AuditAce
FROM INCEPTION TO SUCCESS

Smart Contract Audit

FOR

Red Doge

DATED : 28 Apr 23'

AUDIT SUMMARY

Project name – Red Doge

Date: 28 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	1	0	0	1
Acknowledged	0	0	0	0	0
Resolved	0	1	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 done on BSC Test network, each test has its transaction has attached to it.

3- Slither : Static Analysis

Testnet Link: all tests were done using this contract, tests are done on BSC Testnet

https://testnet.bscscan.com/token/0x265f1066b5136174184a20c5a3282cc1724acdd7

Token Information

Token Name : RedDoge

Token Symbol: RDoge

Decimals: 9

Token Supply:1,000,000,000

Token Address:
0xE175c9E9F56Ca53256365E66775EA1D3b928d7Ee

Checksum:
07e4cb58d75dab187fbbcb0134b8b8c89da70ab8

Owner:
0x74133652573C748dBc9ED66BAf5F944783314255

Deployer:
0x9ff5037b13Df356A3737097c8f16712F4EA864D0

TOKEN OVERVIEW

Fees:

Buy Fees: upto 12 %

Sell Fees: upto 12 %

Transfer Fees: 0%

Fees Privilige: owned

Ownership : Owned

Minting: No mint function

Max Tx Amount/ Max Wallet Amount: No

Blacklist: No

Other Priviliges: including and excluding form fee - changing swap threshold - modify fee

