auditace.tech

AuditAce

FROM INCEPTION TO SUCCESS

Smart Contract Audit

FOR

SCEO

DATED : 21 April 23'

AUDIT SUMMARY

Project name – ScoobyCEO

Date: 21 April, 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	0	0	1
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/0x19f930133019f75e7043adadf50097c0aaded72a

Token Information

Token Name : ScoobyCEO

Token Symbol: SCEO

Decimals: 9

Token Supply: 420,000,000,000,000

Token Address:
0x2dD708CD12d8C0901774e20F1Cf9608857fD5eA4

Checksum:
3f6c6937f966fec20f5d2162988372cefad97c9c

Owner:
0x0670fd9192e2B4De7E971A28B9Bb016A14cf4ef6
(at time of writing the audit)

Deployer:
0x30a8ff0dEcabaB561BA358bdFf064A77C3d542f1

TOKEN OVERVIEW

Fees:

Buy Fees: up to 12%

Sell Fees: up to 12%

Transfer Fees: 0%

Fees Privilege: Owner

Ownership: Owned

Minting: No mint function

Max Tx Amount/ Max Wallet Amount: No

Blacklist: Yes

Other Privileges: excluding from fees - including in fees - 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 isexercised 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
- ✓ Private modifier
- ✓ Multiple Sends
- ✓ Using Suicide
- ✓ Gas Limitand Loops
- ✓ Address hardcoded
- ✓ Exception Disorder
- ✓ Using inline assembly
- ✓ Divide before multiply
- ✓ Missing Zero Address Validation
- ✓ Compiler version not fixed
- ✓ **Gasless Send**
- ✓ Using block.timestamp
- ✓ Re-entrancy
- ✓ Tautology or contradiction
- ✓ Timestamp Dependence
- ✓ Revert/require functions
- ✓ Use of tx.origin
- ✓ Integer overflow/underflow
- ✓ Dangerous strict equalities
- ✓ Using SHA3
- ✓ Using throw

