



Smart Contract Audit

FOR

Sheikh Pepe

DATED : 12 May 23'



AUDIT SUMMARY

Project name - Sheikh Pepe

Date: 12 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	0	1	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/token/0x5d4d29a8a44e2dAEA4a435e6eAf536D6841D1ea5>



Token Information

Token Name: Sheikh Pepe

Token Symbol: SPepe

Decimals: 9

Token Supply: 1,000,000,000

Token Address:

0xE039b8A4C01B6772A4AE159dc657D33c651BC4c4

Checksum:

8cf9bbcf9932d8a3b45a6885f48e462fcf049a4f

Owner:

0x8cFFcaab079b637c9942C12486e8A8Af65b80F68

(at time of writing the audit)

Deployer:

0xDDFB65eE7999f64304Af6dDCdE4ebECfbe8d6232



TOKEN OVERVIEW

Fees:

Buy Fees: up to 10%

Sell Fees: up to 10%

Transfer Fees: 0%

Fees Privilege: Owner

Ownership: Owned

Minting: No mint function

Max Tx Amount/ Max Wallet Amount: No

Blacklist: No

Other Privileges: changing swap threshold - enabling trades - changing 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



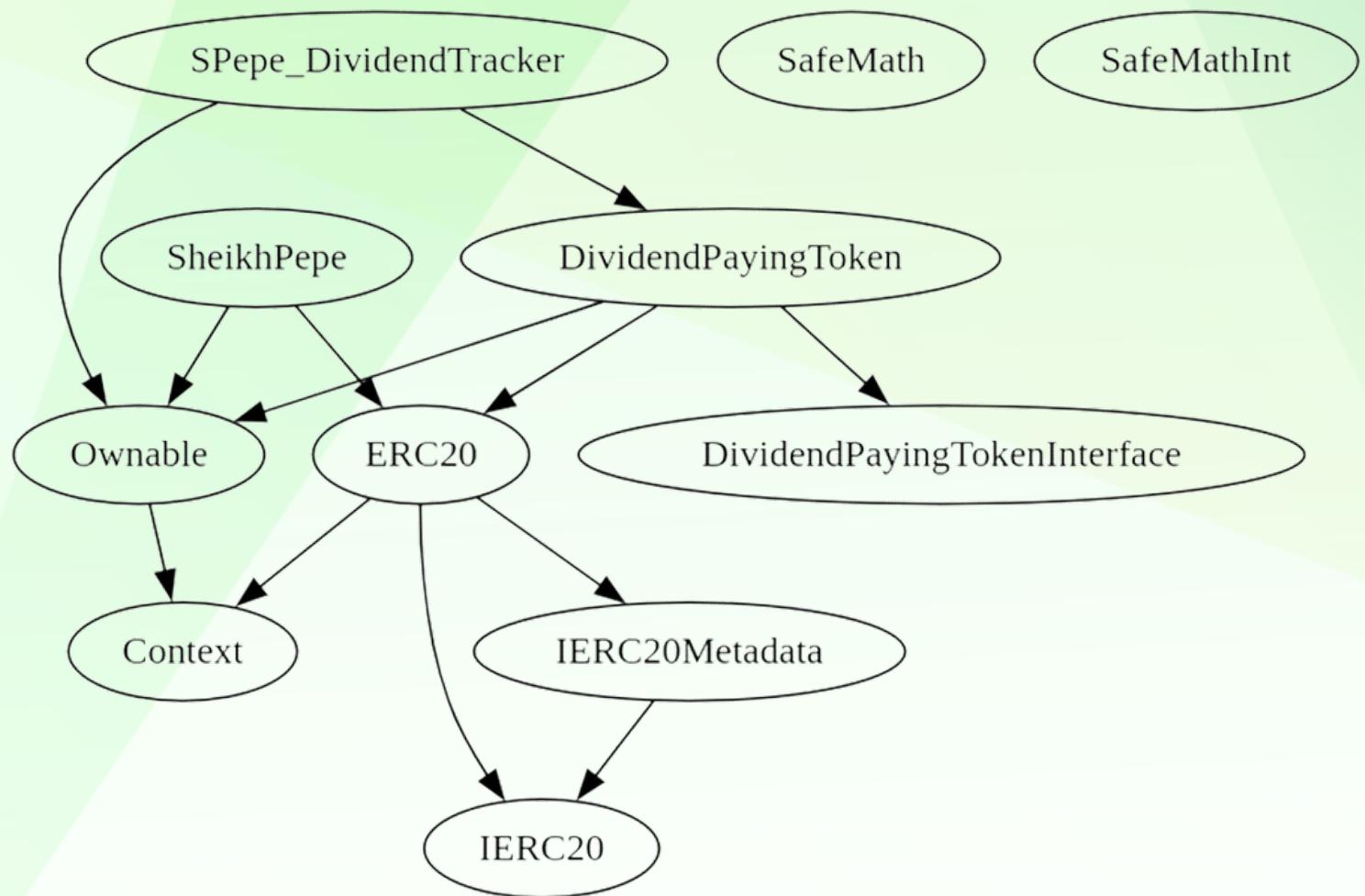
CLASSIFICATION OF RISK

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	0
◆ Medium-Risk	1
◆ Low-Risk	0
◆ Gas Optimization / Suggestions	0

INHERITANCE TREE





POINTS TO NOTE

- Owner is not able to set buy/sell fees more than 10% (20% max fee)
- Owner is not able to set transfer fees (0% always)
- Owner is not able to set max buy/sell/transfer/hold 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 trading for investors**



CONTRACT ASSESSMENT

Contract	Type	Bases			
L **Function Name** **Visibility** **Mutability** **Modifiers** **Context** Implementation					
L _msgSender	Internal	🔒			
L _msgData	Internal	🔒			
