auditace.tech

AuditAce

FROM INCEPTION TO SUCCESS

Smart Contract Audit

FOR

AI EXCHANGE TOKEN

DATED : 22 Feb, 2024

AUDIT SUMMARY

Project name – AI EXCHANGE TOKEN

Date: 22 Feb, 2024

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/address/0xf47baa67de5204ca64b8b5dd4f4daf4fa87b0367#code

Token Information

Token Name : AI EXCHANGE TOKEN

Token Symbol: AIX

Decimals: 18

Token Supply: 1000000000

Network: Binance smart chain

Token Type: BEP-20

Token Address:
0x4a99C2b605Fc87acEE2dc3f61587c176F47b199F

Checksum:
A2032c616934aeb47e6039f76b20d322

Owner:
0x854a5919db5B5FDD60B748C2B23f9A88841D4c9D
(at time of writing the audit)

Deployer:
0x854a5919db5B5FDD60B748C2B23f9A88841D4c9D

TOKEN OVERVIEW

Fees:
Buy Tax: 25%
Sell Tax: 25%
Transfer Fee: 25%

Fees Privilege: Owner

Ownership: Owned

Minting: None

Max Tx Amount/ Max Wallet Amount: No

Blacklist: No

AUDIT METHODOLOGY

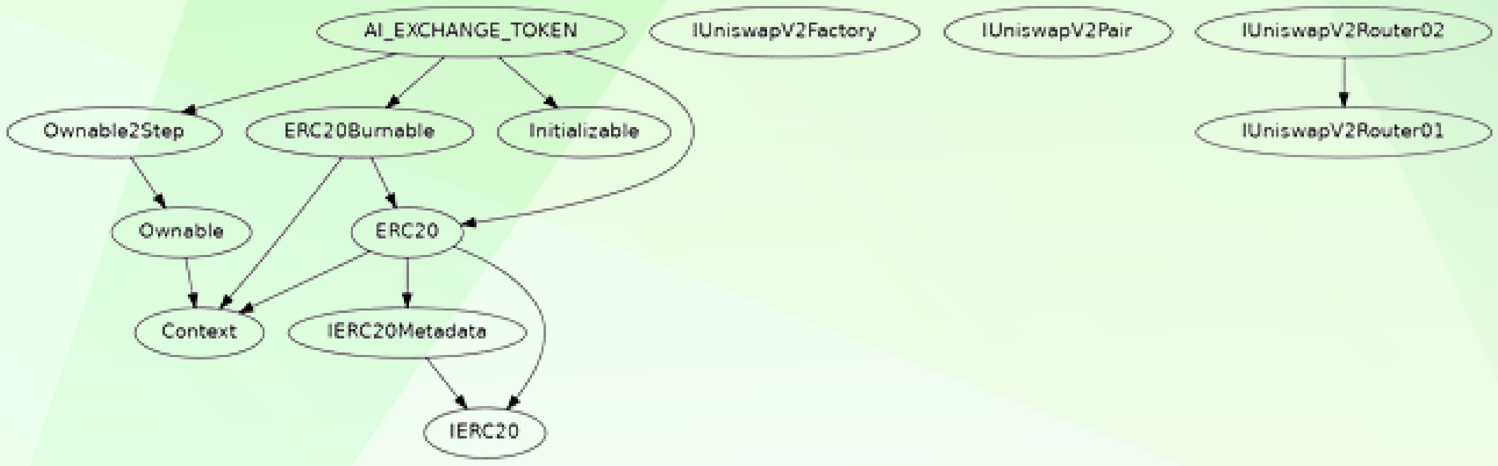
The auditing process will follow a routine as special considerations by Auditace:

