expelee

Building the Futuristic Blockchain Ecosystem

SECURITY AUDIT REPORT

xAIWOJAK

TOKEN OVERVIEW

Risk Findings

Severity	Found
🔴 High	2
🟠 Medium	0
🟡 Low	0
🔵 Informational	0

Centralization Risks

Owner Privileges	Description
🟢 Can Owner Set Taxes >25% ?	Not Detected
🔴 Owner needs to enable trading ?	Yes, owner needs to enable trades
🟢 Can Owner Disable Trades ?	Not Detected
🟢 Can Owner Mint ?	Not Detected
🟢 Can Owner Blacklist ?	Not Detected
🟢 Can Owner set Max Wallet amount ?	Not Detected
🟢 Can Owner Set Max TX amount ?	Not Detected

TABLE OF CONTENTS

OVERVIEW

The Expelee team has performed a line-by-line manual analysis and automated review of the smart contract. The smart contract was analysed mainly for common smart contract vulnerabilities, exploits, and manipulation hacks. According to the smart contract audit:

Audit Result	**Passed With High Risk**
KYC Verification	**-**
Audit Date	**24 July 2023**

CONTRACT DETAILS

Token Name: xAI WOJAK

Symbol: xAI WOJAK

Network: Ethereum

Language: Solidity

Contract Address:
0x3AA21317FbF59D9F3aDBE5ED8cDd5861B9fd29D6

Total Supply: 1,000,000,000

Owner's Wallet:
0x8F53C3d9e8f2ecE5710b92AdDcdA3F953f36e26B

Deployer's Wallet:
0x8F53C3d9e8f2ecE5710b92AdDcdA3F953f36e26B

Testnet.
https://testnet.bscscan.com/token/0xd05b578E7fa370515A
D0366E475a7630d61FB389

AUDIT METHODOLOGY

Audit Details

Our comprehensive audit report provides a full overview of the audited system's architecture, smart contract codebase, and details on any vulnerabilities found within the system.

Audit Goals

The audit goal is to ensure that the project is built to protect investors and users, preventing potentially catastrophic vulnerabilities after launch, that lead to scams and rugpulls.

Code Quality

Our analysis includes both automatic tests and manual code analysis for the following aspects:

- Exploits
- Back-doors
- Vulnerability
- Accuracy
- Readability

Tools

- DE
- Open Zeppelin
- Code Analyzer
- Solidity Code
- Compiler
- Hardhat

VULNERABILITY CHECKS

Design Logic	Passed
Compiler warnings	Passed
Private user data leaks	Passed
Timestamps dependence	Passed
Integer overflow and underflow	Passed
Race conditions & reentrancy. Cross-function race conditions	Passed
Possible delays in data delivery	Passed
Oracle calls	Passed
Front Running	Passed
DoS with Revert	Passed
DoS with block gas limit	Passed
Methods execution permissions	Passed
Economy model	Passed
Impact of the exchange rate on the logic	Passed
Malicious event log	Passed
Scoping and declarations	Passed
Uninitialized storage pointers	Passed
Arithmetic accuracy	Passed
Cross-function race conditions	Passed
Safe Zepplin module	Passed

RISK CLASSIFICATION

When performing smart contract audits, our specialists look for known vulnerabilities as well as logical and acces control issues within the code. The exploitation of these issues by malicious actors may cause serious financial damage to projects that failed to get an audit in time. We categorize these vulnerabilities by the following levels:

High Risk

Issues on this level are critical to the smart contract's performance/functionality and should be fixed before moving to a live environment.

Medium Risk

Issues on this level are critical to the smart contract's performance/functionality and should be fixed before moving to a live environment.

Low Risk

Issues on this level are minor details and warning that can remain unfixed.

Informational

Issues on this level are minor details and warning that can remain unfixed.

