expelee

Building the Futuristic Blockchain Ecosystem

SECURITY AUDIT REPORT

PEPEMO

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 ?	Not Detected
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

NOTE

Please note that the audited token is currently deployed and being traded on Pancakeswap V2. Our audit aimed to identify potential codebase issues at the time of auditing. However, due to the nature of blockchain technology and the ongoing use of the token:

- We do not assume responsibility for any future issues, vulnerabilities, or risks that may arise and have not been identified in this report.
- We have no ongoing liability or duty of care for the token after the conclusion of the audit.

Please exercise your own judgement and caution when interacting with this token.

expelee

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	**25 June 2023**

CONTRACT DETAILS

Token Name: PepeMo

Symbol: PEPEMO

Network: Binance smart chain

Language: Solidity

Contract Address:
0x5D9cC38c5BeD98EA9e35Fc712D2c97C850BbBA33

Total Supply: 420,690,000,000,000

Owner's Wallet:
0xFd74E1785101aF7332ca12D33B0Ee0D3fcCb889b

Deployer's Wallet:
0xFd74E1785101aF7332ca12D33B0Ee0D3fcCb889b

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

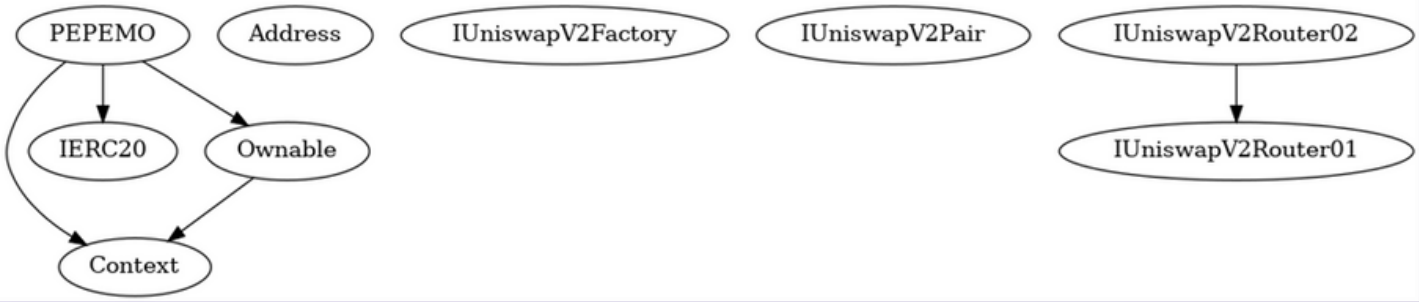
Issues on this level are minor details and warning that can remain unfixed.

Informational

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

expelee

INHERITANCE TREES



TESTNET VERSION

Adding Liquidity ✓
Tx:
https://testnet.bscscan.com/tx/0x15977cb3da3df6c7afaa67f3cad1b3db8aa02d837b28bcfcb9a377715090498e

==========

Buying when excluded from fees ✓
Tx (0% tax):
https://testnet.bscscan.com/tx/0xb220593bfbe6ae072de8f324c20b8ec285870575d5490a8f422c19ed4124dd41

==========

Selling when excluded from fees ✓
Tx (0% tax):
https://testnet.bscscan.com/tx/0x5abf0253f9095b15edb4ef66d3e2dd4b168ea09514bab703a4a8a4ed3c799885

==========

Transferring when excluded from fees ✓
Tx (0% tax):
https://testnet.bscscan.com/tx/0x81e291418a57a7ea997d5e91d8458006ea90b92fb5f3140adf6f4b612b0d3ced

==========

TESTNET VERSION

Buying ✓
Tx (4% tax):
https://testnet.bscscan.com/tx/0x66ba7886a35fae5de6a32f5ed8332bdda4eb648f4e16086ef41250bdbe19dbaf

==========

Selling ✓
Tx (4% tax):
https://testnet.bscscan.com/tx/0xb97ada27bcc1c5c088cec0ca45ec49c90d2a1027289eec39e7da2e07dc0dacf2

==========

Transferring ✓
Tx (0% tax):
https://testnet.bscscan.com/tx/0xb998d9b34184ca0bb8cb5e56cbceb037648446f502ac1fc8251b312e83b11aa3