- 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

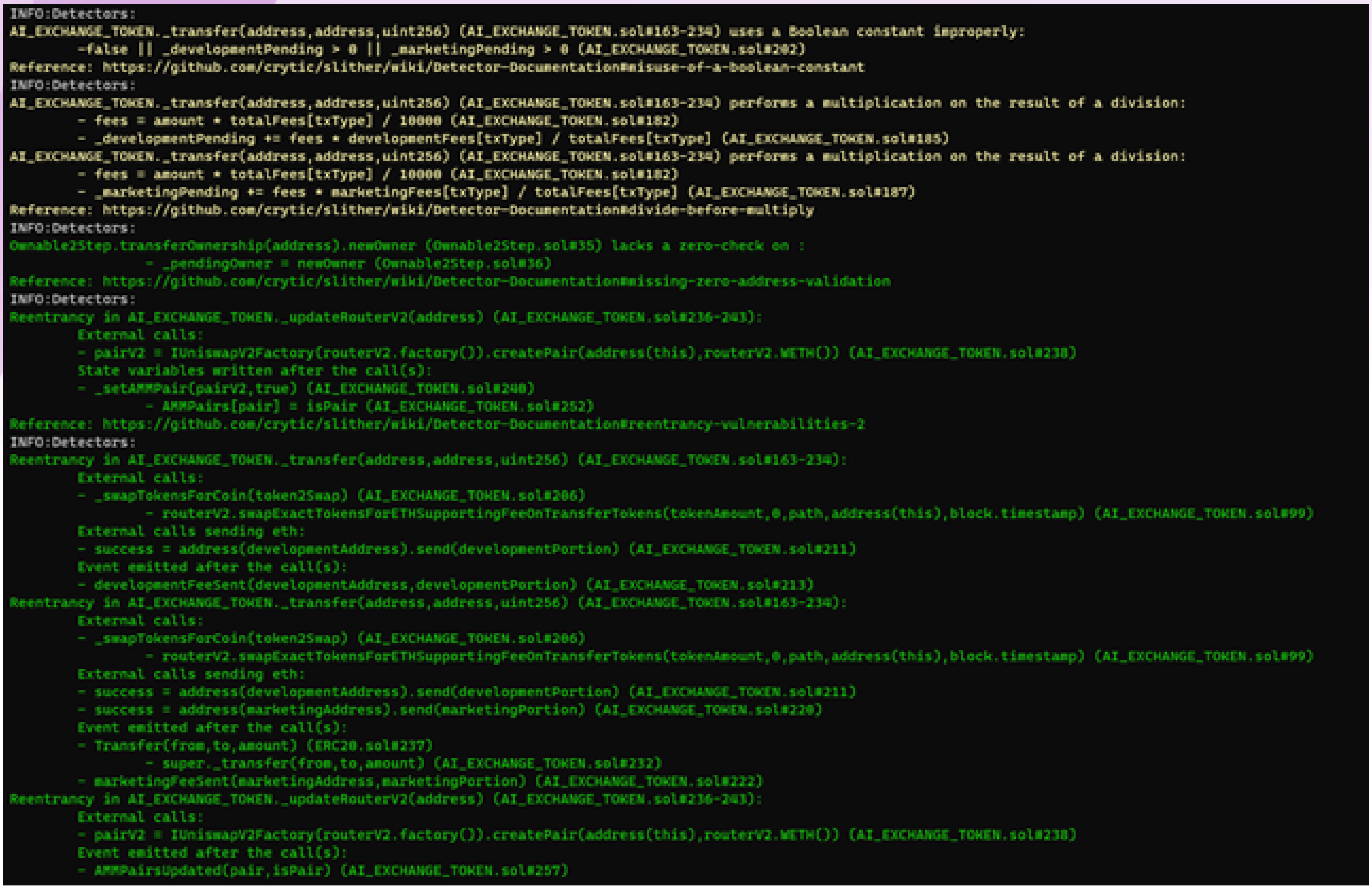
INHERITANCE TREE



STATIC ANALYSIS

