expelee

Building the Futuristic Blockchain Ecosystem

SECURITY AUDIT REPORT

AIPEPE

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**
KYC Verification	**No**
Audit Date	**21 April 2023**

PROJECT DESCRIPTION

AIPEPE

He’s AI. He’s Pepe. He’s CEO. The first meme token that gives $Arb rewards!

SOCIAL MEDIA PROFILES

AIPEPE

https://t.me/AI_PEPE_CEO

https://twitter.com/AI_PEPE_CEO

https://www.ai-pepe-ceo.com/

It's always good to check the social profiles of the project, before making your investment.

Team Expelee

CONTRACT DETAILS

Token Name: AI-Pepe-CEO

Symbol: AI-Pepe-CEO

Network: Arbitrum

Language: Solidity

Contract Address:
0xb29Fa64770A85a0D37182b643B4D5F5fFD9BBa9D

Total Supply: 420,000,000,000,000

Contract SHA-256 Checksum:
aa1e6198802d5b018dde3b30cdd11e09c9e38b00

Owner's Wallet:
0x58c3118d75be9bacbf66d7e1dd3725dfe35bfff4

Testnet:
https://testnet.bscscan.com/token/0x179bb17efd81b87e36149a31a4140f9875381377

OWNER PRIVILEGES

- Contract owner is not able to set buy/sell taxes over 10% each
- Contract owner is not able to set transfer fee (0% transfer fee)
- Contract owner is not able to set limits for buy/sell/transfer amounts
- Contract owner is not able to blacklist an arbitrary wallet
- Contract owner is not able to disable trades/transfers
- Contract owner is not able to mint new tokens
- Contract owner must enable trades for public

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

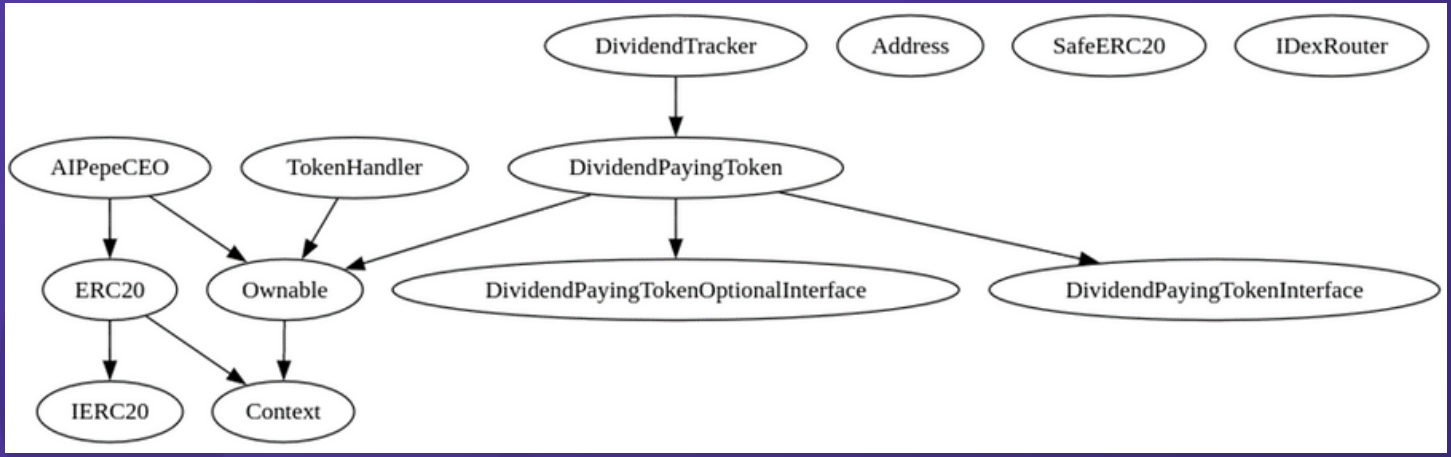
Issues on this level are minor details and warning that can remain unfixed.

