniswapV2Pair (contracts/Token.sol#1224) should be immutable
ElonFloki.uniswapV2Router (contracts/Token.sol#1223) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
```

**Result => A static analysis of contract's source code has been performed using slither,
No major issues were found in the output**



FUNCTIONAL TESTING

1- Adding liquidity (**passed**):

<https://testnet.bscscan.com/tx/0x9a628b556ede369bf896f8ac79e1b0c0b073d7c0a45c6513df8f7bb0d4f12d73>

2- Buying when excluded (0% tax) (**passed**):

<https://testnet.bscscan.com/tx/0xddac207534690fcfb10ae6baebdadb18f1ad9199195e5ac6a3a6f603bf3d8b6>

3- Selling when excluded (0% tax) (**passed**):

<https://testnet.bscscan.com/tx/0x39e87570e7b1c48a963e3264d4ee071539bad157bc4a3bac9e31bae50a261d99>

4- Transferring when excluded from fees (0% tax) (**passed**):

<https://testnet.bscscan.com/tx/0x1e90a233944c4dddf7335f2ef6f288aa2f149d77c73af0fc860929c19f6e4c1f>

5- Buying when not excluded from fees (up to 20% tax) (**passed**):

<https://testnet.bscscan.com/tx/0x24561cb8094f43c34d4be5a0d97a0cd396d5f9e35683702bf8b0503f243a1191>

6- Selling when not excluded from fees (up to 20% tax) (**passed**):

<https://testnet.bscscan.com/tx/0xfaafba25be1a1d27d36bfe0e06b63433440ba751c94873962c96677a87df7804>



FUNCTIONAL TESTING

7- Transferring when not excluded from fees (up to 20% tax) (passed):

<https://testnet.bscscan.com/tx/0xd9f2a870acb93d1c59729eede55a940108a014f73947979fb63c1f570c11035e>

7- Internal swap (rewards, marketing fee, liquidity and burn) (passed):

<https://testnet.bscscan.com/tx/0xd9f2a870acb93d1c59729eede55a940108a014f73947979fb63c1f570c11035e>



MANUAL TESTING

Centralization – Trades must be enabled

Severity: High

function: enableTrading

Status: Not Resolved

Overview:

The smart contract owner must enable trades for holders. If trading remain disabled, no one would be able to buy/sell/transfer tokens.

```
function enableTrading() external onlyOwner {  
    require(!isTradeEnabled, "Trading already enabled");  
    isTradeEnabled = true;  
}
```

Suggestion

To mitigate this centralization issue, we propose the following options:

1. Renounce Ownership: Consider relinquishing control of the smart contract by renouncing ownership. This would remove the ability for a single entity to manipulate the router, reducing centralization risks.
2. Multi-signature Wallet: Transfer ownership to a multi-signature wallet. This would require multiple approvals for any changes to the mainRouter, adding an additional layer of security and reducing the centralization risk.
3. Transfer ownership to a trusted and valid 3rd party in order to guarantee enabling of the trades (**applied**)



MANUAL TESTING

Centralization - EOA receiving LP tokens

Severity: Medium

Function: addLiquidity

Lines: 1629

Status: Not Resolved

Overview:

LP tokens generated from auto-liquidity are sent to owner's wallet. This LP tokens can be used to remove a portion of liquidity pool (BNB and Tokens)

```
uniswapV2Router.addLiquidityETH{value: newBalance}(  
    address(this),  
    otherHalf,  
    0, // slippage is unavoidable  
    0, // slippage is unavoidable  
    address(0x2e6312DA7541e67bfa5d945aD200043A6e28fB68),  
    block.timestamp  
>);
```

Recommendation:

- Burn LP tokens
- or
- Lock LP tokens



MANUAL TESTING

Centralization - Excessive tax

Severity: Medium

Function: updateBuyFees and updateSellFees

Status: Not Resolved

Overview:

Owner is able to set buy/sell/transfer tax each one up to 20%

```
function updateBuyFees(
    uint256 _liquidityFeeOnBuy,
    uint256 _marketingFeeOnBuy,
    uint256 _rewardsFeeOnBuy,
    uint256 _trueBurnFeeOnBuy
) external onlyOwner {
    liquidityFeeOnBuy = _liquidityFeeOnBuy;
    marketingFeeOnBuy = _marketingFeeOnBuy;
    rewardsFeeOnBuy = _rewardsFeeOnBuy;
    trueBurnFeeOnBuy = _trueBurnFeeOnBuy;

    totalBuyFee =
        liquidityFeeOnBuy +
        marketingFeeOnBuy +
        rewardsFeeOnBuy +
        trueBurnFeeOnBuy;

    require(totalBuyFee <= 20, "Buy fee cannot be more than 20%");

    emit BuyFeesUpdated(totalBuyFee);
}
```

Recommendation:

According to pinksale safu criteria, sum of buy+sell tax should not exceed 25% percent, so its suggested to ensure that sum of both taxes (totalBuyFee and totalSellfee) are always less than 25%



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



ABOUT AUDITACE

We specialize in providing thorough and reliable audits for Web3 projects. With a team of experienced professionals, we use cutting-edge technology and rigorous methodologies to evaluate the security and integrity of blockchain systems. We are committed to helping our clients ensure the safety and transparency of their digital assets and transactions.



<https://auditace.tech/>



https://t.me/Audit_Ace



https://twitter.com/auditace_



<https://github.com/Audit-Ace>