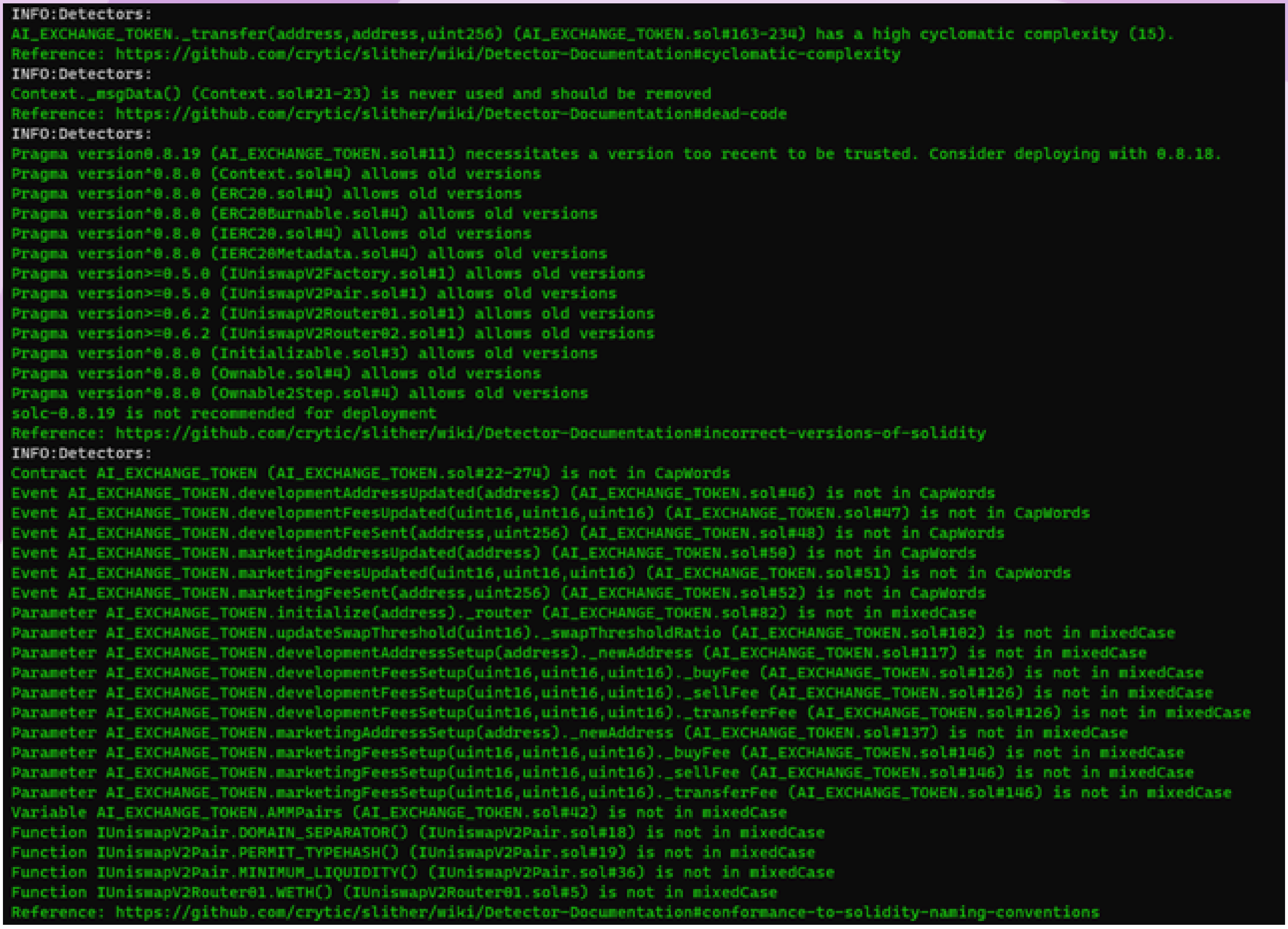
A static analysis of the code was performed using Slither.

No issues were found.

