Building the Futuristic Blockchain Ecosystem

SECURITY AUDIT REPORT

SUPER PEPE

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	Pass
KYC Verification	-
Audit Date	24 May 2023

PROJECT DESCRIPTION

Super Pepe Coin is the first MEME-themed cryptocurrency based on the story of Pepe Superman. We aim to build a large and active community where every member can participate and gain profits by holding and trading Super Pepe Coins.

SOCIAL MEDIA PROFILES

SUPER PEPE

superpepe.club

Super Pepe Coin is the first MEME-themed cryptocurrency based on the story of Pepe Superman. We aim to build a large and active community where every member can participate and gain profits

superpepe_bsc

superpepe.club

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

Team Expelee

CONTRACT DETAILS

Token Name: Superpepe

Symbol: Superpepe

Network: Binance Smart Chain

Language: Solidity

Contract Address:
0x50baB0905Cb04b53a0208b6970ed6f20d78d0854

Total Supply: 1000

Owner's Wallet:
0xB8658374Cd665501641829771B90Cd87CA2A6bFe

Deployer's Wallet:
0xB8658374Cd665501641829771B90Cd87CA2A6bFe

OWNER PRIVILEGES

- Owner can include/exclude account from reward
- Owner can change fees max 25%
- Owner exclude account from fees
- Owner can change swap settings

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.

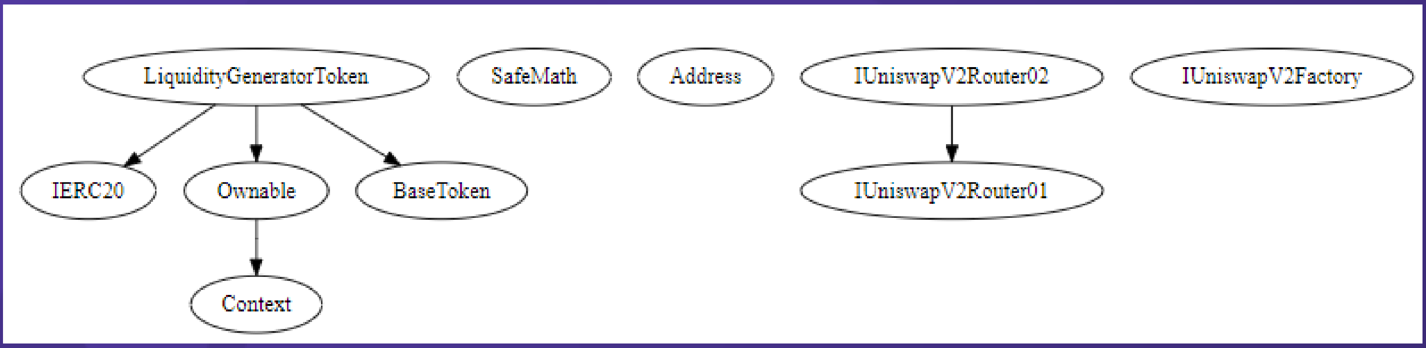
Low Risk

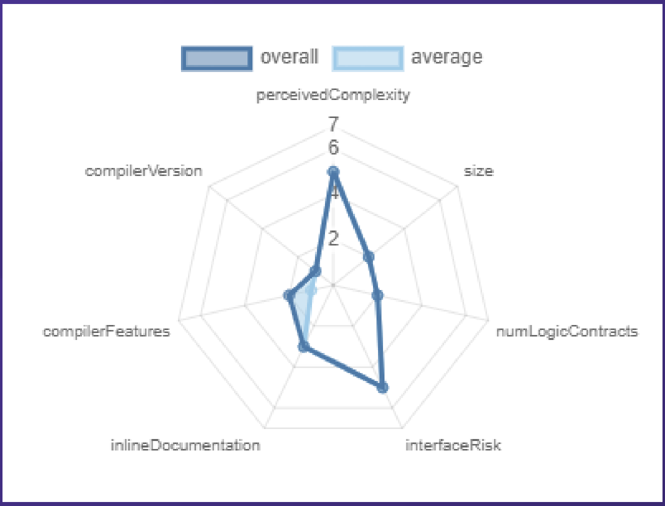
Issues on this level are minor details and warning that can remain unfixed.