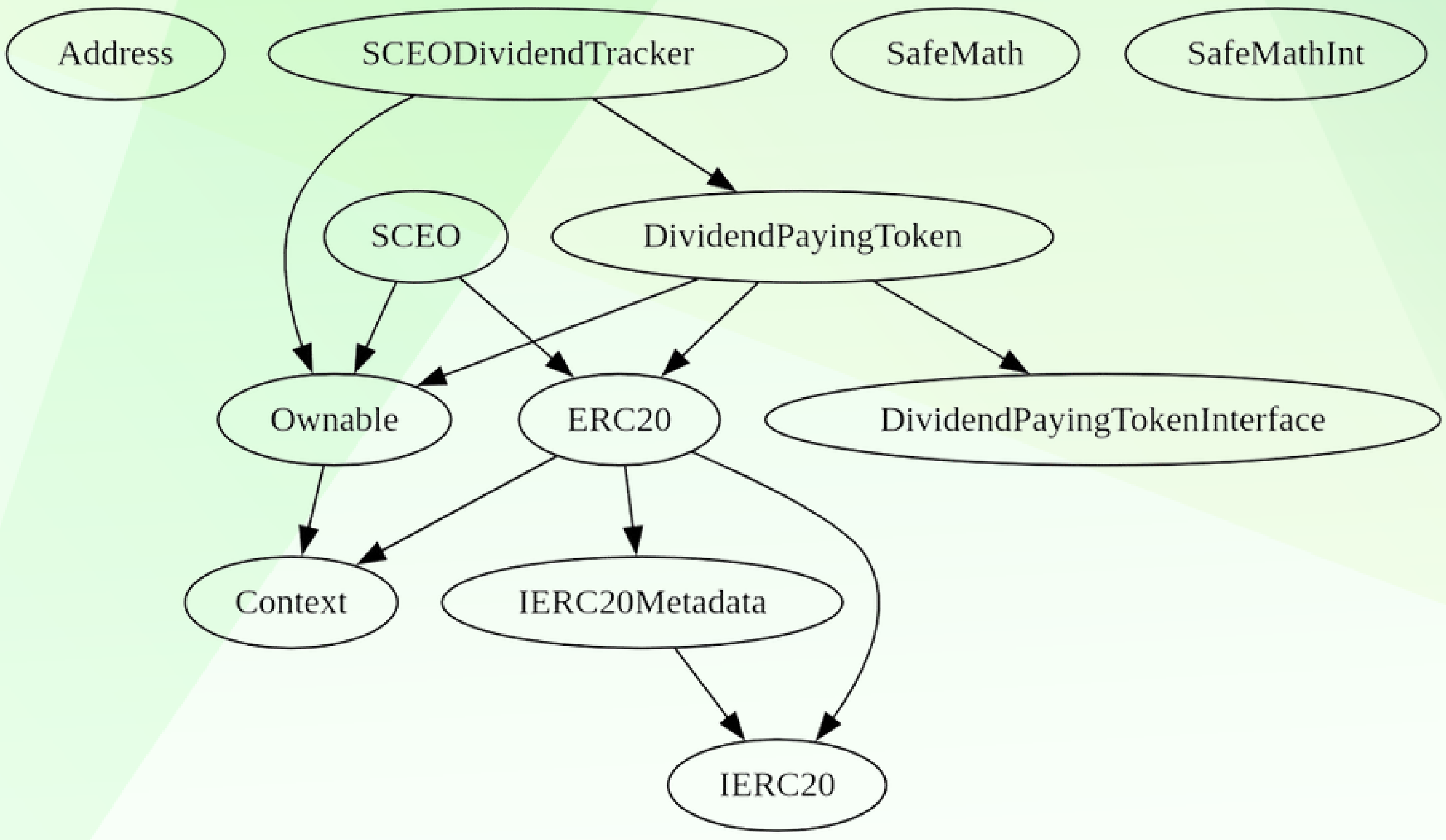
CLASSIFICATION OF RISK

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

INHERITANCE TREE



POINTS TO NOTE

- **Owner is not able to set buy/sell fees higher than 12%**
- **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 in order to be able to trade**

CONTRACT ASSESMENT

```
| Contract |       Type        |      Bases       |                  |                 |
|:----------:|:-------------------:|:----------------:|:----------------:|:---------------:|
|    └     | **Function Name** | **Visibility** | **Mutability** | **Modifiers** |
||||||
| **Address** | Library | |||
| └ | sendValue | Internal 🔒 | 🛑 | |
||||||
| **SCEO** | Implementation | ERC20, Ownable |||
| └ | <Constructor> | Public ❗️ | 🛑 | ERC20 |
| └ | <Receive Ether> | External ❗️ | 💵 |NO ❗️ |
| └ | updateDividendTracker | Public ❗️ | 🛑 | onlyOwner |
| └ | processDividendTracker | External ❗️ | 🛑 |NO ❗️ |
| └ | updateBuyTaxes | Public ❗️ | 🛑 | onlyOwner |
| └ | updateSellTaxes | Public ❗️ | 🛑 | onlyOwner |
| └ | claimRewards | External ❗️ | 🛑 |NO ❗️ |
| └ | withdrawToken | External ❗️ | 🛑 | onlyOwner |
| └ | withdrawETH | External ❗️ | 🛑 | onlyOwner |
| └ | excludeFromFees | Public ❗️ | 🛑 | onlyOwner |
| └ | excludeMultipleAccountsFromFees | Public ❗️ | 🛑 | onlyOwner |
| └ | excludeFromDividends | External ❗️ | 🛑 | onlyOwner |
| └ | setMarketingWallet | External ❗️ | 🛑 | onlyOwner |
| └ | setSwapTokensAtAmount | External ❗️ | 🛑 | onlyOwner |
| └ | setSwapEnabled | External ❗️ | 🛑 | onlyOwner |
| └ | EnableTrading | External ❗️ | 🛑 | onlyOwner |
| └ | setAntiBotBlocks | External ❗️ | 🛑 | onlyOwner |
| └ | setMinBalanceForDividends | External ❗️ | 🛑 | onlyOwner |
| └ | _setAutomatedMarketMakerPair | Private 🔐 | 🛑 | |
| └ | setGasForProcessing | External ❗️ | 🛑 | onlyOwner |
| └ | setClaimWait | External ❗️ | 🛑 | onlyOwner |
| └ | getClaimWait | External ❗️ | |NO ❗️ |
| └ | getTotalDividendsDistributed | External ❗️ | |NO ❗️ |
| └ | isExcludedFromFees | Public ❗️ | |NO ❗️ |
| └ | withdrawableDividendOf | Public ❗️ | |NO ❗️ |
| └ | getCurrentRewardToken | External ❗️ | |NO ❗️ |
| └ | dividendTokenBalanceOf | Public ❗️ | |NO ❗️ |
| └ | getAccountDividendsInfo | External ❗️ | |NO ❗️ |
| └ | getAccountDividendsInfoAtIndex | External ❗️ | |NO ❗️ |
| └ | getLastProcessedIndex | External ❗️ | |NO ❗️ |
| └ | getNumberOfDividendTokenHolders | External ❗️ | |NO ❗️ |
| └ | _transfer | Internal 🔒 | 🛑 | |
| └ | swapAndLiquify | Private 🔐 | 🛑 | |
| └ | swapTokensForBNB | Private 🔐 | 🛑 | |
||||||
```