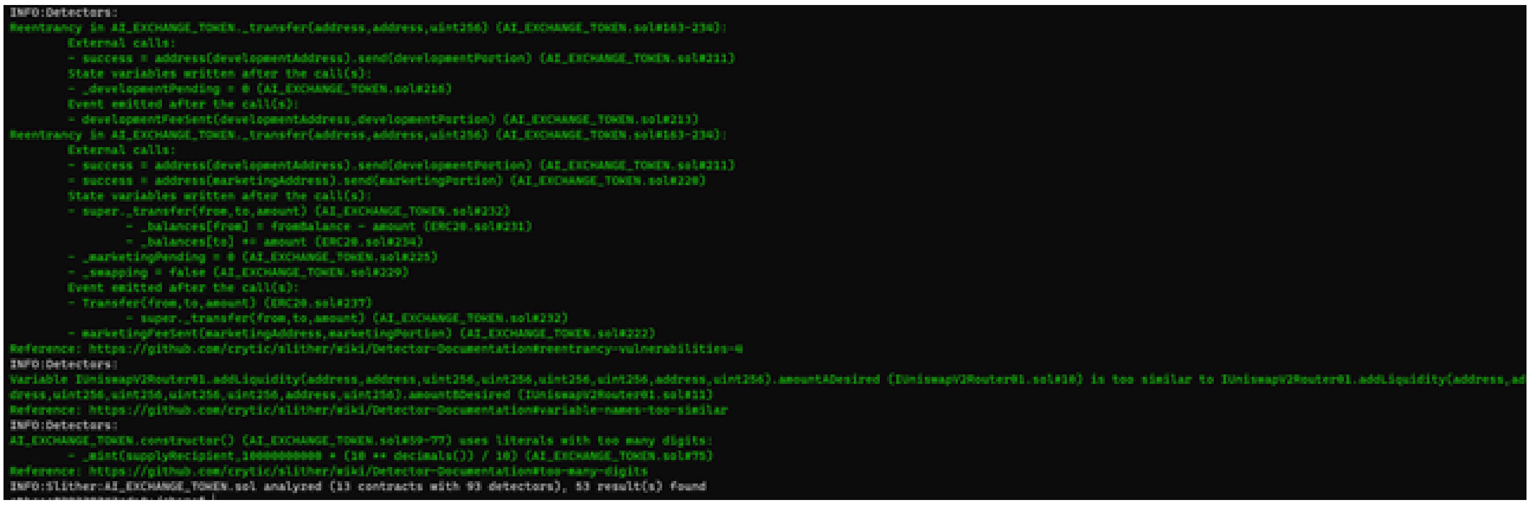


```
INFO:Detectors:
AI_EXCHANGE_TOKEN._transfer(address,address,uint256) (AI_EXCHANGE_TOKEN.sol#163-234) uses a Boolean constant improperly:
        -false || _developmentPending > 0 || _marketingPending > 0 (AI_EXCHANGE_TOKEN.sol#202)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#misuse-of-a-boolean-constant
INFO:Detectors:
AI_EXCHANGE_TOKEN._transfer(address,address,uint256) (AI_EXCHANGE_TOKEN.sol#163-234) performs a multiplication on the result of a division:
        - fees = amount * totalFees[txType] / 10000 (AI_EXCHANGE_TOKEN.sol#182)
        - _developmentPending += fees * developmentFees[txType] / totalFees[txType] (AI_EXCHANGE_TOKEN.sol#185)
AI_EXCHANGE_TOKEN._transfer(address,address,uint256) (AI_EXCHANGE_TOKEN.sol#163-234) performs a multiplication on the result of a division:
        - fees = amount * totalFees[txType] / 10000 (AI_EXCHANGE_TOKEN.sol#182)
        - _marketingPending += fees * marketingFees[txType] / totalFees[txType] (AI_EXCHANGE_TOKEN.sol#187)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#divide-before-multiply
INFO:Detectors:
Ownable2Step.transferOwnership(address).newOwner (Ownable2Step.sol#35) lacks a zero-check on :
                - _pendingOwner = newOwner (Ownable2Step.sol#36)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation
INFO:Detectors:
Reentrancy in AI_EXCHANGE_TOKEN._updateRouterV2(address) (AI_EXCHANGE_TOKEN.sol#236-243):
        External calls:
        - pairV2 = IUniswapV2Factory(routerV2.factory()).createPair(address(this),routerV2.WETH()) (AI_EXCHANGE_TOKEN.sol#238)
        State variables written after the call(s):
        - _setAMMPair(pairV2,true) (AI_EXCHANGE_TOKEN.sol#240)
                - AMMPairs[pair] = isPair (AI_EXCHANGE_TOKEN.sol#252)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2
INFO:Detectors:
Reentrancy in AI_EXCHANGE_TOKEN._transfer(address,address,uint256) (AI_EXCHANGE_TOKEN.sol#163-234):
        External calls:
        - _swapTokensForCoin(token2Swap) (AI_EXCHANGE_TOKEN.sol#206)
                - routerV2.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,address(this),block.timestamp) (AI_EXCHANGE_TOKEN.sol#99)
        External calls sending eth:
        - success = address(developmentAddress).send(developmentPortion) (AI_EXCHANGE_TOKEN.sol#211)
        Event emitted after the call(s):
        - developmentFeeSent(developmentAddress,developmentPortion) (AI_EXCHANGE_TOKEN.sol#213)
Reentrancy in AI_EXCHANGE_TOKEN._transfer(address,address,uint256) (AI_EXCHANGE_TOKEN.sol#163-234):
        External calls:
        - _swapTokensForCoin(token2Swap) (AI_EXCHANGE_TOKEN.sol#206)
                - routerV2.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,address(this),block.timestamp) (AI_EXCHANGE_TOKEN.sol#99)
        External calls sending eth:
        - success = address(developmentAddress).send(developmentPortion) (AI_EXCHANGE_TOKEN.sol#211)
        - success = address(marketingAddress).send(marketingPortion) (AI_EXCHANGE_TOKEN.sol#220)
        Event emitted after the call(s):
        - Transfer(from,to,amount) (ERC20.sol#237)
                - super._transfer(from,to,amount) (AI_EXCHANGE_TOKEN.sol#232)
        - marketingFeeSent(marketingAddress,marketingPortion) (AI_EXCHANGE_TOKEN.sol#222)
Reentrancy in AI_EXCHANGE_TOKEN._updateRouterV2(address) (AI_EXCHANGE_TOKEN.sol#236-243):
        External calls:
        - pairV2 = IUniswapV2Factory(routerV2.factory()).createPair(address(this),routerV2.WETH()) (AI_EXCHANGE_TOKEN.sol#238)
        Event emitted after the call(s):
        - AMMPairsUpdated(pair,isPair) (AI_EXCHANGE_TOKEN.sol#257)
```

