expelee

Building the Futuristic Blockchain Ecosystem

SECURITY AUDIT REPORT

PEPEFANS TOKEN

HIGH RISK ANALYSIS

This section contains a brief Summary of all the High Risks present in this Project's Smart Contract

Findings	Found
High Risk	2

High Risk Details

Owner can pause trade
Owner can change fees up to 100%

Risk : Centralisation and High

Overview

Trading functionality with the ability to enable or disable trading. The contract owner has the privilege to control the trading state through the setTradingEnabled function.

Owner of the contract to update the buy/sell fees of the contract. But unfortunately owner has overpowered role and it has arbitrary limit

More Details of this High Risk are given on Page 13 of this Audit Report

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	**1 Jun 2023**

CONTRACT DETAILS

Token Name: PEPEFANS

Symbol: PEPEF

Network: Binance Smart Chain

Language: Solidity

Contract Address:
0xC20296414821297B401c278d5BAb824d602EeB14

Total Supply: 420690000000

Owner's Wallet:
0x6c837E921B81234c078a50A369D8f1eC81A0CB23

Deployer's Wallet:
0x6c837E921B81234c078a50A369D8f1eC81A0CB23

OWNER PRIVILEGES

- Owner can pause trade
- Owner can change fees up to 100%
- Owner can exclude account from fees
- Owner can change swap setting

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.