IERC20 Interface					
L totalSupply	External	!	NO	!	
L balanceOf	External	!	NO	!	
L transfer	External	!	●	NO	!
L allowance	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	!	
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 _tokengeneration	Internal	🔒	●		
L _burn	Internal	🔒	●		
L _approve	Internal	🔒	●		
L _beforeTokenTransfer	Internal	🔒	●		
SafeMath Library					
L add	Internal	🔒			
L sub	Internal	🔒			
L sub	Internal	🔒			



CONTRACT ASSESSMENT

```
| 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 | 🔒 | |||
|||||
| **Ownable** | Implementation | Context ||
| L | <Constructor> | Public ! | ● NO ! |
| L | owner | Public ! | | NO ! |
| L | renounceOwnership | Public ! | | ● | onlyOwner |
| L | transferOwnership | Public ! | | ● | onlyOwner |
|||||
| **IPair** | Interface | |||
| L | sync | External ! | | ● | NO ! |
|||||
| **IFactory** | Interface | |||
| L | createPair | External ! | | ● | NO ! |
| L | getPair | External ! | | NO ! |
|||||
| **IRouter** | Interface | |||
| L | factory | External ! | | NO ! |
| L | WETH | External ! | | NO ! |
| L | addLiquidityETH | External ! | | $ | NO ! |
| L | swapExactTokensForTokensSupportingFeeOnTransferTokens | External ! | | ● | NO ! |
| L | swapExactETHForTokens | External ! | | $ | NO ! |
| L | swapExactTokensForETHSupportingFeeOnTransferTokens | External ! | | ● | NO ! |
|||||
| **DividendPayingTokenInterface** | Interface | |||
| L | dividendOf | External ! | | NO ! |
| L | distributeDividends | External ! | | $ | NO ! |
| L | withdrawableDividendOf | External ! | | NO ! |
| L | withdrawnDividendOf | External ! | | NO ! |
| L | accumulativeDividendOf | External ! | | NO ! |
```