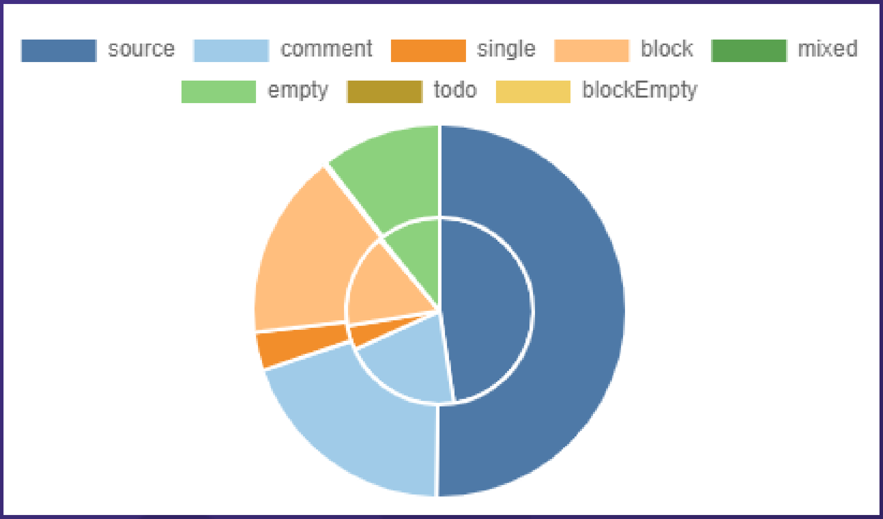
Informational

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

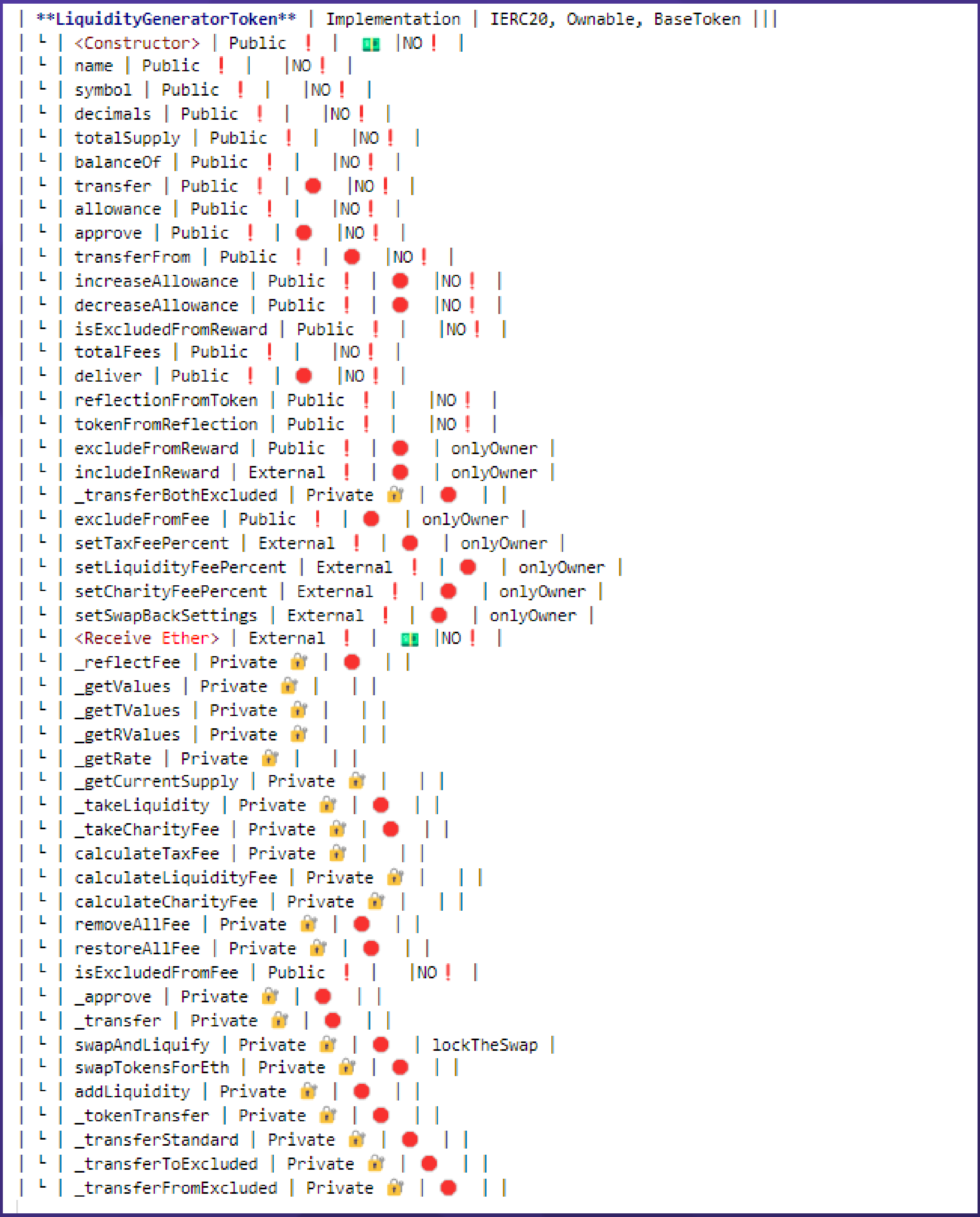
INHERITANCE TREES







FUNCTION DETAILS



```
| **LiquidityGeneratorToken** | Implementation | IERC20, Ownable, BaseToken |||
| └ | <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❗️ |
| └ | totalFees | Public ❗️ |   |NO❗️ |
| └ | deliver | Public ❗️ | 🛑  |NO❗️ |
| └ | reflectionFromToken | Public ❗️ |   |NO❗️ |
| └ | tokenFromReflection | Public ❗️ |   |NO❗️ |
| └ | excludeFromReward | Public ❗️ | 🛑  | onlyOwner |
| └ | includeInReward | External ❗️ | 🛑  | onlyOwner |
| └ | _transferBothExcluded | Private 🔐 | 🛑  | |
| └ | excludeFromFee | Public ❗️ | 🛑  | onlyOwner |
| └ | setTaxFeePercent | External ❗️ | 🛑  | onlyOwner |
| └ | setLiquidityFeePercent | External ❗️ | 🛑  | onlyOwner |
| └ | setCharityFeePercent | External ❗️ | 🛑  | onlyOwner |
| └ | setSwapBackSettings | External ❗️ | 🛑  | onlyOwner |
| └ | <Receive Ether> | External ❗️ | 💵 |NO❗️ |
