AuditAce
FROM INCEPTION TO SUCCESS

Smart Contract Audit

FOR

X-AI

DATED : 26 August 23'

MANUAL TESTING

Centralization – Enabling Trades
Severity: High
function: enableTrading
Status: Open
Overview:
The launch function permits only the contract owner to activate trading capabilities. Until this function is executed, no investors can buy, sell, or transfer their tokens. This places a high degree of control and centralization in the hands of the contract owner.

```
function enableTrading() external onlyOwner {
require(tradingEnabled == false, "Trading is already enabled");
tradingEnabled = true;
  }
```

Suggestion
To reduce centralization and potential manipulation, consider one of the following approaches:
1.Automatically enable trading after a specified condition, such as the completion of a presale, is met.
2.If manual activation is still desired, consider transferring the ownership of the contract to a trustworthy, third-party entity like a certified "PinkSale Safu" developer. This can provide investors with more confidence in the eventual activation of trading capabilities, mitigating concerns of potential bad faith actions by the original owner.

AUDIT SUMMARY

Project name – X-AI

Date: 24 August 2023

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

Audit Status: **Passed With High Risk**

Issues Found

Status	Critical	High	Medium	Low	Suggestion
Open	0	1	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/0xBdCC83b975feb93060A255beb3AA0C05A046f502

Token Information

Token Address :
0xF30153d8A69Afde3f7e41E009480a4D5f70fBb44

Name: X-AI

Symbol: XAI

Decimals: 18

Network: Binance smart chain

Token Type: BEP20

Owner: 0x6B71BC620DDc1EBFEAfE7d4D010bd1a858Fe4FF4

Deployer: 0x6B71BC620DDc1EBFEAfE7d4D010bd1a858Fe4FF4

Token Supply: 3,000,000

Checksum:
481a8c4dcb4665feeac96a69412e38db5afd3ae8

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/0xBdCC83b975feb93060A255beb3AA0C05A046f502

TOKEN OVERVIEW

buy fee: 0-20%

Sell fee: 0-20%

transfer fee: 0%

Fee Privilege: Owner

Ownership: Owned

Minting: None

Max Tx: No

Blacklist: No

Other Privileges:

- Modifying fees
- Enabling trades
- Initial distribution of the tokens

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

- [x] Return values of low-level calls
- [x] Private modifier
- [x] Multiple Sends
- [x] Using Suicide
- [x] Gas Limitand Loops
- [x] Address hardcoded
- [x] Exception Disorder
- [x] Using inline assembly
- [x] Divide before multiply
- [x] Missing Zero Address Validation
- [x] Compiler version not fixed
- [x] **Gasless Send**
- [x] Using block.timestamp
- [x] Re-entrancy
- [x] Tautology or contradiction
- [x] Timestamp Dependence
- [x] Revert/require functions
- [x] Use of tx.origin
- [x] Integer overflow/underflow
- [x] Dangerous strict equalities
- [x] Using SHA3
- [x] Using throw

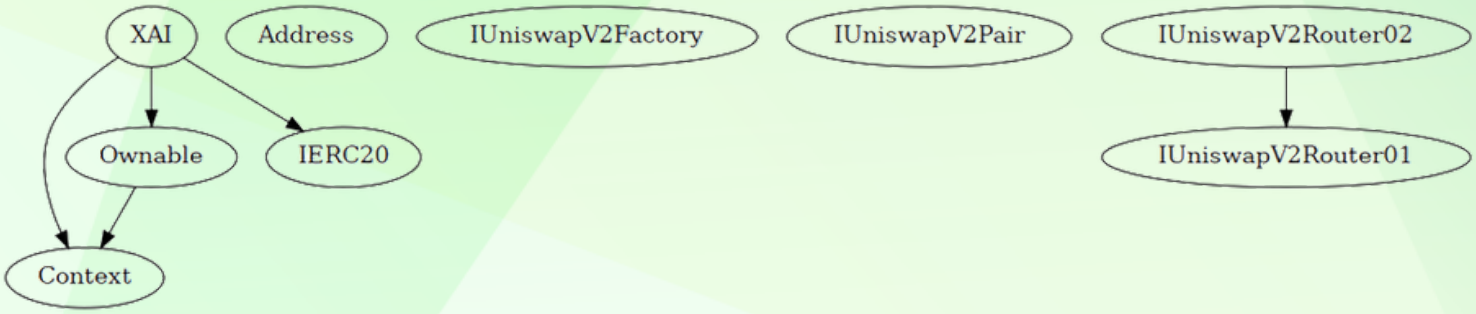
CLASSIFICATION OF RISK

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	0
Low-Risk	0
Gas Optimization / Suggestions	1