CONTRACT ASSESSMENT

```
||||| |
| **DividendPayingToken** | Implementation | ERC20, DividendPayingTokenInterface, Ownable |||  
| L |<Constructor> | Public ! | ● | ERC20 |  
| L |<Receive Ether> | External ! | $ | NO ! |  
| L | distributeDividends | Public ! | $ | NO ! |  
| L | _withdrawDividendOfUser | Internal 🔒 | ● |  
| L | setRewardToken | External ! | ● | onlyOwner |  
| L | swapBnbForCustomToken | Internal 🔒 | ● |  
| L | dividendOf | Public ! | NO ! |  
| L | withdrawableDividendOf | Public ! | NO ! |  
| L | withdrawnDividendOf | Public ! | NO ! |  
| L | accumulativeDividendOf | Public ! | NO ! |  
| L | _transfer | Internal 🔒 | ● |  
| L | _tokengeneration | Internal 🔒 | ● |  
| L | _burn | Internal 🔒 | ● |  
| L | _setBalance | Internal 🔒 | ● |  
|||||  
| **IterableMapping** | Library | |||  
| L | get | Internal 🔒 | | |  
| L | getIndexOfKey | Internal 🔒 | | |  
| L | getKeyAtIndex | Internal 🔒 | | |  
| L | size | Internal 🔒 | | |  
| L | set | Internal 🔒 | ● |  
| L | remove | Internal 🔒 | ● |  
|||||  
| **Address** | Library | |||  
| L | sendValue | Internal 🔒 | ● |  
|||||  
| **SheikhPepe** | Implementation | ERC20, Ownable |||  
| L |<Constructor> | Public ! | ● | ERC20 |  
| L |<Receive Ether> | External ! | $ | NO ! |  
| L | processDividendTracker | External ! | ● | NO ! |  
| L | claim | External ! | ● | NO ! |  
| L | ClearBEP20Tokens | External ! | ● | onlyOwner |  
| L | ClearStuckBNB | External ! | ● | NO ! |  
| L | excludeFromFees | Public ! | ● | onlyOwner |  
| L | excludeMultipleAccountsFromFees | Public ! | ● | onlyOwner |  
| L | excludeFromDividends | External ! | ● | onlyOwner |  
| L | setMarketingWallet | External ! | ● | onlyOwner |  
| L | setSwapTokensAtAmount | External ! | ● | onlyOwner |  
| L | UpdateBuyTaxes | External ! | ● | onlyOwner |  
| L | UpdateSellTaxes | External ! | ● | onlyOwner |  
| L | setSwapEnabled | External ! | ● | onlyOwner |
```



CONTRACT ASSESSMENT

```
| L | EnableTradingEnabled | External ! | ● | onlyOwner | |
| L | setBotBlocks | External ! | ● | onlyOwner |
| L | setMinBalanceForDividends | External ! | ● | onlyOwner |
| L | _setAutomatedMarketMakerPair | Private 🔒 | ● |||
| L | setGasForProcessing | External ! | ● | onlyOwner |
| L | setClaimWait | External ! | ● | onlyOwner |
| L | getClaimWait | External ! | | NO ! |
| L | getTotalDividendsDistributed | External ! | | NO ! |
| L | isExcludedFromFees | Public ! | | NO ! |
| L | withdrawableDividendOf | Public ! | | NO ! |
| L | getCurrentRewardToken | External ! | | NO ! |
| L | dividendTokenBalanceOf | Public ! | | NO ! |
| L | getAccountDividendsInfo | External ! | | NO ! |
| L | getAccountDividendsInfoAtIndex | External ! | | NO ! |
| L | getLastProcessedIndex | External ! | | NO ! |
| L | getNumberOfDividendTokenHolders | External ! | | NO ! |
| L | _transfer | Internal 🔒 | ● |||
| L | swapAndLiquify | Private 🔒 | ● |||
| L | swapTokensForBNB | Private 🔒 | ● |||
| L | addLiquidity | Private 🔒 | ● |||
|||||
| **SPepe_DividendTracker** | Implementation | Ownable, DividendPayingToken ||
| L | <Constructor> | Public ! | ● | DividendPayingToken |
| L | _transfer | Internal 🔒 | |||
| L | setMinBalanceForDividends | External ! | ● | onlyOwner |
| L | excludeFromDividends | External ! | ● | onlyOwner |
| L | updateClaimWait | External ! | ● | onlyOwner |
| L | getLastProcessedIndex | External ! | | NO ! |
| L | getNumberOfTokenHolders | External ! | | NO ! |
| L | getCurrentRewardToken | External ! | | NO ! |
| L | getAccount | Public ! | | NO ! |
| L | getAccountAtIndex | Public ! | | NO ! |
| L | canAutoClaim | Private 🔒 | |||
| L | setBalance | Public ! | ● | onlyOwner |
| L | process | Public ! | ● | NO ! |
| L | processAccount | Public ! | ● | onlyOwner |
|||||
| **DividendDistributor** | Implementation | IDividendDistributor ||
| L | <Constructor> | Public ! | ● | NO ! |
| L | setDistributionCriteria | External ! | ● | onlyToken |
| L | setShare | External ! | ● | onlyToken |
| L | deposit | External ! | 💸 | onlyToken |
```