Informational

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

expelee

INHERITANCE TREES



FUNCTION DETAILS

Symbol	Meaning
🔴	Function can modify state
💵	Function is payable

Contract Assesment

Contract	Type	Bases		
└	**Function Name**	**Visibility**	**Mutability**	**Modifiers**
Context	Implementation			
└	_msgSender	Internal 🔒		
└	_msgData	Internal 🔒		
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 🔐		
SafeERC20	Library			
└	safeTransfer	Internal 🔒	🔴	
└	_callOptionalReturn	Private 🔐	🔴	
IERC20	Interface			
└	totalSupply	External ❗		NO ❗
└	balanceOf	External ❗		NO ❗
└	transfer	External ❗	🔴	NO ❗
└	allowance	External ❗		NO ❗
└	approve	External ❗	🔴	NO ❗

expelee

FUNCTION DETAILS

└	transferFrom	External ❗	🔴	NO ❗
└	name	External ❗		NO ❗
└	symbol	External ❗		NO ❗
└	decimals	External ❗		NO ❗
ERC20	Implementation	Context, IERC20		
└	<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 🔒	🔴	
└	_createInitialSupply	Internal 🔒	🔴	
└	_approve	Internal 🔒	🔴	
Ownable	Implementation	Context		
└	<Constructor>	Public ❗	🔴	NO ❗
└	owner	Public ❗		NO ❗
└	renounceOwnership	External ❗	🔴	onlyOwner
└	transferOwnership	Public ❗	🔴	onlyOwner
IDexRouter	Interface			
└	factory	External ❗		NO ❗
└	WETH	External ❗		NO ❗
└	swapExactTokensForETHSupportingFeeOnTransferTokens	External ❗	🔴	NO ❗
└	swapExactETHForTokensSupportingFeeOnTransferTokens	External ❗	💵	NO ❗
└	swapExactTokensForTokensSupportingFeeOnTransferTokens	External ❗	🔴	NO ❗
└	swapExactTokensForTokensSupportingFeeOnTransferTokens	External ❗	🔴	NO ❗
└	addLiquidityETH	External ❗	💵	NO ❗
└	addLiquidity	External ❗	🔴	NO ❗
└	getAmountsOut	External ❗		NO ❗
└	removeLiquidity	External ❗	🔴	NO ❗
DividendPayingTokenOptionalInterface	Interface			
└	withdrawableDividendOf	External ❗		NO ❗
└	withdrawnDividendOf	External ❗		NO ❗
└	accumulativeDividendOf	External ❗		NO ❗
DividendPayingTokenInterface	Interface			

expelee

FUNCTION DETAILS

└	dividendOf	External ❗		NO ❗
└	distributeDividends	External ❗	💵	NO ❗
└	withdrawDividend	External ❗	🔴	NO ❗
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 🔒		
DividendPayingToken	Implementation	DividendPayingTokenInterface, DividendPayingTokenOptionalInterface, Ownable		
└	<Receive Ether>	External ❗	💵	NO ❗
└	distributeDividends	Public ❗	💵	NO ❗
└	distributeTokenDividends	Public ❗	🔴	onlyOwner
└	withdrawDividend	Public ❗	🔴	NO ❗
└	_withdrawDividendOfUser	Internal 🔒	🔴	
└	dividendOf	Public ❗		NO ❗
└	withdrawableDividendOf	Public ❗		NO ❗
└	withdrawnDividendOf	Public ❗		NO ❗
└	accumulativeDividendOf	Public ❗		NO ❗
└	_increase	Internal 🔒	🔴	
└	_reduce	Internal 🔒	🔴	
└	_setBalance	Internal 🔒	🔴	
DividendTracker	Implementation	DividendPayingToken		
└	<Constructor>	Public ❗	🔴	NO ❗
└	get	Private 🔐		
└	getIndexOfKey	Private 🔐		
└	getKeyAtIndex	Private 🔐		
└	size	Private 🔐		
└	set	Private 🔐	🔴	

expelee

FUNCTION DETAILS

└	remove	Private 🔐	🔴		
└	excludeFromDividends	External ❗	🔴	onlyOwner
└	includeInDividends	External ❗	🔴	onlyOwner
└	updateClaimWait	External ❗	🔴	onlyOwner
└	getLastProcessedIndex	External ❗		NO ❗	
└	getNumberOfTokenHolders	External ❗		NO ❗	
└	getAccount	Public ❗		NO ❗	
└	getAccountAtIndex	Public ❗		NO ❗	
└	canAutoClaim	Private 🔐		
└	setBalance	External ❗	🔴	onlyOwner
└	process	Public ❗	🔴	NO ❗	
└	processAccount	Public ❗	🔴	onlyOwner
IDexFactory	Interface			
└	createPair	External ❗	🔴	NO ❗	
ILpPair	Interface			
└	sync	External ❗	🔴	NO ❗	
TokenHandler	Implementation	Ownable		
└	sendTokenToOwner	External ❗	🔴	onlyOwner
AIPepeCEO	Implementation	ERC20, Ownable		
└	<Constructor>	Public ❗	🔴	ERC20
└	createPair	Internal 🔒	🔴		
└	updateAllowanceForSwapping	External ❗	🔴	NO ❗	
└	startTrading	External ❗	🔴	onlyOwner
└	excludeFromDividends	External ❗	🔴	onlyOwner
└	includeInDividends	External ❗	🔴	onlyOwner
└	removeLimits	External ❗	🔴	onlyOwner
└	updateMaxBuyAmt	External ❗	🔴	onlyOwner
└	updateMaxSellAmt	External ❗	🔴	onlyOwner
└	removeMaxWallet	External ❗	🔴	onlyOwner
└	updateSwapTokensAtAmt	External ❗	🔴	onlyOwner
└	_excludeFromMaxTransaction	Private 🔐	🔴		
└	airdropToWallets	External ❗	🔴	onlyOwner
└	excludeFromMaxTransaction	External ❗	🔴	onlyOwner
└	setAutomatedMarketMakerPair	Public ❗	🔴	onlyOwner
└	updateBuyTax	External ❗	🔴	onlyOwner
└	updateSellTax	External ❗	🔴	onlyOwner
└	excludeFromTax	Public ❗	🔴	onlyOwner
└	updateClaimWait	External ❗	🔴	onlyOwner
└	getClaimWait	External ❗		NO ❗	
└	getTotalDividendsDistributed	External ❗		NO ❗	
└	withdrawableDividendOf	Public ❗		NO ❗	
└	dividendTokenBalanceOf	Public ❗		NO ❗	