INHERITANCE TREE



POINTS TO NOTE

- Owner is able to adjust buy/sell fees within 0-20% (0% transfer fee)

- Owner is able to blacklist an arbitrary wallet
- Owner is able to disable trades
- Owner is not able to mint new tokens
- Owner is not able to set maximum wallet and maximum buy/sell/transfer limits

- Owner must enable trades manually

STATIC ANALYSIS

```
Variable XAI._transferStandard(address,address,uint256).rTransferAmount (contracts/Token.sol#1089) is too similar to XAI._getTValues(uint256).tTransferAmount (contracts/Token.sol#821)
Variable XAI._transferBothExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1157) is too similar to XAI._transferBothExcluded(address,address,uint256).tTransferAmount (c
ontracts/Token.sol#1159)
Variable XAI._getRValues(uint256,uint256,uint256,uint256,uint256).rTransferAmount (contracts/Token.sol#836) is too similar to XAI._getValues(uint256).tTransferAmount (contracts/Token.sol#792
)
Variable XAI._transferStandard(address,address,uint256).rTransferAmount (contracts/Token.sol#1089) is too similar to XAI._transferToExcluded(address,address,uint256).tTransferAmount (contrac
ts/Token.sol#1113)
Variable XAI._getRValues(uint256,uint256,uint256,uint256,uint256).rTransferAmount (contracts/Token.sol#836) is too similar to XAI._getTValues(uint256).tTransferAmount (contracts/Token.sol#82
1)
Variable XAI._getRValues(uint256,uint256,uint256,uint256,uint256).rTransferAmount (contracts/Token.sol#836) is too similar to XAI._transferToExcluded(address,address,uint256).tTransferAmount
 (contracts/Token.sol#1113)
Variable XAI._transferFromExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1134) is too similar to XAI._transferStandard(address,address,uint256).tTransferAmount (contr
acts/Token.sol#1091)
Variable XAI.reflectionFromToken(uint256,bool).rTransferAmount (contracts/Token.sol#728) is too similar to XAI._getValues(uint256).tTransferAmount (contracts/Token.sol#792)
Variable XAI._transferFromExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1134) is too similar to XAI._transferBothExcluded(address,address,uint256).tTransferAmount (c
ontracts/Token.sol#1159)
Variable XAI.reflectionFromToken(uint256,bool).rTransferAmount (contracts/Token.sol#728) is too similar to XAI._getTValues(uint256).tTransferAmount (contracts/Token.sol#821)
Variable XAI._transferToExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1111) is too similar to XAI._getTValues(uint256).tTransferAmount (contracts/Token.sol#821)
Variable XAI._transferToExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1111) is too similar to XAI._getValues(uint256).tTransferAmount (contracts/Token.sol#792)
Variable XAI._transferToExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1111) is too similar to XAI._transferToExcluded(address,address,uint256).tTransferAmount (contr
acts/Token.sol#1113)
Variable XAI.reflectionFromToken(uint256,bool).rTransferAmount (contracts/Token.sol#728) is too similar to XAI._transferToExcluded(address,address,uint256).tTransferAmount (contracts/Token.s
ol#1113)
Variable XAI._transferFromExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1134) is too similar to XAI._transferFromExcluded(address,address,uint256).tTransferAmount (c
ontracts/Token.sol#1136)
Variable XAI._getValues(uint256).rTransferAmount (contracts/Token.sol#797) is too similar to XAI._transferToExcluded(address,address,uint256).tTransferAmount (contracts/Token.sol#1113)
Variable XAI._transferBothExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1157) is too similar to XAI._getValues(uint256).tTransferAmount (contracts/Token.sol#792)
Variable XAI._transferBothExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1157) is too similar to XAI._getTValues(uint256).tTransferAmount (contracts/Token.sol#821)
Variable XAI._getValues(uint256).rTransferAmount (contracts/Token.sol#797) is too similar to XAI._getTValues(uint256).tTransferAmount (contracts/Token.sol#821)
Variable XAI._getValues(uint256).rTransferAmount (contracts/Token.sol#797) is too similar to XAI._getValues(uint256).tTransferAmount (contracts/Token.sol#792)
Variable XAI._transferBothExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1157) is too similar to XAI._transferToExcluded(address,address,uint256).tTransferAmount (con
tracts/Token.sol#1113)
Variable XAI._transferStandard(address,address,uint256).rTransferAmount (contracts/Token.sol#1089) is too similar to XAI._transferStandard(address,address,uint256).tTransferAmount (contracts
/Token.sol#1091)
Variable XAI._getRValues(uint256,uint256,uint256,uint256,uint256).rTransferAmount (contracts/Token.sol#836) is too similar to XAI._transferStandard(address,address,uint256).tTransferAmount (
contracts/Token.sol#1091)
Variable XAI._transferFromExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1134) is too similar to XAI._transferToExcluded(address,address,uint256).tTransferAmount (con
tracts/Token.sol#1113)
Variable XAI._transferFromExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1134) is too similar to XAI._getValues(uint256).tTransferAmount (contracts/Token.sol#792)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-too-similar
INFO:Detectors:
Loop condition `i < _excluded.length` (contracts/Token.sol#848) should use cached array length instead of referencing `length` member of the storage array.
 Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#cache-array-length
INFO:Detectors:
XAI.DEAD (contracts/Token.sol#552) should be constant
XAI._decimals (contracts/Token.sol#527) should be constant
XAI._name (contracts/Token.sol#525) should be constant
XAI._symbol (contracts/Token.sol#526) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant
INFO:Detectors:
XAI._tTotal (contracts/Token.sol#530) should be immutable
XAI.uniswapV2Pair (contracts/Token.sol#555) should be immutable
XAI.uniswapV2Router (contracts/Token.sol#554) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
INFO:Slither:./contracts/Token.sol analyzed (9 contracts with 88 detectors), 99 result(s) found
```