CONTRACT ASSESSMENT

```
| L | process | External ! | ● | onlyToken | |
| L | shouldDistribute | Internal 🔒 | || |
| L | distributeDividend | Internal 🔒 | ● | |
| L | claimDividend | External ! | ● | onlyToken |
| L | getUnpaidEarnings | Public ! | | NO ! |
| L | getCumulativeDividends | Internal 🔒 | || |
| L | addShareholder | Internal 🔒 | ● | |
| L | removeShareholder | Internal 🔒 | ● | |
| L | setDividendTokenAddress | External ! | ● | onlyToken |
|||||||
| **FARMPEPE** | Implementation | IBEP20, Auth ||
| L | <Constructor> | Public ! | ● | Auth |
| L | <Receive Ether> | External ! | 💸 | NO ! |
| L | totalSupply | External ! | | NO ! |
| L | decimals | External ! | | NO ! |
| L | symbol | External ! | | NO ! |
| L | name | External ! | | NO ! |
| L | getOwner | External ! | | NO ! |
| L | balanceOf | Public ! | | NO ! |
| L | allowance | External ! | | NO ! |
| L | approve | Public ! | ● | NO ! |
| L | approveMax | External ! | ● | NO ! |
| L | savetokens | External ! | ● | devwall |
| L | burning | External ! | ● | NO ! |
| L | transfer | External ! | ● | NO ! |
| L | transferFrom | External ! | ● | NO ! |
| L | _transferFrom | Internal 🔒 | ● | |
| L | _basicTransfer | Internal 🔒 | ● | |
| L | shouldTakeFee | Internal 🔒 | || |
| L | shouldTakeFee | Internal 🔒 | || |
| L | getTotalFee | Public ! | | NO ! |
| L | getMultipliedFee | Public ! | | NO ! |
| L | takeFee | Internal 🔒 | ● | |
| L | shouldSwapBack | Internal 🔒 | || |
| L | swapBack | Internal 🔒 | ● | swapping |
| L | buyTokens | Internal 🔒 | ● | swapping |
| L | launched | Internal 🔒 | || |
| L | setFeeReceivers | External ! | ● | devwall |
| L | setSwapBackSettings | External ! | ● | devwall |
| L | setTargetLiquidity | External ! | ● | onlyOwner |
| L | manualSend | External ! | ● | NO ! |
| L | setDistributionCriteria | External ! | ● | onlyOwner |
```