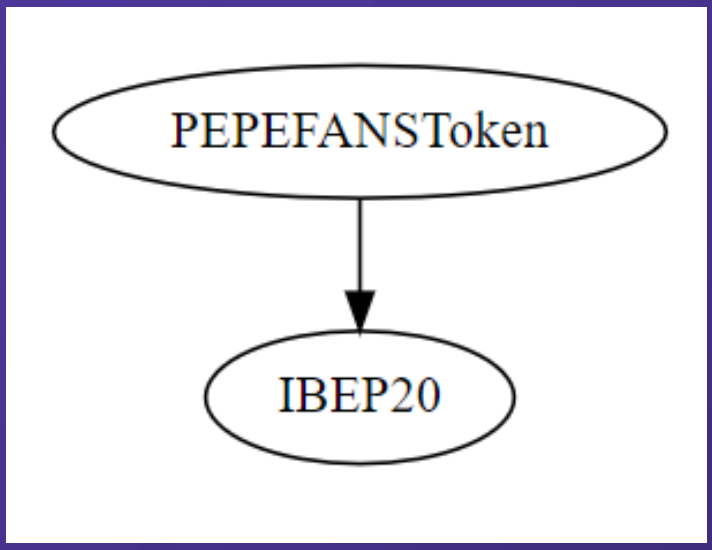
Low Risk

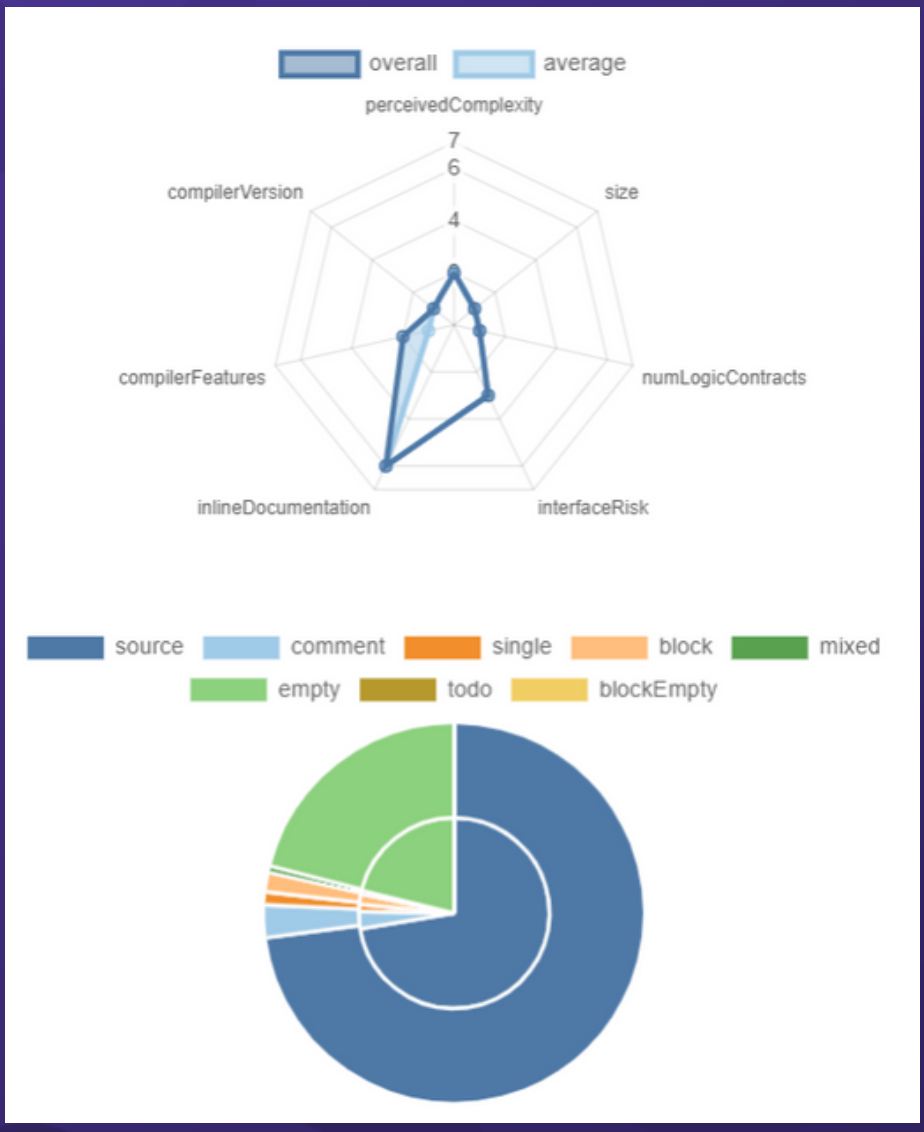
Issues on this level are minor details and warning that can remain unfixed.

Informational

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

INHERITANCE TREES





expelee

FUNCTION DETAILS

IBEP20	Interface				
└	totalSupply	External	!		NO !
└	balanceOf	External	!		NO !
└	transfer	External	!	🔴	NO !
└	allowance	External	!		NO !
└	approve	External	!	🔴	NO !
└	transferFrom	External	!	🔴	NO !
PEPEFANSToken	Implementation	IBEP20			
└	<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 !
└	setMarketingWallet	Public	!	🔴	onlyOwner
└	setBuyTaxes	Public	!	🔴	onlyOwner
└	setSellTaxes	Public	!	🔴	onlyOwner
└	setSwapTokensAtAmount	Public	!	🔴	onlyOwner
└	setSwapTransferFees	Public	!	🔴	onlyOwner
└	setWhitelistStatus	Public	!	🔴	onlyOwner
└	setTradingEnabled	Public	!	🔴	onlyOwner
└	setSwappingEnabled	Public	!	🔴	onlyOwner
└	_transfer	Internal 🔒		🔴	
└	_approve	Internal 🔒		🔴	

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			

expelee

FINDINGS

Findings	Severity	Found
High Risk	High	2
Medium Risk	Medium	0
Low Risk	Low	2
Suggestion & discussion	Informational	5
Gas Optimizations	Gas Opt.	0

HIGH RISK FINDING

Owner can pause trade

Risk : Centralisation
Severity : High

Overview

Trading functionality with the ability to enable or disable trading. The contract owner has the privilege to control the trading state through the **setTradingEnabled** function.

```
function setTradingEnabled(bool enabled) public onlyOwner {
    _tradingEnabled = enabled;
}
```

Recommendation

Thoroughly review the **onlyOwner** modifier implementation to ensure that only the contract owner can modify the trading state. Verify that the modifier effectively validates the contract owner's address and that it cannot be bypassed or manipulated by unauthorized parties.