| └ | _reflectFee | Private 🔐 | 🛑  | |
| └ | _getValues | Private 🔐 |   | |
| └ | _getTValues | Private 🔐 |   | |
| └ | _getRValues | Private 🔐 |   | |
| └ | _getRate | Private 🔐 |   | |
| └ | _getCurrentSupply | Private 🔐 |   | |
| └ | _takeLiquidity | Private 🔐 | 🛑  | |
| └ | _takeCharityFee | Private 🔐 | 🛑  | |
| └ | calculateTaxFee | Private 🔐 |   | |
| └ | calculateLiquidityFee | Private 🔐 |   | |
| └ | calculateCharityFee | Private 🔐 |   | |
| └ | removeAllFee | Private 🔐 | 🛑  | |
| └ | restoreAllFee | Private 🔐 | 🛑  | |
| └ | isExcludedFromFee | Public ❗️ |   |NO❗️ |
| └ | _approve | Private 🔐 | 🛑  | |
| └ | _transfer | Private 🔐 | 🛑  | |
| └ | swapAndLiquify | Private 🔐 | 🛑  | lockTheSwap |
| └ | swapTokensForEth | Private 🔐 | 🛑  | |
| └ | addLiquidity | Private 🔐 | 🛑  | |
| └ | _tokenTransfer | Private 🔐 | 🛑  | |
| └ | _transferStandard | Private 🔐 | 🛑  | |
| └ | _transferToExcluded | Private 🔐 | 🛑  | |
| └ | _transferFromExcluded | Private 🔐 | 🛑  | |
```