CONTRACT ASSESMENT

SCEODividendTracker	Implementation	Ownable, DividendPayingToken		
└	<Constructor>	Public ❗️	🛑	DividendPayingToken
└	_transfer	Internal 🔒		
└	setMinBalanceForDividends	External ❗️	🛑	onlyOwner
└	excludeFromDividends	External ❗️	🛑	onlyOwner
└	updateClaimWait	External ❗️	🛑	onlyOwner
└	getLastProcessedIndex	External ❗️		NO❗️
└	getNumberOfTokenHolders	External ❗️		NO❗️
└	getCurrentRewardToken	External ❗️		NO❗️
└	getAccount	Public ❗️		NO❗️
└	getAccountAtIndex	Public ❗️		NO❗️
└	canAutoClaim	Private 🔐		
└	setBalance	Public ❗️	🛑	onlyOwner
└	process	Public ❗️	🛑	NO❗️
└	processAccount	Public ❗️	🛑	onlyOwner
DividendPayingToken	Implementation	ERC20, DividendPayingTokenInterface, Ownable		
└	<Constructor>	Public ❗️	🛑	ERC20
└	<Receive Ether>	External ❗️	💵	NO❗️
└	distributeDividends	Public ❗️	💵	NO❗️
└	_withdrawDividendOfUser	Internal 🔒	🛑	
└	setRewardToken	External ❗️	🛑	onlyOwner
└	swapBnbForCustomToken	Internal 🔒	🛑	
└	dividendOf	Public ❗️		NO❗️
└	withdrawableDividendOf	Public ❗️		NO❗️
└	withdrawnDividendOf	Public ❗️		NO❗️
└	accumulativeDividendOf	Public ❗️		NO❗️
└	_transfer	Internal 🔒	🛑	
└	_tokengeneration	Internal 🔒	🛑	
└	_burn	Internal 🔒	🛑	
└	_setBalance	Internal 🔒	🛑	
ERC20	Implementation	Context, IERC20, IERC20Metadata		
└	<Constructor>	Public ❗️	🛑	NO❗️
└	name	Public ❗️		NO❗️
└	symbol	Public ❗️		NO❗️
└	decimals	Public ❗️		NO❗️
└	totalSupply	Public ❗️		NO❗️
└	balanceOf	Public ❗️		NO❗️
└	transfer	Public ❗️	🛑	NO❗️
└	allowance	Public ❗️		NO❗️
└	approve	Public ❗️	🛑	NO❗️

```
| └ | transferFrom | Public ❗️ | 🛑  |NO❗️ |
| └ | increaseAllowance | Public ❗️ | 🛑  |NO❗️ |
| └ | decreaseAllowance | Public ❗️ | 🛑  |NO❗️ |
| └ | _transfer | Internal 🔒 | 🛑  | |
| └ | _tokengeneration | Internal 🔒 | 🛑  | |
| └ | _burn | Internal 🔒 | 🛑  | |
| └ | _approve | Internal 🔒 | 🛑  | |
| └ | _beforeTokenTransfer | Internal 🔒 | 🛑  | |
||||||
| **IERC20** | Interface |  |||
| └ | totalSupply | External ❗️ |   |NO❗️ |
| └ | balanceOf | External ❗️ |   |NO❗️ |
| └ | transfer | External ❗️ | 🛑  |NO❗️ |
| └ | allowance | External ❗️ |   |NO❗️ |
| └ | approve | External ❗️ | 🛑  |NO❗️ |
| └ | transferFrom | External ❗️ | 🛑  |NO❗️ |
||||||
| **IERC20Metadata** | Interface | IERC20 |||
| └ | name | External ❗️ |   |NO❗️ |
| └ | symbol | External ❗️ |   |NO❗️ |
| └ | decimals | External ❗️ |   |NO❗️ |
||||||
| **Context** | Implementation |  |||
| └ | _msgSender | Internal 🔒 |   | |
| └ | _msgData | Internal 🔒 |   | |
||||||
| **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 🔒 |   | |
```