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

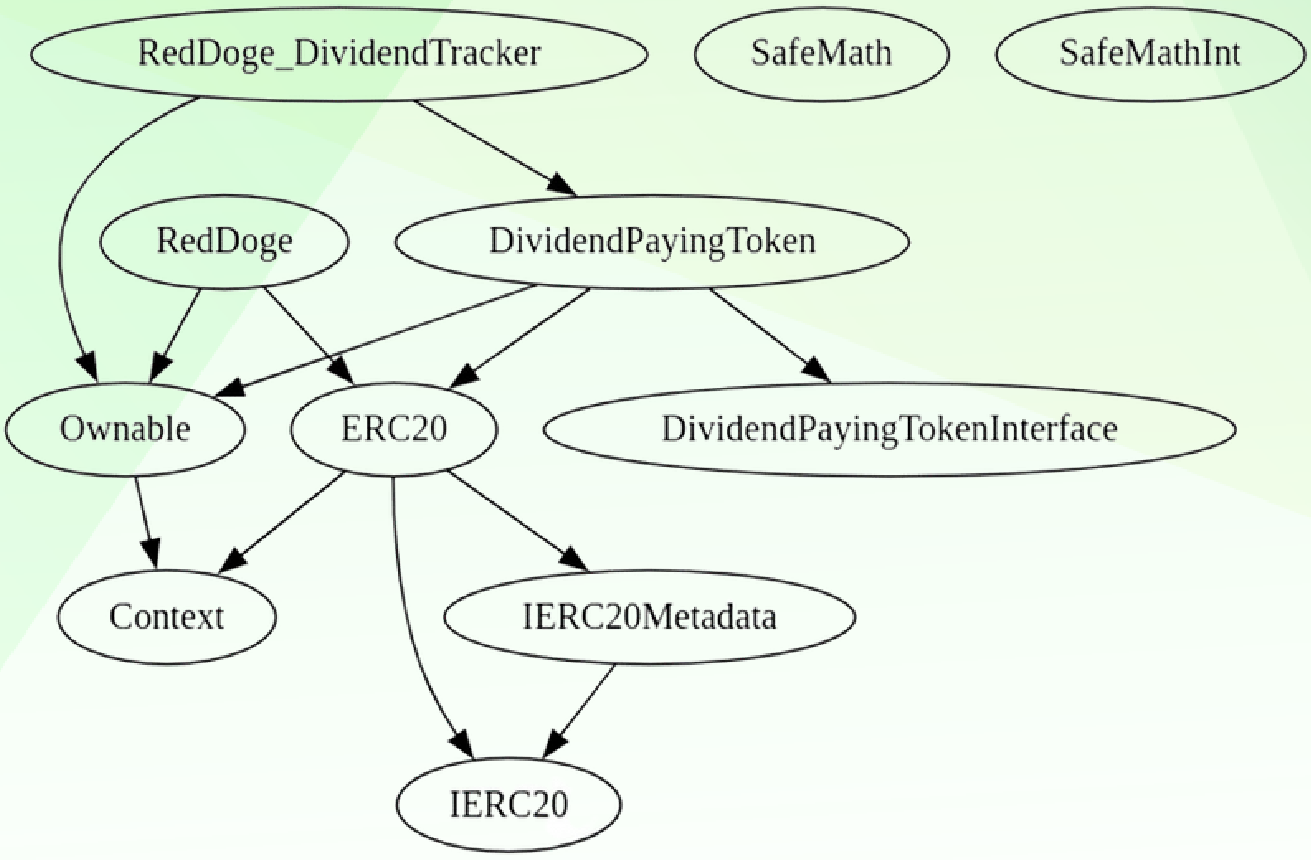
CLASSIFICATION OF RISK

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**	1 (RESOLVED)
◆ **Medium-Risk**	0
◆ **Low-Risk**	0
◆ **Gas Optimization / Suggestions**	1

INHERITANCE TREE



POINTS TO NOTE

- Owner is able to set buy/sell fees more than 12% each
- Owner is not ablet o set transfer taxes (0% forever)
- 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 mint new tokens
- Owner must enable trades for investors manually

CONTRACT ASSESMENT

```
| Contract | Type | Bases | | |
|:----------:|:-------------------:|:----------------:|:---------------:|:---------------:|
| └ | **Function Name** | **Visibility** | **Mutability** | **Modifiers** |
| **Context** | Implementation | |||
| └ | _msgSender | Internal 🔒 | | |
| └ | _msgData | 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❗ |
||||||
| **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 🔒 | 🛑 | |
||||||
| **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 🔒 |   | |
||||||
| **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❗️ |
||||||
| **DividendPayingTokenInterface** | Interface |  |||
| └ | dividendOf | External ❗️ |   |NO❗️ |
| └ | distributeDividends | External ❗️ | 💵 |NO❗️ |
```

CONTRACT ASSESMENT

```
| └ | withdrawableDividendOf | External ❗ |   |NO❗ |
| └ | withdrawnDividendOf | External ❗ |   |NO❗ |
| └ | accumulativeDividendOf | External ❗ |   |NO❗ |
||||||
| **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 🔒 | 🛑  | |
||||||
| **IterableMapping** | Library |  |||
| └ | get | Internal 🔒 |   | |
| └ | getIndexOfKey | Internal 🔒 |   | |
| └ | getKeyAtIndex | Internal 🔒 |   | |
| └ | size | Internal 🔒 |   | |
| └ | set | Internal 🔒 | 🛑  | |
| └ | remove | Internal 🔒 | 🛑  | |
||||||
| **Address** | Library |  |||
| └ | sendValue | Internal 🔒 | 🛑  | |
||||||
| **RedDoge** | Implementation | ERC20, Ownable |||
| └ | <Constructor> | Public ❗ | 🛑  | ERC20 |
| └ | <Receive Ether> | External ❗ |  💵 |NO❗ |
| └ | processDividendTracker | External ❗ | 🛑  |NO❗ |
| └ | claim | External ❗ | 🛑  |NO❗ |
| └ | ClearBEP20Tokens | External ❗ | 🛑  | onlyOwner |
| └ | ClearStuckBNB | External ❗ | 🛑  |NO❗ |
| └ | excludeFromFees | Public ❗ | 🛑  | onlyOwner |
| └ | excludeMultipleAccountsFromFees | Public ❗ | 🛑  | onlyOwner |
| └ | excludeFromDividends | External ❗ | 🛑  | onlyOwner |
| └ | setMarketingWallet | External ❗ | 🛑  | onlyOwner |
```