expelee

FUNCTION DETAILS

└	getAccountDividendsInfo	External ❗	|NO ❗
└	getAccountDividendsInfoAtIndex	External ❗	|NO ❗
└	claim	External ❗	🔴 |NO ❗
└	getLastProcessedIndex	External ❗	|NO ❗
└	getNumberOfDividendTokenHolders	External ❗	|NO ❗
└	getNumberOfDividends	External ❗	|NO ❗
└	_transfer	Internal 🔒	🔴	
└	swapTokensForREWARDTOKEN	Private 🔐	🔴	
└	swapBack	Private 🔐	🔴	
└	setMarketingAddress	External ❗	🔴	onlyOwner
└	forceSwapBack	External ❗	🔴	onlyOwner
└	transferForeignToken	External ❗	🔴	onlyOwner
└	updateGasForProcessing	External ❗	🔴	onlyOwner

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				
Impact	HIGH	Medium	High	Critical
	MEDIUM	Low	Medium	High
	LOW	Note	Low	Medium
		LOW	MEDIUM	HIGH
	Likelihood			

FINDINGS

Findings	Severity	Found
High Risk	High	1
Medium Risk	Medium	1
Low Risk	Low	0
Suggestion & discussion	Informational	0
Gas Optimizations	Gas Opt.	0

HIGH RISK FINDING

Enabling trades is not guaranteed

Category: Centralization
Impact: High
Overview:
The owner of the contract must enable trades for public, otherwise no one would be able to buy/sell/transfer their tokens except whitelisted wallets.

```
  function startTrading() external onlyOwner {
require(!tradingLive, "Trading is already active, cannot     relaunch.");
require(lpPair != address(0), "Create Pair First");
tradingLive = true;
swapEnabled = true;
      tradingLiveBlock = block.number;
      emit StartedTrading();
  }
```

Suggestion
To mitigate this issue there are several options:
- Temporary transfer ownership of the contract to a pinksale safu developer (done)
- Enable tradings before presale

Issue Status: Open

MEDIUM RISK FINDING

Trade limits

Category: Centralization
Impact: Medium
Overview:
The owner of the contract is able to set limits for max amount of buy/sell/holding. The safeguard for this limit is 1% of total supply (meaning this limits can not be lower than this amount).
Max wallet is 1% always, unless disabled by owner using removeMaxWallet function (once disabled can not be enabled again)

```
  function updateMaxBuyAmt(uint256 newNum) external onlyOwner {
    require(
newNum >= ((totalSupply() * 1) / 100) / 1e18,
"Cannot set max sell amt lower than 1%"
    );
maxBuyAmt = newNum * (10 ** 18);
    emit UpdatedMaxBuyAmt(maxBuyAmt);
  }

  function updateMaxSellAmt(uint256 newNum) external onlyOwner {
    require(
newNum >= ((totalSupply() * 1) / 100) / 1e18,
"Cannot set max sell amt lower than 1%"
    );
maxSellAmt = newNum * (10 ** 18);
    emit UpdatedMaxSellAmt(maxSellAmt);
  }
```

MEDIUM RISK FINDING

Suggestion

To mitigate this issue there are several options:
- Make sure to follow pinksale safu criteria for proper safeguards (done)

Issue Status: Resolved

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
expelee
expelee_official
expelee-co

expelee

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.

Building the Futuristic Blockchain Ecosystem