MANUAL REVIEW

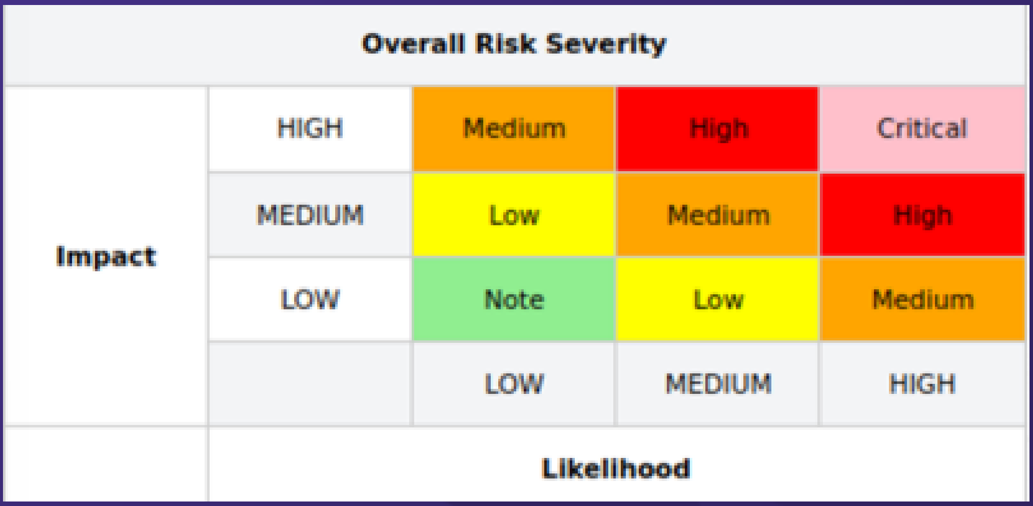
Severity Criteria

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	0
Medium Risk	🟠 Medium	0
Low Risk	🟡 Low	4
Suggestion & discussion	🔵 Informational	0
Gas Optimizations	🟣 Gas Opt.	0

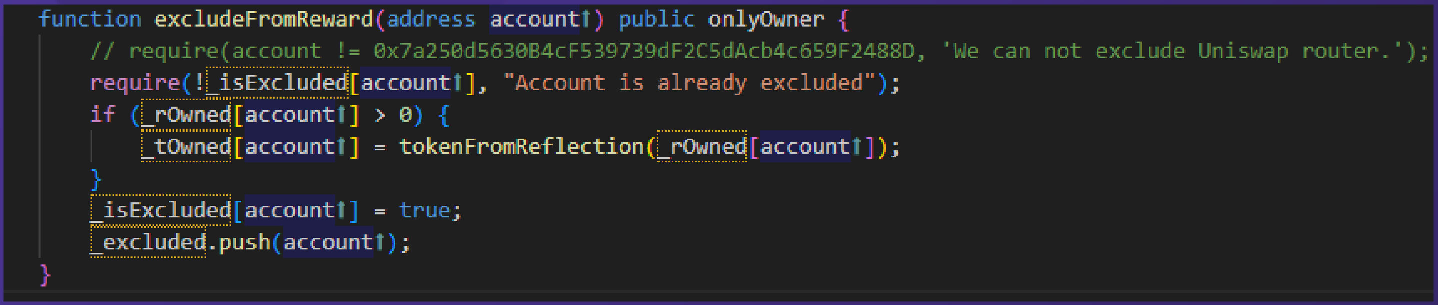
LOW RISK FINDING

Owner can include/exclude account from reward

Severity : Low

Overview

Function that allows the owner of the contract to exclude an address from receiving dividends



```
function excludeFromReward(address account) public onlyOwner {
    // require(account != 0x7a250d5630B4cF539739dF2C5dAcb4c659F2488D, 'We can not exclude Uniswap router.');
    require(!_isExcluded[account], "Account is already excluded");
    if (_rOwned[account] > 0) {
        _tOwned[account] = tokenFromReflection(_rOwned[account]);
    }
    _isExcluded[account] = true;
    _excluded.push(account);
}
```

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.