CONTRACT ASSESMENT

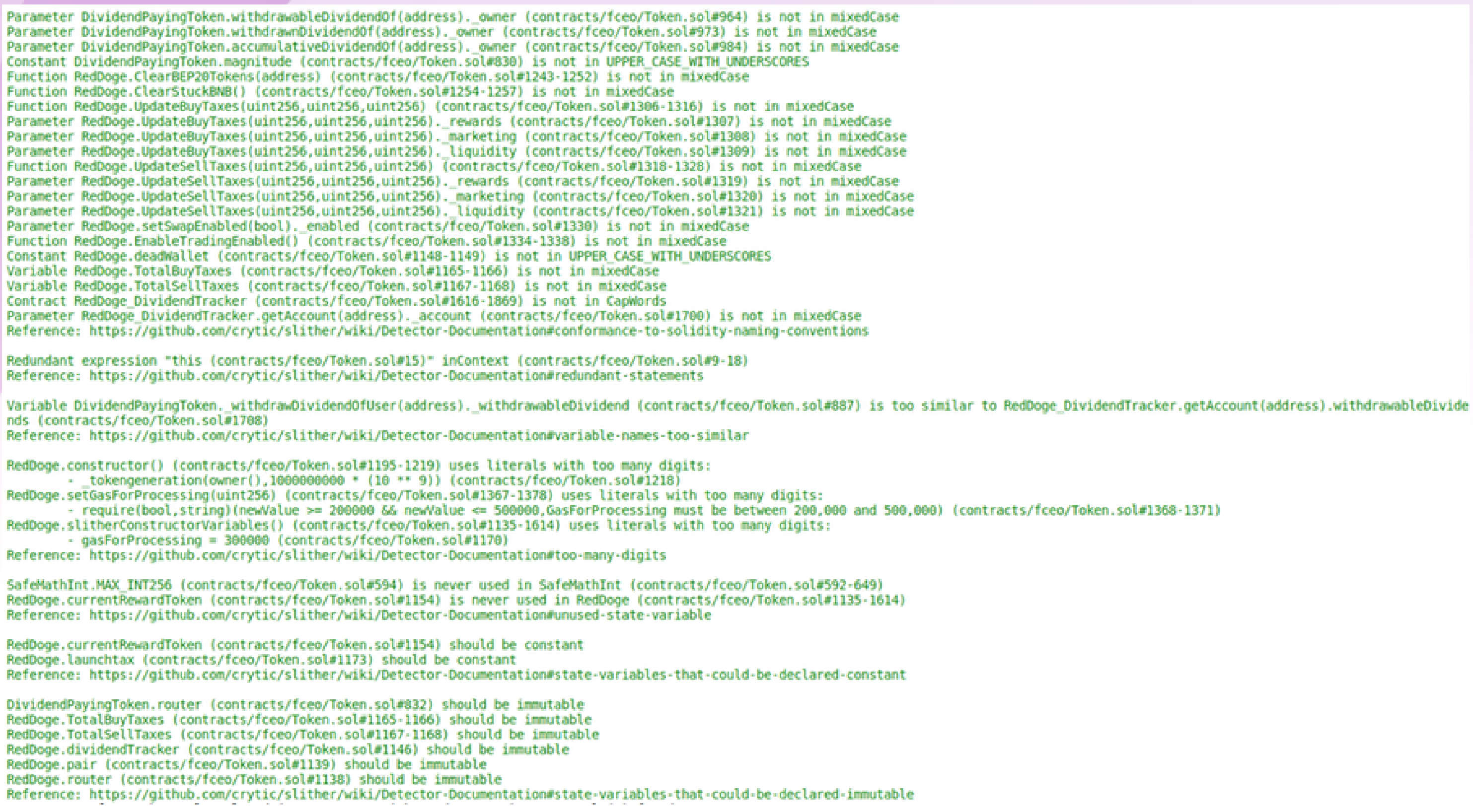
```
| └ | setSwapTokensAtAmount | External ❗ | 🛑  | onlyOwner |
| └ | UpdateBuyTaxes | External ❗ | 🛑  | onlyOwner |
| └ | UpdateSellTaxes | External ❗ | 🛑  | onlyOwner |
| └ | setSwapEnabled | External ❗ | 🛑  | onlyOwner |
| └ | EnableTradingEnabled | External ❗ | 🛑  | onlyOwner |
| └ | setBotBlocks | 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 🔐 | 🛑  | |
| └ | addLiquidity | Private 🔐 | 🛑  | |
||||||
| **RedDoge_DividendTracker** | 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 |
```