CONTRACT ASSESSMENT

L claimDividend External ! ● NO !
L getUnpaidEarnings Public ! NO !
L setDistributorSettings External ! ● onlyOwner
L getCirculatingSupply Public ! NO !
L getLiquidityBacking Public ! NO !
L isOverLiquified Public ! NO !
Auth Implementation
L <Constructor> Public ! ● NO !
L isOwner Public ! NO !
L isdevwallet Public ! NO !
L transferOwnership Public ! ● onlyOwner
L renounceOwnership Public ! ● onlyOwner
IERC165 Interface
L supportsInterface External ! NO !
IBEP20 Interface
L totalSupply External ! NO !
L decimals External ! NO !
L symbol External ! NO !
L name External ! NO !
L getOwner External ! NO !
L balanceOf External ! NO !
L transfer External ! ● NO !
L burning External ! ● NO !
L allowance External ! NO !
L approve External ! ● NO !
L transferFrom External ! ● NO !
IDEXFactory Interface
L createPair External ! ● NO !
IDEXRouter Interface
L factory External ! NO !
L WETH External ! NO !
L addLiquidity External ! ● NO !
L addLiquidityETH External ! \$ NO !
L swapExactTokensForTokensSupportingFeeOnTransferTokens External ! ● NO !
L swapExactETHForTokensSupportingFeeOnTransferTokens External ! \$ NO !
L swapExactTokensForETHSupportingFeeOnTransferTokens External ! ● NO !
SafeMath Library

CONTRACT ASSESSMENT

	L	tryAdd	Internal			
	L	trySub	Internal			
	L	tryMul	Internal			
	L	tryDiv	Internal			
	L	tryMod	Internal			
	L	add	Internal			
	L	sub	Internal			
	L	mul	Internal			
	L	div	Internal			
	L	mod	Internal			
	L	sub	Internal			
	L	div	Internal			
	L	mod	Internal			
	IDividendDistributor	Interface				
	L	setDistributionCriteria	External			 NO 
	L	setShare	External			 NO 
	L	deposit	External			 NO 
	L	process	External			 NO 

	Symbol	Meaning	
	----- -----		
		Function can modify state	
		Function is payable	



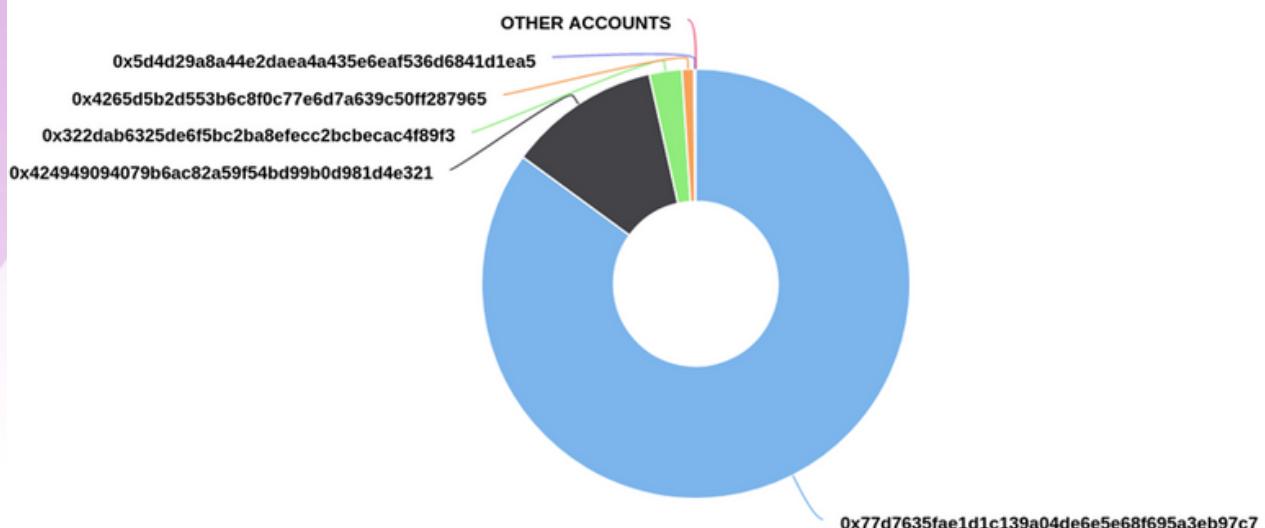
TOKEN DISTRIBUTION AT TIME OF AUDIT

💡 The top 100 holders collectively own 100.00% (1,000,000,000.00 Tokens) of Sheikh Pepe