CONTRACT ASSESMENT

```
||||||
| **SafeMathUint** | Library |  |||
| └ | toInt256Safe | Internal 🔒 |   | |
||||||
| **DividendPayingTokenInterface** | Interface |  |||
| └ | dividendOf | External ❗️ |   |NO❗️ |
| └ | distributeDividends | External ❗️ |  💵 |NO❗️ |
| └ | withdrawableDividendOf | External ❗️ |   |NO❗️ |
| └ | withdrawnDividendOf | External ❗️ |   |NO❗️ |
| └ | accumulativeDividendOf | External ❗️ |   |NO❗️ |
||||||
| **Ownable** | Implementation | Context |||
| └ | <Constructor> | Public ❗️ | 🛑  |NO❗️ |
| └ | owner | Public ❗️ |   |NO❗️ |
| └ | renounceOwnership | Public ❗️ | 🛑  | onlyOwner |
| └ | transferOwnership | Public ❗️ | 🛑  | onlyOwner |
||||||
| **IPair** | Interface |  |||
| └ | sync | External ❗️ | 🛑  |NO❗️ |
||||||
| **IFactory** | Interface |  |||
| └ | createPair | External ❗️ | 🛑  |NO❗️ |
| └ | getPair | External ❗️ |   |NO❗️ |
||||||
| **IRouter** | Interface |  |||
| └ | factory | External ❗️ |   |NO❗️ |
| └ | WETH | External ❗️ |   |NO❗️ |
| └ | addLiquidityETH | External ❗️ |  💵 |NO❗️ |
| └ | swapExactTokensForTokensSupportingFeeOnTransferTokens | External ❗️ | 🛑  |NO❗️ |
| └ | swapExactETHForTokens | External ❗️ |  💵 |NO❗️ |
| └ | swapExactTokensForETHSupportingFeeOnTransferTokens | External ❗️ | 🛑  |NO❗️ |
||||||
| **IterableMapping** | Library |  |||
| └ | get | Internal 🔒 |   | |
| └ | getIndexOfKey | Internal 🔒 |   | |
| └ | getKeyAtIndex | Internal 🔒 |   | |
| └ | size | Internal 🔒 |   | |
| └ | set | Internal 🔒 | 🛑  | |
| └ | remove | Internal 🔒 | 🛑  | |
```