==========

Internal swap (marketing wallet received BNB) ✓
Tx :
https://testnet.bscscan.com/address/0x107b2d8783c839d752ccbba151b5cde7b27829b0#internaltx

==========

expelee

FUNCTION DETAILS

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 ❗
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 🔐		
Ownable	Implementation	Context		
└	<Constructor>	Public ❗	🔴	NO ❗
└	owner	Public ❗		NO ❗
└	_checkOwner	Internal 🔒		
└	renounceOwnership	Public ❗	🔴	onlyOwner

expelee

FUNCTION DETAILS

└	transferOwnership	Public ❗	🔴	onlyOwner
└	_transferOwnership	Internal 🔒	🔴	
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 ❗
IUniswapV2Router01	Interface			
└	factory	External ❗		NO ❗
└	WETH	External ❗		NO ❗
└	addLiquidity	External ❗	🔴	NO ❗
└	addLiquidityETH	External ❗	💵	NO ❗

FUNCTION DETAILS

└	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 ❗

|||||

PEPEMO	Implementation	Context, IERC20, Ownable		
└	<Constructor>	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 ❗
└	_approve	Private 🔐	🔴	
└	_transfer	Private 🔐	🔴	
└	swapAndLiquify	Public ❗	🔴	lockTheSwap
└	swapTokensForEth	Private 🔐	🔴	
└	_tokenTransfer	Private 🔐	🔴	
└	isExcludedFromFee	External ❗		NO ❗
└	excludeFromFee	External ❗	🔴	onlyOwner
└	includeInFee	External ❗	🔴	onlyOwner
└	setTokensToSwap	External ❗	🔴	onlyOwner
└	setSwapAndLiquifyEnabled	External ❗	🔴	onlyOwner
└	setMarketingWallet	External ❗	🔴	onlyOwner
└	transferToAddressETH	Private 🔐	🔴	

FUNCTION DETAILS

└	<Receive Ether>	External ❗	💵	NO ❗
└	swapETHForTokens	Private 🔐	🔴	
└	recoverTokensFromContract	External ❗	🔴	onlyOwner
└	enableTrading	External ❗	🔴	onlyOwner
└	recoverETHfromContract	External ❗	🔴	onlyOwner
└	setRouterAddress	Public ❗	🔴	onlyOwner

Legend

Symbol	Meaning
🔴	Function can modify state
💵	Function is payable

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			

HIGH RISK FINDING

Category: Centralization
Subject: Trades must be enabled manually
Status: Resolved (Token is being activly traded on pancakeswap v2)
Severity : High

Overview

Owner of the contract must call startTrading function in order for holders to be able to transfer/sell their tokens. If Owner refuse to enable trades for any reason, holders wont be able to transfer/sell their tokens and their assets will be locked in liquidity pool forever.

```
    function enableTrading() external onlyOwner {
require(!tradingEnabled, "Trading already enabled.");
        tradingEnabled = true;
swapAndLiquifyEnabled = true;
        emit AuditLog("We have Enable Trading and Automatic Swaps:",
msg.sender);
    }
```

Suggestion

To resolve this issue you can:
- Enable trades prior to presale: this ensures investors about safety of their assets
- Transfer ownreship of the contract to pinksale: this ensures that trades will be enabled eventually at right time
- Create a time lock feature: this ensures that trades will be enabled automatically after a fixed amount of time.

HIGH RISK FINDING

Category: Configurations
Subject: Updating pcsv2 router
Status: Open – Not applicable
Severity : High

Overview

Owner of the contract is able to update pcs router to a new address. This router is used to perform internal swaps of the contract.
If new router is set to a malicious or broken contract, sell swaps and transfers could be disabled if contract try to perform an internal swap

```
    function setRouterAddress(address newRouter) public onlyOwner {
      // PepeMo Change Router
      IUniswapV2Router02 _newPancakeRouter =
IUniswapV2Router02(newRouter);
      uniswapV2Pair =
IUniswapV2Factory(_newPancakeRouter.factory()).createPair(address(this),
_newPancakeRouter.WETH());
      uniswapV2Router = _newPancakeRouter;
    }
```

Suggestion

Ensure that Rotuer is not upgradeable, since token is already launched and being traded on pcsv2, there are no other options other than renouncing the ownership for resolving this issue, which also means owner won't have control on any functions on the contract. For example owner wont be able to whitelist an exchange wallet.

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.