💡 Token Total Supply: 1,000,000,000.00 Token | Total Token Holders: 5

Sheikh Pepe Top 100 Token Holders

Source: BscScan.com





STATIC ANALYSIS

```
Function IRouter.WETH() (contracts/Token.sol#740) is not in mixedCase
Parameter DividendPayingToken.dividendOf(address).owner (contracts/Token.sol#956) is not in mixedCase
Parameter DividendPayingToken.withdrawableDividendOf(address).owner (contracts/Token.sol#964) is not in mixedCase
Parameter DividendPayingToken.withdrawnDividendOf(address).owner (contracts/Token.sol#973) is not in mixedCase
Parameter DividendPayingToken.accumulativeDividendOf(address).owner (contracts/Token.sol#984) is not in mixedCase
Constant DividendPayingToken.magnitude (contracts/Token.sol#830) is not in UPPER_CASE_WITH_UNDERSCORES
Function SheikhPepe.ClearBEP20Tokens(address) (contracts/Token.sol#1243-1252) is not in mixedCase
Function SheikhPepe.ClearStuckBNB() (contracts/Token.sol#1254-1257) is not in mixedCase
Function SheikhPepe.UpdateBuyTaxes(uint256,uint256,uint256) (contracts/Token.sol#1306-1316) is not in mixedCase
Parameter SheikhPepe.UpdateBuyTaxes(uint256,uint256,uint256).rewards (contracts/Token.sol#1307) is not in mixedCase
Parameter SheikhPepe.UpdateBuyTaxes(uint256,uint256,uint256).marketing (contracts/Token.sol#1308) is not in mixedCase
Parameter SheikhPepe.UpdateBuyTaxes(uint256,uint256,uint256).liquidity (contracts/Token.sol#1309) is not in mixedCase
Function SheikhPepe.UpdateSellTaxes(uint256,uint256,uint256) (contracts/Token.sol#1318-1328) is not in mixedCase
Parameter SheikhPepe.UpdateSellTaxes(uint256,uint256,uint256).rewards (contracts/Token.sol#1319) is not in mixedCase
Parameter SheikhPepe.UpdateSellTaxes(uint256,uint256,uint256).marketing (contracts/Token.sol#1320) is not in mixedCase
Parameter SheikhPepe.UpdateSellTaxes(uint256,uint256,uint256).liquidity (contracts/Token.sol#1321) is not in mixedCase
Parameter SheikhPepe.setSwapEnabled(bool).enabled (contracts/Token.sol#1330) is not in mixedCase
Function SheikhPepe.EnableTradingEnabled() (contracts/Token.sol#1334-1338) is not in mixedCase
Constant SheikhPepe.deadWallet (contracts/Token.sol#1148-1149) is not in UPPER_CASE_WITH_UNDERSCORES
Variable SheikhPepe.BuyTaxes (contracts/Token.sol#1165-1166) is not in mixedCase
Variable SheikhPepe.SellTaxes (contracts/Token.sol#1167-1168) is not in mixedCase
Contract SPePe_DividendTracker (contracts/Token.sol#1616-1866) is not in CapWords
Parameter SPePe_DividendTracker.getAccount(address).account (contracts/Token.sol#1697) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
Redundant expression "this (contracts/Token.sol#15)" inContext (contracts/Token.sol#9-18)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

Variable DividendPayingToken._withdrawDividendOfUser(address)._withdrawableDividend (contracts/Token.sol#887) is too similar to SPePe_DividendTracker.getAccount(address).withdrawableDividends (contracts/Token.sol#1705)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-too-similar

SheikhPepe.constructor() (contracts/Token.sol#1195-1219) uses literals with too many digits:
- _tokengeneration(owner(),1000000000 * (10 ** 9)) (contracts/Token.sol#1218)
SheikhPepe.setGasForProcessing(uint256) (contracts/Token.sol#1367-1378) uses literals with too many digits:
- require(bool,string)(newValue >= 200000 && newValue <= 500000, GasForProcessing must be between 200,000 and 500,000) (contracts/Token.sol#1368-1371)