INHERITANCE TREES
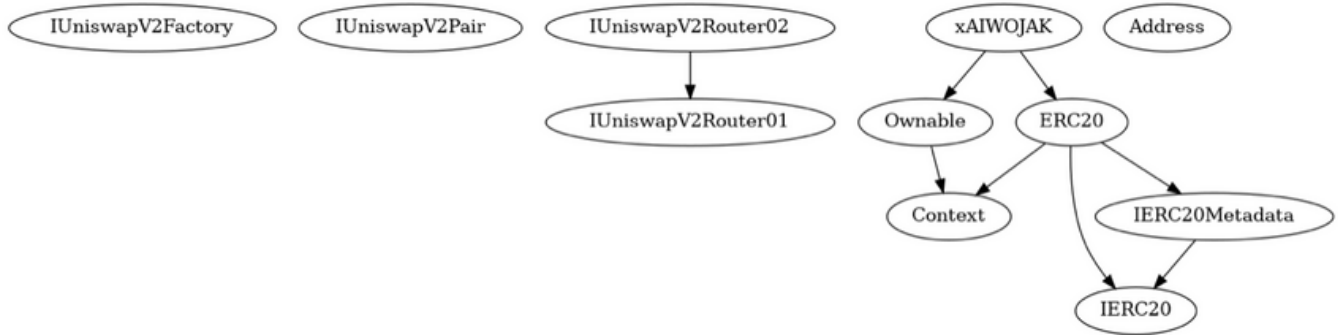
FUNCTION DETAILS

| Contract	Type	Bases		
└	**Function Name**	**Visibility**	**Mutability**	**Modifiers**
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 ❗	

FUNCTION DETAILS

└	mint	External ❗	🔴	NO ❗
└	burn	External ❗	🔴	NO ❗
└	swap	External ❗	🔴	NO ❗
└	skim	External ❗	🔴	NO ❗
└	sync	External ❗	🔴	NO ❗
└	initialize	External ❗	🔴	NO ❗
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 ❗
IERC20	Interface			

FUNCTION DETAILS

└	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 ❗
Address	Library			
└	isContract	Internal 🔒		
└	sendValue	Internal 🔒	🛑	
└	functionCall	Internal 🔒	🛑	
└	functionCall	Internal 🔒	🛑	
└	functionCallWithValue	Internal 🔒	🛑	
└	functionCallWithValue	Internal 🔒	🛑	
└	functionStaticCall	Internal 🔒		
└	functionStaticCall	Internal 🔒		
└	functionDelegateCall	Internal 🔒	🛑	
└	functionDelegateCall	Internal 🔒	🛑	
└	verifyCallResultFromTarget	Internal 🔒		
└	verifyCallResult	Internal 🔒		
└	_revert	Private 🔐		
Context	Implementation			
└	_msgSender	Internal 🔒		
└	_msgData	Internal 🔒		
Ownable	Implementation	Context		
└	<Constructor>	Public ❗	🛑	NO ❗
└	owner	Public ❗			NO ❗
└	renounceOwnership	Public ❗	🛑	onlyOwner
└	transferOwnership	Public ❗	🛑	onlyOwner
ERC20	Implementation	Context, IERC20, IERC20Metadata		

FUNCTION DETAILS

└	<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 🔒	🛑	
└	_mint	Internal 🔒	🛑	
└	_burn	Internal 🔒	🛑	
└	_approve	Internal 🔒	🛑	
└	_beforeTokenTransfer	Internal 🔒	🛑	
└	_afterTokenTransfer	Internal 🔒	🛑	
xAIWOJAK	Implementation	ERC20, Ownable		
└	<Constructor>	Public ❗	🛑	ERC20
└	<Receive Ether>	External ❗	💲	NO ❗
└	claimStuckTokens	External ❗	🛑	onlyOwner
└	excludeFromFees	External ❗	🛑	onlyOwner
└	isExcludedFromFees	Public ❗		NO ❗
└	changeMarketingWallet	External ❗	🛑	onlyOwner
└	enableTrading	External ❗	🛑	onlyOwner
└	_transfer	Internal 🔒	🛑	
└	setSwapEnabled	External ❗	🛑	onlyOwner
└	setSwapTokensAtAmount	External ❗	🛑	onlyOwner
└	swapAndLiquify	Private 🔐	🛑	
└	swapAndSendMarketing	Private 🔐	🛑	

Legend
Symbol	Meaning
🛑	Function can modify state

FUNCTION DETAILS

| | 💵 | Function is payable |

TESTNET VERSION

Adding Liquidity ✅
Tx:
https://testnet.bscscan.com/tx/0xc9a09f258cc86cdb3ab8415e573714fb2bd9c9f4d3a07b3f45dcd2ccbb570ecf

===

Buying when excluded from fees ✅
Tx (2% tax):
https://testnet.bscscan.com/tx/0x29d0801a2504aeddb0dc598ccf378c77e7fd64545c208d1b93948f4597740f4b