CONTRACT ASSESMENT

Legend

Symbol	Meaning
🛑	Function can modify state
💵	Function is payable

STATIC ANALYSIS



```
Parameter DividendPayingToken.withdrawableDividendOf(address)._owner (contracts/fceo/Token.sol#964) is not in mixedCase
Parameter DividendPayingToken.withdrawnDividendOf(address)._owner (contracts/fceo/Token.sol#973) is not in mixedCase
Parameter DividendPayingToken.accumulativeDividendOf(address)._owner (contracts/fceo/Token.sol#984) is not in mixedCase
Constant DividendPayingToken.magnitude (contracts/fceo/Token.sol#830) is not in UPPER_CASE_WITH_UNDERSCORES
Function RedDoge.ClearBEP20Tokens(address) (contracts/fceo/Token.sol#1243-1252) is not in mixedCase
Function RedDoge.ClearStuckBNB() (contracts/fceo/Token.sol#1254-1257) is not in mixedCase
Function RedDoge.UpdateBuyTaxes(uint256,uint256,uint256) (contracts/fceo/Token.sol#1306-1316) is not in mixedCase
Parameter RedDoge.UpdateBuyTaxes(uint256,uint256,uint256)._rewards (contracts/fceo/Token.sol#1307) is not in mixedCase
Parameter RedDoge.UpdateBuyTaxes(uint256,uint256,uint256)._marketing (contracts/fceo/Token.sol#1308) is not in mixedCase
Parameter RedDoge.UpdateBuyTaxes(uint256,uint256,uint256)._liquidity (contracts/fceo/Token.sol#1309) is not in mixedCase
Function RedDoge.UpdateSellTaxes(uint256,uint256,uint256) (contracts/fceo/Token.sol#1318-1328) is not in mixedCase
Parameter RedDoge.UpdateSellTaxes(uint256,uint256,uint256)._rewards (contracts/fceo/Token.sol#1319) is not in mixedCase
Parameter RedDoge.UpdateSellTaxes(uint256,uint256,uint256)._marketing (contracts/fceo/Token.sol#1320) is not in mixedCase
Parameter RedDoge.UpdateSellTaxes(uint256,uint256,uint256)._liquidity (contracts/fceo/Token.sol#1321) is not in mixedCase
Parameter RedDoge.setSwapEnabled(bool)._enabled (contracts/fceo/Token.sol#1330) is not in mixedCase
Function RedDoge.EnableTradingEnabled() (contracts/fceo/Token.sol#1334-1338) is not in mixedCase
Constant RedDoge.deadWallet (contracts/fceo/Token.sol#1148-1149) is not in UPPER_CASE_WITH_UNDERSCORES
Variable RedDoge.TotalBuyTaxes (contracts/fceo/Token.sol#1165-1166) is not in mixedCase
Variable RedDoge.TotalSellTaxes (contracts/fceo/Token.sol#1167-1168) is not in mixedCase
Contract RedDoge_DividendTracker (contracts/fceo/Token.sol#1616-1869) is not in CapWords
Parameter RedDoge_DividendTracker.getAccount(address)._account (contracts/fceo/Token.sol#1700) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (contracts/fceo/Token.sol#15)" inContext (contracts/fceo/Token.sol#9-18)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

Variable DividendPayingToken._withdrawDividendOfUser(address)._withdrawableDividend (contracts/fceo/Token.sol#887) is too similar to RedDoge_DividendTracker.getAccount(address).withdrawableDivide
nds (contracts/fceo/Token.sol#1708)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-too-similar

RedDoge.constructor() (contracts/fceo/Token.sol#1195-1219) uses literals with too many digits:
        - _tokengeneration(owner(),1000000000 * (10 ** 9)) (contracts/fceo/Token.sol#1218)
RedDoge.setGasForProcessing(uint256) (contracts/fceo/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/fceo/Token.sol#1368-1371)
RedDoge.slitherConstructorVariables() (contracts/fceo/Token.sol#1135-1614) uses literals with too many digits:
        - gasForProcessing = 300000 (contracts/fceo/Token.sol#1170)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits

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

RedDoge.currentRewardToken (contracts/fceo/Token.sol#1154) should be constant