Result => A static analysis of contract's source code has been performed using slither,

No major issues were found in the output

CONTRACT ASSESMENT

Contract	Type	Bases		
└	**Function Name**	**Visibility**	**Mutability**	**Modifiers**
Context	Implementation			
└	_msgSender	Internal 🔒		
└	_msgData	Internal 🔒		
Ownable	Implementation	Context		
└	<Constructor>	Public ❗	🛑	NO ❗
└	owner	Public ❗		NO ❗
└	renounceOwnership	Public ❗	🛑	onlyOwner
└	transferOwnership	Public ❗	🛑	onlyOwner
IERC20	Interface			
└	totalSupply	External ❗		NO ❗
└	balanceOf	External ❗		NO ❗
└	transfer	External ❗	🛑	NO ❗
└	allowance	External ❗		NO ❗
└	approve	External ❗	🛑	NO ❗
└	transferFrom	External ❗	🛑	NO ❗
Address	Library			
└	isContract	Internal 🔒		
└	sendValue	Internal 🔒	🛑	
└	functionCall	Internal 🔒	🛑	
└	functionCall	Internal 🔒	🛑	
└	functionCallWithValue	Internal 🔒	🛑	
└	functionCallWithValue	Internal 🔒	🛑	
└	_functionCallWithValue	Private 🔐	🛑	

CONTRACT ASSESMENT

IUniswapV2Factory	Interface			
└	feeTo	External ❗		NO❗
└	feeToSetter	External ❗		NO❗
└	getPair	External ❗		NO❗
└	allPairs	External ❗		NO❗
└	allPairsLength	External ❗		NO❗
└	createPair	External ❗	🛑	NO❗
└	setFeeTo	External ❗	🛑	NO❗
└	setFeeToSetter	External ❗	🛑	NO❗
IUniswapV2Pair	Interface			
└	name	External ❗		NO❗
└	symbol	External ❗		NO❗
└	decimals	External ❗		NO❗
└	totalSupply	External ❗		NO❗
└	balanceOf	External ❗		NO❗
└	allowance	External ❗		NO❗
└	approve	External ❗	🛑	NO❗
└	transfer	External ❗	🛑	NO❗
└	transferFrom	External ❗	🛑	NO❗
└	DOMAIN_SEPARATOR	External ❗		NO❗
└	PERMIT_TYPEHASH	External ❗		NO❗
└	nonces	External ❗		NO❗
└	permit	External ❗	🛑	NO❗
└	MINIMUM_LIQUIDITY	External ❗		NO❗
└	factory	External ❗		NO❗
└	token0	External ❗		NO❗
└	token1	External ❗		NO❗
└	getReserves	External ❗		NO❗
└	price0CumulativeLast	External ❗		NO❗
└	price1CumulativeLast	External ❗		NO❗
└	kLast	External ❗		NO❗
└	burn	External ❗	🛑	NO❗
└	swap	External ❗	🛑	NO❗
└	skim	External ❗	🛑	NO❗
└	sync	External ❗	🛑	NO❗
└	initialize	External ❗	🛑	NO❗