===

Selling when excluded from fees ✅
Tx (2% tax):
https://testnet.bscscan.com/tx/0x51a0af1bad02df85b039085c1fbcf7377ba3faaf5a58c034eddfd9ddc0a4532e

===

Transferring when excluded from fees ✅
Tx (2% tax):
https://testnet.bscscan.com/tx/0x035a45d13e959cdb611f75a47ce4b8bae54e65cb85d1d3922734ae699f32e7eb

===

Buying ✅
Tx (2% tax):
https://testnet.bscscan.com/tx/0x31b07af6483b193f6f7dd869dadc75706a3509011459c7270af059cb76f12b59

TESTNET VERSION

Selling ✅
Tx (2% tax):
https://testnet.bscscan.com/tx/0x86b375b7ca5ba2240627faffc122a48cb1ac37997a7ab82a4636531c1ff75880

==

Transferring ✅
Tx (2% tax):
https://testnet.bscscan.com/tx/0xf35fb1e7e7b2d028a5f4764c4ea6f7a7f496ee0e5aa378eea22c4b5b19b0cc86

==

Internal swap (BNB to marketing wallet | reward token to dividend tracker | reward distribution) ✅
Tx:
https://testnet.bscscan.com/address/0x97f848b4b35df1e6ee84f1dd2ba903191d835aa2#internaltx

MANUAL REVIEW

Severity Criteria

Expelee assesses the severity of disclosed vulnerabilities according to methodology based on OWASP standarts.

Vulnerabilities are dividend into three primary risk categroies:
High
Medium
Low

High-level considerations for vulnerabilities span the following key areas when conducting assessments:

- Malicious input handling
- Escalation of privileges
- Arithmetic
- Gas use

Overall Risk Severity				
	HIGH	Medium	High	Critical
Impact	MEDIUM	Low	Medium	High
	LOW	Note	Low	Medium
		LOW	MEDIUM	HIGH
		Likelihood		

HIGH RISK FINDING

Trades are disabled by default

Category: Centralization
Status: Open
Impact: High

Overview:
The contract has been structured such that all trading is disabled by default, necessitating the contract owner's manual intervention to enable trading. This can lead to a situation where, if trades remain disabled, token holders won't be able to buy, sell, or trade their tokens, causing a severe impact on the token's usability and market liquidity.

```
function enableTrading() external onlyOwner {
   require(!tradingEnabled, "Trading already enabled.");
   tradingEnabled = true;
   swapEnabled = true;
}
```

Suggestion:
To mitigate this risk, it is recommended that trading be enabled before the token presale. This can be achieved by invoking the "enableTrading" function or by transferring ownership of the contract to a third-party that has established trust with the community, such as a Certified SAFU developer. This reduces the concentration of power and the potential for malicious actions, thereby promoting a more decentralized and fair environment for all participants.

HIGH RISK FINDING

Whitelisted wallets paying fees

Category: Logical
Status: Open
Impact: High

Overview:
_transfer function is not excluding whitelisted wallets from fees. This means even if a wallet is included in _excludeFromFees mapping, it still has to pay 2% fees

```
_totalFees = 2;
_burnFee = 0;

if (_totalFees > 0) {
    uint256 fees = (amount * _totalFees) / 100;
    amount = amount - fees;
    super._transfer(from, address(this), fees);
}
```

Suggestion:
Ensure that _totalFees is set to 0 if one of sender or recipient wallets is excluded from fees:

```
if(_isExcludedFromFees[from] _isExcludedFromFees[to]){
_totalFees = 0;
}
```

ABOUT EXPELEE

Expelee is a product–based aspirational Web3 start-up. Coping up with numerous solutions for blockchain security and constructing a Web3 ecosystem from deal making platform to developer hosting open platform, while also developing our own commercial and sustainable blockchain.

🌐 www.expelee.com

🐦 expeleeofficial   Ⓜ expelee

✈ Expelee   in expelee

📷 expelee_official   🐙 expelee-co

Building the Futuristic Blockchain Ecosystem

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 guarantess against the sale of team tokens or the removal of liquidity by the project audited in this document.

Always do your own research and project yourselves from being scammed. The Expelee 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 Expelee receive a payment to manipulate those results or change the awarding badge that we will be adding in our website. Alway 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 Expelee team disclaims any liability for the resulting losses.

expelee

Building the Futuristic Blockchain Ecosystem