RedDoge.launchtax (contracts/fceo/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/fceo/Token.sol#832) should be immutable
RedDoge.TotalBuyTaxes (contracts/fceo/Token.sol#1165-1166) should be immutable
RedDoge.TotalSellTaxes (contracts/fceo/Token.sol#1167-1168) should be immutable
RedDoge.dividendTracker (contracts/fceo/Token.sol#1146) should be immutable
RedDoge.pair (contracts/fceo/Token.sol#1139) should be immutable
RedDoge.router (contracts/fceo/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/0x49c78adc7affb614f5ef21642056e13159937e6f4a9aaf3b5e5038a81dbe328e

2- Buying when excluded (0% tax) **(passed):**
https://testnet.bscscan.com/tx/0x490a2f627cb9210bebf130f37a97937cc3ec9a1273647b3f153fb6bc4cbd7139

3- Selling when excluded (0% tax) **(passed):**
https://testnet.bscscan.com/tx/0x548b1cb0daa306e25fb3eea7248d4fc4dfe00b2ae6fa00615ece794c15a57f45

4- Transferring when excluded (0% tax) **(passed):**
https://testnet.bscscan.com/tx/0xe37e8512811ca2ffe62c16ae4cef3f8b40ba0b247b2194cc3526edc28a92dc69

5- Buying when not excluded (up to 12% tax) **(passed):**
https://testnet.bscscan.com/tx/0xd9bb0d976ed7c75f5d233c85934335df1b36f240064354df98c2ce015c1305a4

6- Selling when not excluded (up to 12% tax) **(passed):**
https://testnet.bscscan.com/tx/0xc2d963c9155b23db4c20d9fa02efadac2f99b507565336a3660233763ea1a77c

FUNCTIONAL TESTING

7- Transferring when not excluded (0% tax) (passed):
https://testnet.bscscan.com/tx/0xe5fa96dad0897d29ea83a3b157671f4e063003164fe69396d4fb4f2a42b82876

8- Internal swap (passed):
Marketing wallet received BNB
https://testnet.bscscan.com/address/0x4235af3386f45ebfcba4f4afa159853300048248#internaltx

9- Reflections(passed):
https://testnet.bscscan.com/tx/0xc2d963c9155b23db4c20d9fa02efadac2f99b507565336a3660233763ea1a77c

10- Auto liquidity (passed):
https://testnet.bscscan.com/token/0x4b5c5e4268b30992baa1f011f765a701ecc91f03?a=0x0000000000000000000000000000000000000000

MANUAL TESTING

Centralization - Owner Must Enable Trades

Severity: High / Informational

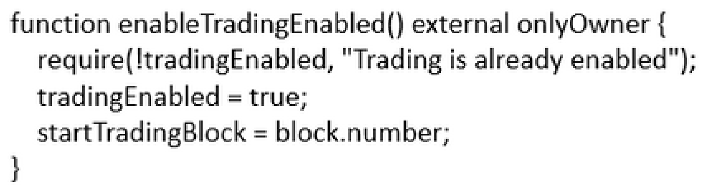
Function: enableTradingEnabled

Lines: 1145

Status: Resolved (Token is owned by safu developer for 14 days after presale)

Overview:

The owner is required to enable trading for investors. If trading remains disabled, token holders will not have the ability to buy, sell, or transfer their tokens.



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

Recommendation:

While the presence of this function is considered a feature rather than a flaw, it is crucial to highlight the centralization risk it inherently poses. To address this issue and ensure the enablement of trades, one possible solution would be to transfer the contract's ownership to a trusted third party, such as a Pinksale Safu developer. This would help mitigate the centralization risk associated with this function.

MANUAL TESTING

Informational – Reward token

Overview:

The current implementation of the contract uses **Doge** as its reward token (**0xbA2aE424d960c26247Dd6c32edC70B295c744C43**). **Doge** is beyond the scope of this audit, and any exploits or issues found in the **Doge** token can have an impact on the rewards system in **RDoge**. This may include consequences such as high gas usage or trade disablement.

Recommendation:

Use a simpler token such as BUSD, USDC etc as reward token in order to reduce overall gas usage and mitigate other potential issues.

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