STATIC ANALYSIS



```
INFO:Detectors:
AI_EXCHANGE_TOKEN._transfer(address,address,uint256) (AI_EXCHANGE_TOKEN.sol#163-234) has a high cyclomatic complexity (15).
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#cyclomatic-complexity
INFO:Detectors:
Context._msgData() (Context.sol#21-23) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
Pragma version0.8.19 (AI_EXCHANGE_TOKEN.sol#11) necessitates a version too recent to be trusted. Consider deploying with 0.8.18.
Pragma version^0.8.0 (Context.sol#4) allows old versions
Pragma version^0.8.0 (ERC20.sol#4) allows old versions
Pragma version^0.8.0 (ERC20Burnable.sol#4) allows old versions
Pragma version^0.8.0 (IERC20.sol#4) allows old versions
Pragma version^0.8.0 (IERC20Metadata.sol#4) allows old versions
Pragma version>=0.5.0 (IUniswapV2Factory.sol#1) allows old versions
Pragma version>=0.5.0 (IUniswapV2Pair.sol#1) allows old versions
Pragma version>=0.6.2 (IUniswapV2Router01.sol#1) allows old versions
Pragma version>=0.6.2 (IUniswapV2Router02.sol#1) allows old versions
Pragma version^0.8.0 (Initializable.sol#3) allows old versions
Pragma version^0.8.0 (Ownable.sol#4) allows old versions
Pragma version^0.8.0 (Ownable2Step.sol#4) allows old versions
solc-0.8.19 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Contract AI_EXCHANGE_TOKEN (AI_EXCHANGE_TOKEN.sol#22-274) is not in CapWords
Event AI_EXCHANGE_TOKEN.developmentAddressUpdated(address) (AI_EXCHANGE_TOKEN.sol#46) is not in CapWords
Event AI_EXCHANGE_TOKEN.developmentFeesUpdated(uint16,uint16,uint16) (AI_EXCHANGE_TOKEN.sol#47) is not in CapWords
Event AI_EXCHANGE_TOKEN.developmentFeeSent(address,uint256) (AI_EXCHANGE_TOKEN.sol#48) is not in CapWords
Event AI_EXCHANGE_TOKEN.marketingAddressUpdated(address) (AI_EXCHANGE_TOKEN.sol#50) is not in CapWords
Event AI_EXCHANGE_TOKEN.marketingFeesUpdated(uint16,uint16,uint16) (AI_EXCHANGE_TOKEN.sol#51) is not in CapWords
Event AI_EXCHANGE_TOKEN.marketingFeeSent(address,uint256) (AI_EXCHANGE_TOKEN.sol#52) is not in CapWords
Parameter AI_EXCHANGE_TOKEN.initialize(address)._router (AI_EXCHANGE_TOKEN.sol#82) is not in mixedCase
Parameter AI_EXCHANGE_TOKEN.updateSwapThreshold(uint16)._swapThresholdRatio (AI_EXCHANGE_TOKEN.sol#102) is not in mixedCase
Parameter AI_EXCHANGE_TOKEN.developmentAddressSetup(address)._newAddress (AI_EXCHANGE_TOKEN.sol#117) is not in mixedCase
Parameter AI_EXCHANGE_TOKEN.developmentFeesSetup(uint16,uint16,uint16)._buyFee (AI_EXCHANGE_TOKEN.sol#126) is not in mixedCase
Parameter AI_EXCHANGE_TOKEN.developmentFeesSetup(uint16,uint16,uint16)._sellFee (AI_EXCHANGE_TOKEN.sol#126) is not in mixedCase
Parameter AI_EXCHANGE_TOKEN.developmentFeesSetup(uint16,uint16,uint16)._transferFee (AI_EXCHANGE_TOKEN.sol#126) is not in mixedCase
Parameter AI_EXCHANGE_TOKEN.marketingAddressSetup(address)._newAddress (AI_EXCHANGE_TOKEN.sol#137) is not in mixedCase
Parameter AI_EXCHANGE_TOKEN.marketingFeesSetup(uint16,uint16,uint16)._buyFee (AI_EXCHANGE_TOKEN.sol#146) is not in mixedCase