SheikhPepe.slitherConstructorVariables() (contracts/Token.sol#1135-1614) uses literals with too many digits:
- gasForProcessing = 300000 (contracts/Token.sol#1170)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits

SafeMathInt.MAX_INT256 (contracts/Token.sol#594) is never used in SafeMathInt (contracts/Token.sol#592-649)
SheikhPepe.currentRewardToken (contracts/Token.sol#1154) is never used in SheikhPepe (contracts/Token.sol#1135-1614)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

SheikhPepe.currentRewardToken (contracts/Token.sol#1154) should be constant
SheikhPepe.launchtax (contracts/Token.sol#1173) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant

DividendPayingToken.router (contracts/Token.sol#832) should be immutable
SheikhPepe.BuyTaxes (contracts/Token.sol#1165-1166) should be immutable
SheikhPepe.SellTaxes (contracts/Token.sol#1167-1168) should be immutable
SheikhPepe.dividendTracker (contracts/Token.sol#1146) should be immutable
SheikhPepe.pair (contracts/Token.sol#1139) should be immutable
SheikhPepe.router (contracts/Token.sol#1138) 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

Router (PCS V2):

0xD99D1c33F9fC3444f8101754aBC46c52416550D1

All the functionalities have been tested, no issues were found

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

<https://testnet.bscscan.com/tx/0xddbe106d589eeb1faca5a0325f32b8869d9606e7b353c229923df26d58b0194f>

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

<https://testnet.bscscan.com/tx/0x023d83d73db8e08488f5c85499e8ce78c65f6bcb473b1c9350d00ae3d0ee7ab8>

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

<https://testnet.bscscan.com/tx/0x184e327505af65a2913d86f1551b493141965f6c6942105e9ce92c77b2c6a039>

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

<https://testnet.bscscan.com/tx/0xd1d0f9a78a9934ecd044b1e350ce279d0972eadb289c6916048b147465566646>

5- Buying when not excluded from fees (up to 10% tax) (**passed**):

<https://testnet.bscscan.com/tx/0x505748031254edcd92551cebf310544ffc1c3a35d5fc57544bc3b286c3b5145d>

6- Selling when not excluded from fees (up to 10% tax) (**passed**):

<https://testnet.bscscan.com/tx/0x5cf51a591626c6cbdbd456626e28224d3f7658e6400c3c43056f244c420783b4>



FUNCTIONAL TESTING

7- Transferring when not excluded from fees (0% tax) (passed):

<https://testnet.bscscan.com/tx/0x435f7e269b1a14357516be08b25ef3500faf1ba6b6f1f70ddca551cbecd22b05>

8- Internal swap (passed):

As can seen in this transaction, marketing wallet received BNB

<https://testnet.bscscan.com/address/0xd930f6ea08de4eeab6e680ad196d2b6d0c2eeb23#internaltx>

9- Auto Liquidity (passed):

Auto liquidity generated tokens are burnt, as can be seen in this transaction

https://testnet.bscscan.com/token/0x424949094079b6ac82a59f54bd99b0d981d4e321a=0x00dead

8- Distribution of rewards (passed):

BUSD tokens are distributed between holders, this can be seen in this transaction

<https://testnet.bscscan.com/tx/0x5cf51a591626c6cbdbd456626e28224d3f7658e6400c3c43056f244c420783b4>



MANUAL TESTING

Centralization – Trades must be enabled

Severity: Medium

function: EnableTradingEnabled

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 EnableTradingEnabled() external onlyOwner {  
    require(!tradingEnabled, "Trading is already enabled");  
    tradingEnabled = true;  
    startTradingBlock = block.number;  
}
```

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



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>