CONTRACT ASSESMENT

IUniswapV2Router01	Interface			
└	factory	External ❗		NO ❗
└	WETH	External ❗		NO ❗
└	addLiquidity	External ❗	🛑	NO ❗
└	addLiquidityETH	External ❗	💵	NO ❗
└	removeLiquidity	External ❗	🛑	NO ❗
└	removeLiquidityETH	External ❗	🛑	NO ❗
└	removeLiquidityWithPermit	External ❗	🛑	NO ❗
└	removeLiquidityETHWithPermit	External ❗	🛑	NO ❗
└	swapExactTokensForTokens	External ❗	🛑	NO ❗
└	swapTokensForExactTokens	External ❗	🛑	NO ❗
└	swapExactETHForTokens	External ❗	💵	NO ❗
└	swapTokensForExactETH	External ❗	🛑	NO ❗
└	swapExactTokensForETH	External ❗	🛑	NO ❗
└	swapETHForExactTokens	External ❗	💵	NO ❗
└	quote	External ❗		NO ❗
└	getAmountOut	External ❗		NO ❗
└	getAmountIn	External ❗		NO ❗
└	getAmountsOut	External ❗		NO ❗
└	getAmountsIn	External ❗		NO ❗
IUniswapV2Router02	Interface	IUniswapV2Router01		
└	removeLiquidityETHSupportingFeeOnTransferTokens	External ❗	🛑
NO ❗
└	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens
External ❗	🛑	NO ❗
└	swapExactTokensForTokensSupportingFeeOnTransferTokens	External
❗	🛑	NO ❗
└	swapExactETHForTokensSupportingFeeOnTransferTokens	External ❗
💵	NO ❗
└	swapExactTokensForETHSupportingFeeOnTransferTokens	External ❗
🛑	NO ❗

CONTRACT ASSESMENT

XAI	Implementation	Context, IERC20, Ownable		
└	<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 ❗
└	isExcludedFromReward	Public ❗		NO ❗
└	totalReflectionDistributed	Public ❗		NO ❗
└	reflectionFromToken	Public ❗		NO ❗
└	tokenFromReflection	Public ❗		NO ❗
└	excludeFromReward	Public ❗	🛑	onlyOwner
└	includeInReward	External ❗	🛑	onlyOwner
└	<Receive Ether>	External ❗	💵	NO ❗
└	claimStuckTokens	External ❗	🛑	onlyOwner
└	_reflectFee	Private 🔐	🛑	
└	_getValues	Private 🔐		
└	_getTValues	Private 🔐		
└	_getRValues	Private 🔐		
└	_getRate	Private 🔐		
└	_getCurrentSupply	Private 🔐		
└	_takeLiquidity	Private 🔐	🛑	
└	_takeMarketing	Private 🔐	🛑	
└	calculateTaxFee	Private 🔐		
└	calculateLiquidityFee	Private 🔐		
└	calculateMarketingFee	Private 🔐		
└	removeAllFee	Private 🔐	🛑	
└	setBuyFee	Private 🔐	🛑	
└	setSellFee	Private 🔐	🛑	
└	isExcludedFromFee	Public ❗		NO ❗
└	_approve	Private 🔐	🛑	
└	enableTrading	External ❗	🛑	onlyOwner

CONTRACT ASSESMENT

└	_transfer	Private 🔐	🛑	
└	swapAndLiquify	Private 🔐	🛑	
└	swapAndSendMarketing	Private 🔐	🛑	
└	setSwapTokensAtAmount	External ❗	🛑	onlyOwner
└	setSwapEnabled	External ❗	🛑	onlyOwner
└	_tokenTransfer	Private 🔐	🛑	
└	_transferStandard	Private 🔐	🛑	
└	_transferToExcluded	Private 🔐	🛑	
└	_transferFromExcluded	Private 🔐	🛑	
└	_transferBothExcluded	Private 🔐	🛑	
└	excludeFromFees	External ❗	🛑	onlyOwner
└	changeMarketingWallet	External ❗	🛑	onlyOwner
└	setBuyFeePercentages	External ❗	🛑	onlyOwner
└	setSellFeePercentages	External ❗	🛑	onlyOwner