Parameter AI_EXCHANGE_TOKEN.marketingFeesSetup(uint16,uint16,uint16)._sellFee (AI_EXCHANGE_TOKEN.sol#146) is not in mixedCase
Parameter AI_EXCHANGE_TOKEN.marketingFeesSetup(uint16,uint16,uint16)._transferFee (AI_EXCHANGE_TOKEN.sol#146) is not in mixedCase
Variable AI_EXCHANGE_TOKEN.AMMPairs (AI_EXCHANGE_TOKEN.sol#42) is not in mixedCase
Function IUniswapV2Pair.DOMAIN_SEPARATOR() (IUniswapV2Pair.sol#18) is not in mixedCase
Function IUniswapV2Pair.PERMIT_TYPEHASH() (IUniswapV2Pair.sol#19) is not in mixedCase
Function IUniswapV2Pair.MINIMUM_LIQUIDITY() (IUniswapV2Pair.sol#36) is not in mixedCase
Function IUniswapV2Router01.WETH() (IUniswapV2Router01.sol#5) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
```



```
INFO:Detectors:
Reentrancy in AI_EXCHANGE_TOKEN._transfer(address,address,uint256) (AI_EXCHANGE_TOKEN.sol#163-234):
        External calls:
        - success = address(developmentAddress).send(developmentPortion) (AI_EXCHANGE_TOKEN.sol#211)
        State variables written after the call(s):
        - _developmentPending = 0 (AI_EXCHANGE_TOKEN.sol#216)
        Event emitted after the call(s):
        - developmentFeeSent(developmentAddress,developmentPortion) (AI_EXCHANGE_TOKEN.sol#213)
Reentrancy in AI_EXCHANGE_TOKEN._transfer(address,address,uint256) (AI_EXCHANGE_TOKEN.sol#163-234):
        External calls:
        - success = address(developmentAddress).send(developmentPortion) (AI_EXCHANGE_TOKEN.sol#211)
        - success = address(marketingAddress).send(marketingPortion) (AI_EXCHANGE_TOKEN.sol#220)
        State variables written after the call(s):
        - super._transfer(from,to,amount) (AI_EXCHANGE_TOKEN.sol#232)
                - _balances[from] = fromBalance - amount (ERC20.sol#231)
                - _balances[to] += amount (ERC20.sol#234)
        - _marketingPending = 0 (AI_EXCHANGE_TOKEN.sol#225)
        - _swapping = false (AI_EXCHANGE_TOKEN.sol#229)
        Event emitted after the call(s):
        - Transfer(from,to,amount) (ERC20.sol#237)
                - super._transfer(from,to,amount) (AI_EXCHANGE_TOKEN.sol#232)
        - marketingFeeSent(marketingAddress,marketingPortion) (AI_EXCHANGE_TOKEN.sol#222)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-4
INFO:Detectors:
Variable IUniswapV2Router01.addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256).amountADesired (IUniswapV2Router01.sol#10) is too similar to IUniswapV2Router01.addLiquidity(address,ad
dress,uint256,uint256,uint256,uint256,address,uint256).amountBDesired (IUniswapV2Router01.sol#11)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-too-similar
INFO:Detectors:
AI_EXCHANGE_TOKEN.constructor() (AI_EXCHANGE_TOKEN.sol#59-77) uses literals with too many digits:
        - _mint(supplyRecipient,10000000000 * (10 ** decimals()) / 10) (AI_EXCHANGE_TOKEN.sol#75)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits
INFO:Slither:AI_EXCHANGE_TOKEN.sol analyzed (13 contracts with 93 detectors), 53 result(s) found
```