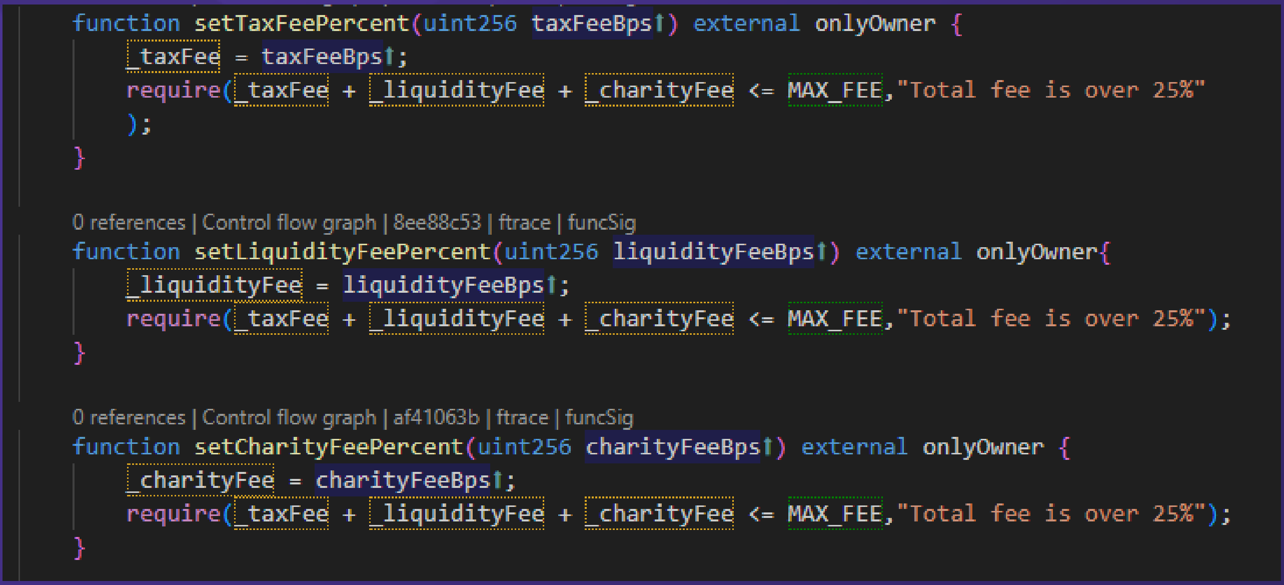
LOW RISK FINDING

Owner can change fees max 25%

Severity : Low

Overview

Functions that allows the owner of the contract to update the fees of the contract. These functions assumes that the input parameters are valid and do not exceed the maximum limit of total 25%



```
function setTaxFeePercent(uint256 taxFeeBps) external onlyOwner {
    _taxFee = taxFeeBps;
    require(_taxFee + _liquidityFee + _charityFee <= MAX_FEE,"Total fee is over 25%"
    );
}

0 references | Control flow graph | 8ee88c53 | ftrace | funcSig
function setLiquidityFeePercent(uint256 liquidityFeeBps) external onlyOwner{
    _liquidityFee = liquidityFeeBps;
    require(_taxFee + _liquidityFee + _charityFee <= MAX_FEE,"Total fee is over 25%");
}

0 references | Control flow graph | af41063b | ftrace | funcSig
function setCharityFeePercent(uint256 charityFeeBps) external onlyOwner {
    _charityFee = charityFeeBps;
    require(_taxFee + _liquidityFee + _charityFee <= MAX_FEE,"Total fee is over 25%");
}
```

Recommendation

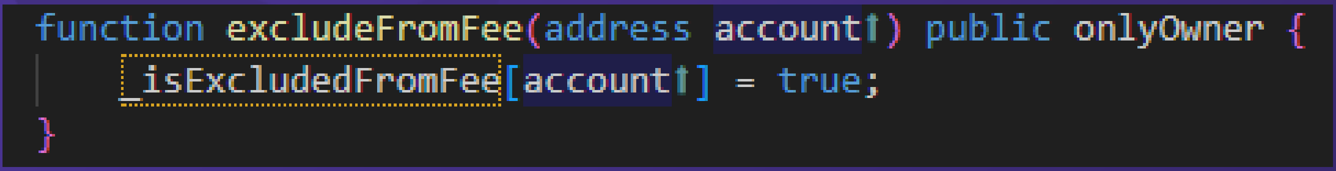
It is recommended to add additional access control measures, such as multi-factor authentication or time-based restrictions, to limit the number of authorized users who can call these functions.

LOW RISK FINDING

Owner can exclude accounts from fees

Severity : Low

Overview
Excludes/Includes an address from the collection of fees



```
function excludeFromFee(address account) public onlyOwner {
    _isExcludedFromFee[account] = true;
}
```

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.

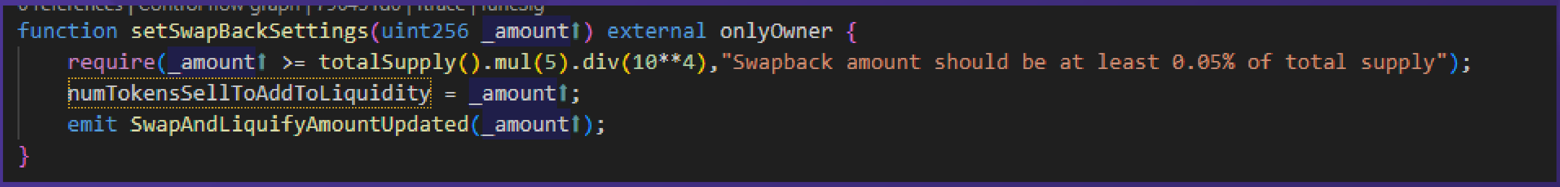
LOW RISK FINDING

Owner can change swap setting

Severity : Low

Overview

Functions allows the contract owner to updating swap token at amounts within reasonable limit.



```
function setSwapBackSettings(uint256 _amount) external onlyOwner {
    require(_amount >= totalSupply().mul(5).div(10**4),"Swapback amount should be at least 0.05% of total supply");
    numTokensSellToAddToLiquidity = _amount;
    emit SwapAndLiquifyAmountUpdated(_amount);
}
```

Recommendation

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

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