HIGH RISK FINDING

Owner can change fees up to 100%

Severity : High

Overview

Functions that allows the owner of the contract to update the buy/sell fees of the contract. But unfortunately owner has overpowered role and it has arbitrary limit

```
function setBuyTaxes(uint256 taxPercentage) public onlyOwner {
    _buyTaxPercentage = taxPercentage;
}

0 references | Control flow graph | 0940bbc7 | ftrace | funcSig
function setSellTaxes(uint256 taxPercentage) public onlyOwner {
    _sellTaxPercentage = taxPercentage;
}
```

Recommendation

Fees rate should be acceptable limit for investors there is no max limit in these functions. It is recommended to add additional access control measures, such as multi-factor authentication or time-based restrictions, to limit the number of authorized users who can call these functions.

LOW RISK FINDING

Owner can exclude accounts from fees

Severity : Low

Overview

Excludes/Includes an address from the collection of fees

```
function setWhitelistStatus(address account, bool whitelisted) public onlyOwner {
    _whitelistedAccounts[account] = whitelisted;
}
```

Recommendation

It is recommended to add additional access control measures, such as multi-factor authentication or time-based restrictions, to limit the number of authorized users who can call these functions. The contract owner account is well secured and only accessible by authorized parties.

LOW RISK FINDING

Owner can change swap setting

Severity : Low

Overview

s**etSwapingEnabled** function allows the contract owner to enable or disable the automatic **swapping**.

```
function setSwappingEnabled(bool enabled) public onlyOwner {
    _swappingEnabled = enabled;
}
```

Recommendation

Be ensure that the contract owner account is well secured and only accessible by authorized parties.

INFORMATONAL FINDING

Lack of zero address check

Severity : Informational

Overview

Detect missing zero address validation.

```
function setMarketingWallet(address walletAddress) public onlyOwner {
    _marketingWallet = walletAddress;
}
```

Recommendation

Check that the address is not zero.

INFORMATONAL FINDING

Missing events arithmetic

Severity : Informational

Overview

Events are used to emit information about an action that has occurred on the blockchain, so that it can be observed by external systems or users. The contract was found to be missing these events on the function .

setSwappingEnabled()
setWhitelistStatus()
setTradingEnabled()
setSellTaxes()
setBuyTaxes()
setMarketingWallet()

Recommendation

Consider emitting events for the functions mentioned above. It is also recommended to have the addresses indexed. Emit an event for critical parameter changes.

INFORMATONAL FINDING

Unused variables/states/functions

Severity : Informational

Overview

Unused variables/states/functions detected.

```
function setSwapTransferFees(bool enabled) public onlyOwner {
    _swapTransferFeesEnabled = enabled;
}
```

```
function setSwapTokensAtAmount(uint256 amount) public onlyOwner {
    _swapTokensAtAmount = amount;
}
```

Recommendation

Remove unused variables/states/functions

expelee

INFORMATONAL FINDING

Too many digits

Severity : Informational

Overview

Literals with many digits are difficult to read and review.

```
_totalSupply = 420690000000000000000000000000000;
```

Recommendation

While 1_ether looks like 1 ether, it is 10 ether. As a result, it's likely to be used incorrectly.

INFORMATONAL FINDING

Outdated versions and floating pragma;

Severity : Informational

Overview

Outdated versions were detected **pragma solidity ^0.8.0;**

```
pragma solidity ^0.8.0;
```

Recommendation

Consider using the latest version of Solidity for testing. Should lock pragmas to a specific compiler version. Besides, consider the known compiler bugs in the following references and check whether the contracts include those bugs.

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.