Legend

Symbol	Meaning
🛑	Function can modify state
💵	Function is payable

FUNCTIONAL TESTING

1- Adding liquidity (passed):
https://testnet.bscscan.com/tx/0x277cc2d79a070de2bc153a5c2c2304cba95dac87067b73695d8bfa01e361567b

2- Buying when excluded (0% tax) (passed):
https://testnet.bscscan.com/tx/0xbf674210ca5c7771399c1150f03f8facbf62ff42610be883b03eccdf52ef0e37

3- Selling when excluded (0% tax) (passed):
https://testnet.bscscan.com/tx/0xa0db1b353ae9dcbe60afd5e5a58538d5901e69c7a5a410b7542fdeedc40f76e4

4- Transferring when excluded from fees (0% tax) (passed):
https://testnet.bscscan.com/tx/0x3c78e287b7f814889071e9ab9397913e15ff7919e66735ae740e4696e943a3b6

5- Buying when not excluded from fees (tax 0-20%) (passed):
https://testnet.bscscan.com/tx/0x51c8f3001de1f43459ed8af068a25ba623273deb6bd1ba330f373198f93966b4

6- Selling when not excluded from fees (tax 0-20%) (passed):
https://testnet.bscscan.com/tx/0xc9ac35472972698431708 4addd2716448c073da9857a09fdf98335157526a53e

7- Transferring when not excluded from fees (0% tax) (passed):
https://testnet.bscscan.com/tx/0x182d4af4a1ecd4bcad0d816857b34e03c38eb68221d97d0893845d2050ec1bdb

8- Internal swap (BNB set to marketing wallet | Auto-liquidity)(passed):
https://testnet.bscscan.com/tx/0xc9ac35472972698431708 4addd2716448c073da9857a09fdf98335157526a53e

MANUAL TESTING

Centralization – Enabling Trades
Severity: High
function: enableTrading
Status: Open
Overview:
The launch function permits only the contract owner to activate trading capabilities. Until this function is executed, no investors can buy, sell, or transfer their tokens. This places a high degree of control and centralization in the hands of the contract owner.

```
function enableTrading() external onlyOwner {
require(tradingEnabled == false, "Trading is already enabled");
tradingEnabled = true;
  }
```

Suggestion
To reduce centralization and potential manipulation, consider one of the following approaches:
1.Automatically enable trading after a specified condition, such as the completion of a presale, is met.
2.If manual activation is still desired, consider transferring the ownership of the contract to a trustworthy, third-party entity like a certified "PinkSale Safu" developer. This can provide investors with more confidence in the eventual activation of trading capabilities, mitigating concerns of potential bad faith actions by the original owner.

MANUAL TESTING

Centralization – Excessive Fees
Severity: Informational
function: setTaxFee – setBuybackFee - setMarketingFee
Status: Open
Overview:
Owner is able to set up to 20% tax on buy/sell

```
function setBuyFeePercentages(
uint256 _taxFeeonBuy,
uint256 _liquidityFeeonBuy,
uint256 _marketingFeeonBuy
  ) external onlyOwner {
taxFeeonBuy = _taxFeeonBuy;
liquidityFeeonBuy = _liquidityFeeonBuy;
marketingFeeonBuy = _marketingFeeonBuy;
totalBuyFees = _taxFeeonBuy + _liquidityFeeonBuy + _marketingFeeonBuy;
require(totalBuyFees <= 20, "Buy fees cannot be greater than 20%");
emit BuyFeesChanged(taxFeeonBuy, liquidityFeeonBuy, marketingFeeonBuy);
  }

function setSellFeePercentages(
uint256 _taxFeeonSell,
uint256 _liquidityFeeonSell,
uint256 _marketingFeeonSell
  ) external onlyOwner {
taxFeeonSell = _taxFeeonSell;
liquidityFeeonSell = _liquidityFeeonSell;
marketingFeeonSell = _marketingFeeonSell;
totalSellFees =
_taxFeeonSell +
_liquidityFeeonSell +
_marketingFeeonSell;
require(totalSellFees <= 20, "Sell fees cannot be greater than 20%");
emit SellFeesChanged(
   taxFeeonSell,
liquidityFeeonSell,
marketingFeeonSell
);
  }
```

Suggestion
Keep fees within 0-10%

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 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