FUNCTIONAL TESTING

1- Approve (passed):
https://testnet.bscscan.com/tx/0xdeeb45fc729c0923c3d806cba25c38f4f52cb095d9f736109dae159de4aeacfc

2- Development Address Setup (passed):
https://testnet.bscscan.com/tx/0xa02dd2a8f6f32831916dee5868df9d60d7f5e5c942572ce9d9d29d191492362a

3- Development Fees Setup (passed):
https://testnet.bscscan.com/tx/0xc714166f743adf4928509854bd97215c22483acf1f2d7f097ef67735882b60bd

4- Marketing Address Setup (passed):
https://testnet.bscscan.com/tx/0x5b23565c8440f85846f9e4d514cf410122b1ce58f6f4f245013efbb0f3ca3c08

5- Marketing Fees Setup (passed):
https://testnet.bscscan.com/tx/0xa3581324e9a2046dd68b0eb4ee5ec65b1ace5d334784085fe70ac2d971eaf2a2

6- Transfer (passed):
https://testnet.bscscan.com/tx/0x9cdc8a7152ce079be035f51a8a7aadbc78f4a719fc212f1ca25a72937b392406

POINTS TO NOTE

- The owner can transfer ownership.
- The owner can renounce ownership.
- The owner can set developmentAddress/marketingAddress.
- The owner can set buy/sell/transfer fees to not more than 25%.
- The owner can exclude address from fees.

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.