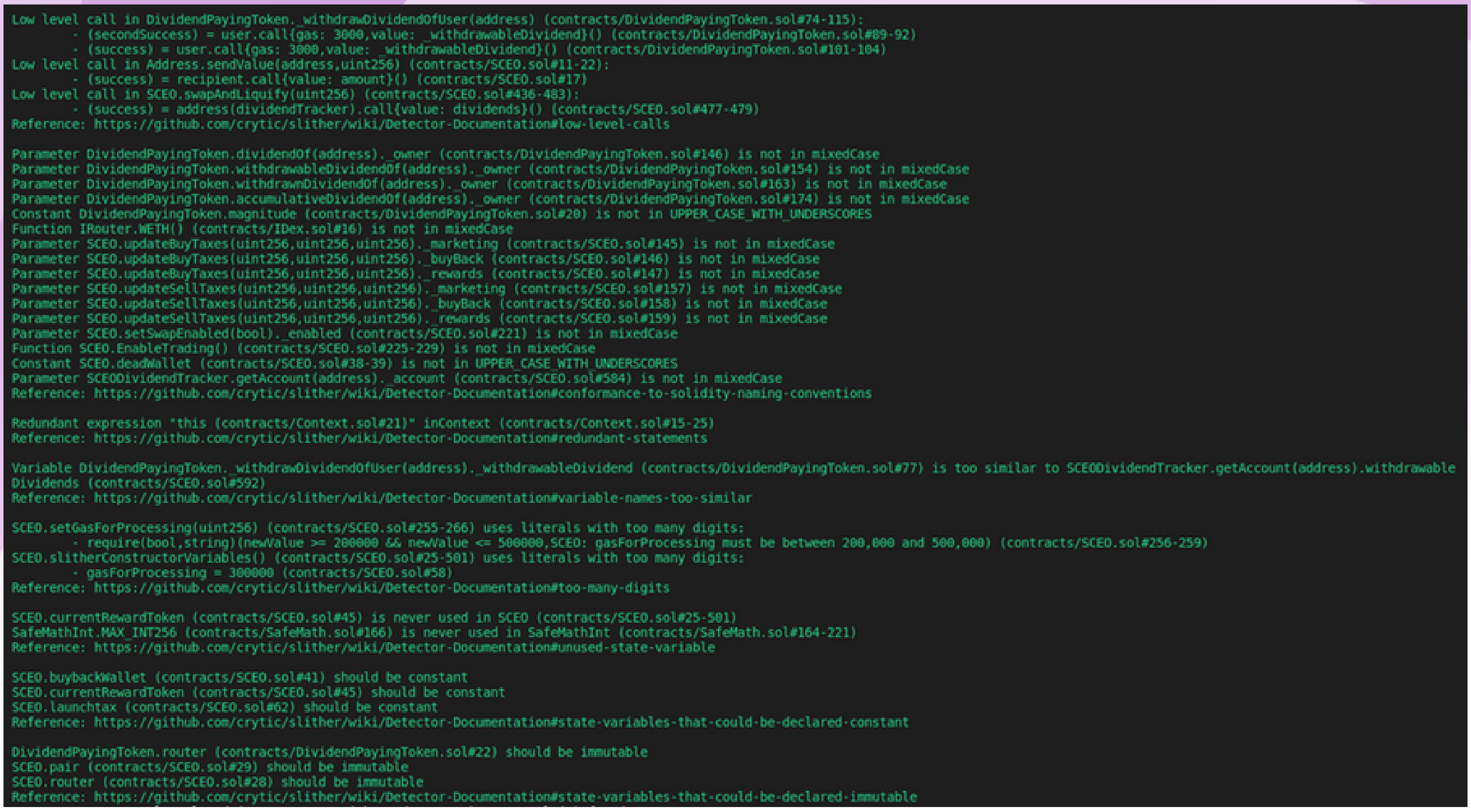
Legend

Symbol	Meaning
🛑	Function can modify state
💵	Function is payable

TOKEN DISTRIBUTION

It should be noted that the owner currently holds 100% of the total supply. However, information about the distribution of these tokens is not available, and it is recommended that investors exercise caution when considering this aspect.