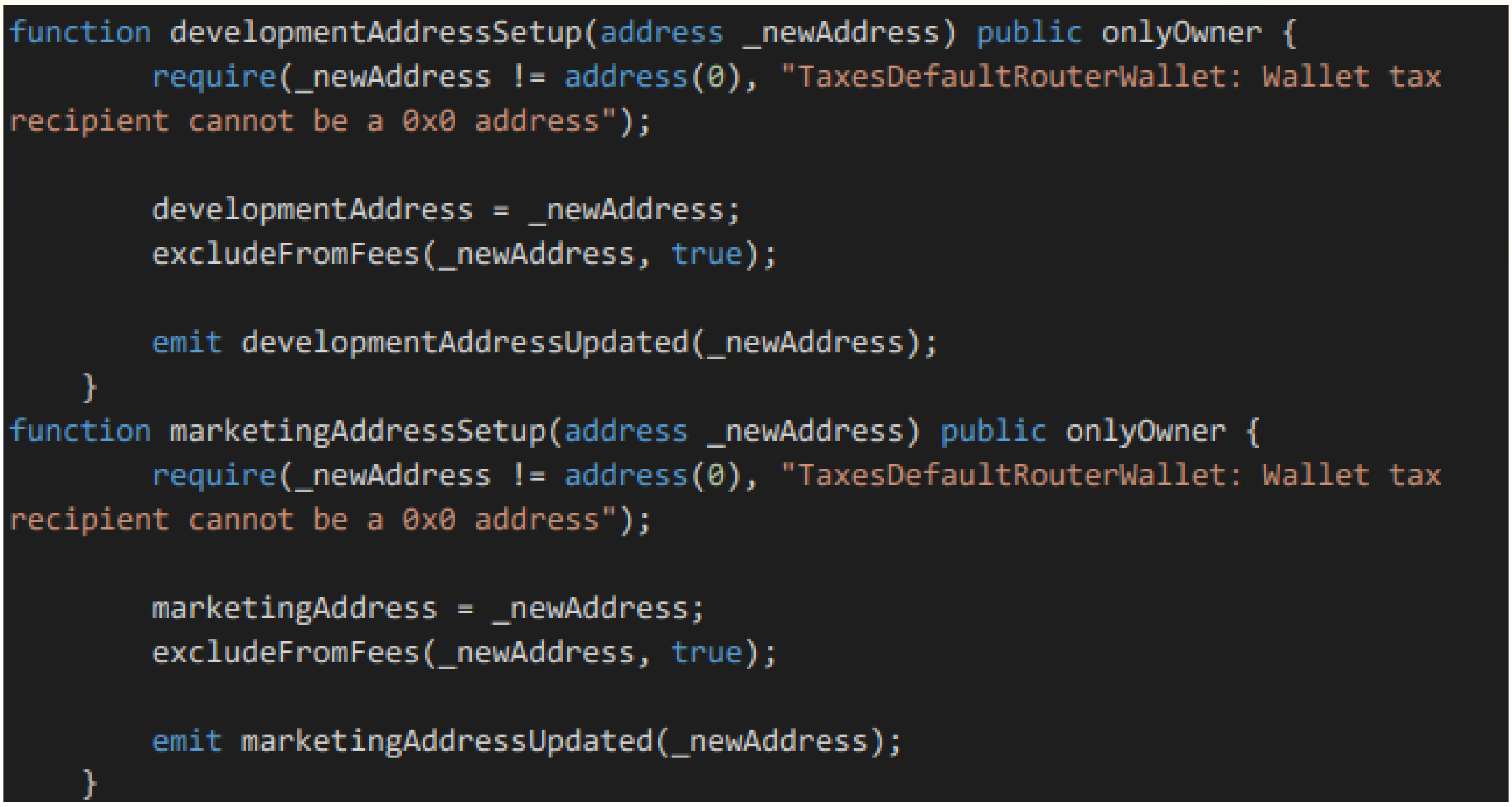
Findings

Severity	Found
🟥 Critical	0
🟫 High-Risk	0
🟧 Medium-Risk	1
🟨 Low-Risk	0
🟦 Gas Optimization / Suggestions	0

MANUAL TESTING

Centralization – Missing Require Check
Severity: Medium
Function: developmentAddressSetup/marketingAddressSetup
Status: Open

Overview:
The owner can set any arbitrary address excluding zero address as this is not recommended because if the owner will set the address to the contract address, then the Eth will not be sent to that address and the transaction will fail and this will lead to a potential honeypot in the contract.



```
function developmentAddressSetup(address _newAddress) public onlyOwner {
        require(_newAddress != address(0), "TaxesDefaultRouterWallet: Wallet tax
recipient cannot be a 0x0 address");

        developmentAddress = _newAddress;
        excludeFromFees(_newAddress, true);

        emit developmentAddressUpdated(_newAddress);
    }
function marketingAddressSetup(address _newAddress) public onlyOwner {
        require(_newAddress != address(0), "TaxesDefaultRouterWallet: Wallet tax
recipient cannot be a 0x0 address");

        marketingAddress = _newAddress;
        excludeFromFees(_newAddress, true);

        emit marketingAddressUpdated(_newAddress);
    }
```

Suggestion: It is recommended that the address should not be able to set as a contract address.

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