STATIC ANALYSIS



```
Low level call in DividendPayingToken._withdrawDividendOfUser(address) (contracts/DividendPayingToken.sol#74-115):
        - (secondSuccess) = user.call{gas: 3000,value: _withdrawableDividend}() (contracts/DividendPayingToken.sol#89-92)
        - (success) = user.call{gas: 3000,value: _withdrawableDividend}() (contracts/DividendPayingToken.sol#101-104)
Low level call in Address.sendValue(address,uint256) (contracts/SCEO.sol#11-22):
        - (success) = recipient.call{value: amount}() (contracts/SCEO.sol#17)
Low level call in SCEO.swapAndLiquify(uint256) (contracts/SCEO.sol#436-483):
        - (success) = address(dividendTracker).call{value: dividends}() (contracts/SCEO.sol#477-479)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Parameter DividendPayingToken.dividendOf(address)._owner (contracts/DividendPayingToken.sol#146) is not in mixedCase
Parameter DividendPayingToken.withdrawableDividendOf(address)._owner (contracts/DividendPayingToken.sol#154) is not in mixedCase
Parameter DividendPayingToken.withdrawnDividendOf(address)._owner (contracts/DividendPayingToken.sol#163) is not in mixedCase
Parameter DividendPayingToken.accumulativeDividendOf(address)._owner (contracts/DividendPayingToken.sol#174) is not in mixedCase
Constant DividendPayingToken.magnitude (contracts/DividendPayingToken.sol#20) is not in UPPER_CASE_WITH_UNDERSCORES
Function IRouter.WETH() (contracts/IDex.sol#16) is not in mixedCase
Parameter SCEO.updateBuyTaxes(uint256,uint256,uint256)._marketing (contracts/SCEO.sol#145) is not in mixedCase
Parameter SCEO.updateBuyTaxes(uint256,uint256,uint256)._buyBack (contracts/SCEO.sol#146) is not in mixedCase
Parameter SCEO.updateBuyTaxes(uint256,uint256,uint256)._rewards (contracts/SCEO.sol#147) is not in mixedCase
Parameter SCEO.updateSellTaxes(uint256,uint256,uint256)._marketing (contracts/SCEO.sol#157) is not in mixedCase
Parameter SCEO.updateSellTaxes(uint256,uint256,uint256)._buyBack (contracts/SCEO.sol#158) is not in mixedCase
Parameter SCEO.updateSellTaxes(uint256,uint256,uint256)._rewards (contracts/SCEO.sol#159) is not in mixedCase
Parameter SCEO.setSwapEnabled(bool)._enabled (contracts/SCEO.sol#221) is not in mixedCase
Function SCEO.EnableTrading() (contracts/SCEO.sol#225-229) is not in mixedCase
Constant SCEO.deadWallet (contracts/SCEO.sol#38-39) is not in UPPER_CASE_WITH_UNDERSCORES
Parameter SCEODividendTracker.getAccount(address)._account (contracts/SCEO.sol#584) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (contracts/Context.sol#21)" inContext (contracts/Context.sol#15-25)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

Variable DividendPayingToken._withdrawDividendOfUser(address)._withdrawableDividend (contracts/DividendPayingToken.sol#77) is too similar to SCEODividendTracker.getAccount(address).withdrawable
Dividends (contracts/SCEO.sol#592)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-too-similar

SCEO.setGasForProcessing(uint256) (contracts/SCEO.sol#255-266) uses literals with too many digits:
        - require(bool,string)(newValue >= 200000 && newValue <= 500000,SCEO: gasForProcessing must be between 200,000 and 500,000) (contracts/SCEO.sol#256-259)
SCEO.slitherConstructorVariables() (contracts/SCEO.sol#25-501) uses literals with too many digits:
        - gasForProcessing = 300000 (contracts/SCEO.sol#58)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits

SCEO.currentRewardToken (contracts/SCEO.sol#45) is never used in SCEO (contracts/SCEO.sol#25-501)
SafeMathInt.MAX_INT256 (contracts/SafeMath.sol#166) is never used in SafeMathInt (contracts/SafeMath.sol#164-221)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

SCEO.buybackWallet (contracts/SCEO.sol#41) should be constant
SCEO.currentRewardToken (contracts/SCEO.sol#45) should be constant
SCEO.launchtax (contracts/SCEO.sol#62) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant

DividendPayingToken.router (contracts/DividendPayingToken.sol#22) should be immutable
SCEO.pair (contracts/SCEO.sol#29) should be immutable
SCEO.router (contracts/SCEO.sol#28) 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/0x1178915eba4f6fcfdd270c0723c1c5c8f4966636db65aac263975036263deb6c

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

https://testnet.bscscan.com/tx/0x6b266d508aa009e8915646f92c711f80d0906613d69ca3f9dac61b8193370fe2

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

https://testnet.bscscan.com/tx/0x02b767db7aa8bac08aa78992ba0f63675d51fcdd85437c409fd4f63c52716608

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

https://testnet.bscscan.com/tx/0x38a843665500e5f687978a1159ce1fc985afcb42c637b4de9d0c8d5a2ecf6cae

5- Buying when not excluded from fees (up to 12% tax) **(passed):**

https://testnet.bscscan.com/tx/0x123553bbcda51a195493344caa9d9a465d0889fea0de4ddf0841e83c250cbbdd

6- Selling when not excluded from fees (up to 12% tax) **(passed):**

https://testnet.bscscan.com/tx/0x063a2ea45e893faed527ab489ae7a119fe286741fdedd1f4eccbe065124cf132

FUNCTIONAL TESTING

7- Transferring when not excluded from fees (0% tax) (passed):
https://testnet.bscscan.com/tx/0xaf3e7487c1aa24da523fd537c7baa1df92d6e3157
3c31cd3ed3dad69e31210a0

8- Internal swap (passed):
Marketing and buyback wallets received BNB
https://testnet.bscscan.com/address/0xa3c518db271c6eca894831d1cc78c1f78c47
b174#internaltx
https://testnet.bscscan.com/address/0x8016d389d8940474c207bf3ea40464485d
e3fd43#internaltx

MANUAL TESTING

Informational – Owner must enable trades

Function: EnableTrading

Overview:

Owner must enable trading for investors, if trading remain disabled holders will not be able to trade their tokens, once trading is enabled can not be disbled again.

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

DISCLAIMER

ABOUT AUDITACE

We specializes 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