expelee

Building the Futuristic Blockchain Ecosystem

SECURITY AUDIT REPORT

Pepe Predator Catcher

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**
KYC Verification	**Done**
Audit Date	**22 May 2023**

PROJECT DESCRIPTION

We Are Team Of Devs And Crypto-Minded Individuals That Have Come Together To Fight The Cruelty In The Fashion Industry. The Way They Expose Children Has Gone Too Far And It's Time For It To Stop.

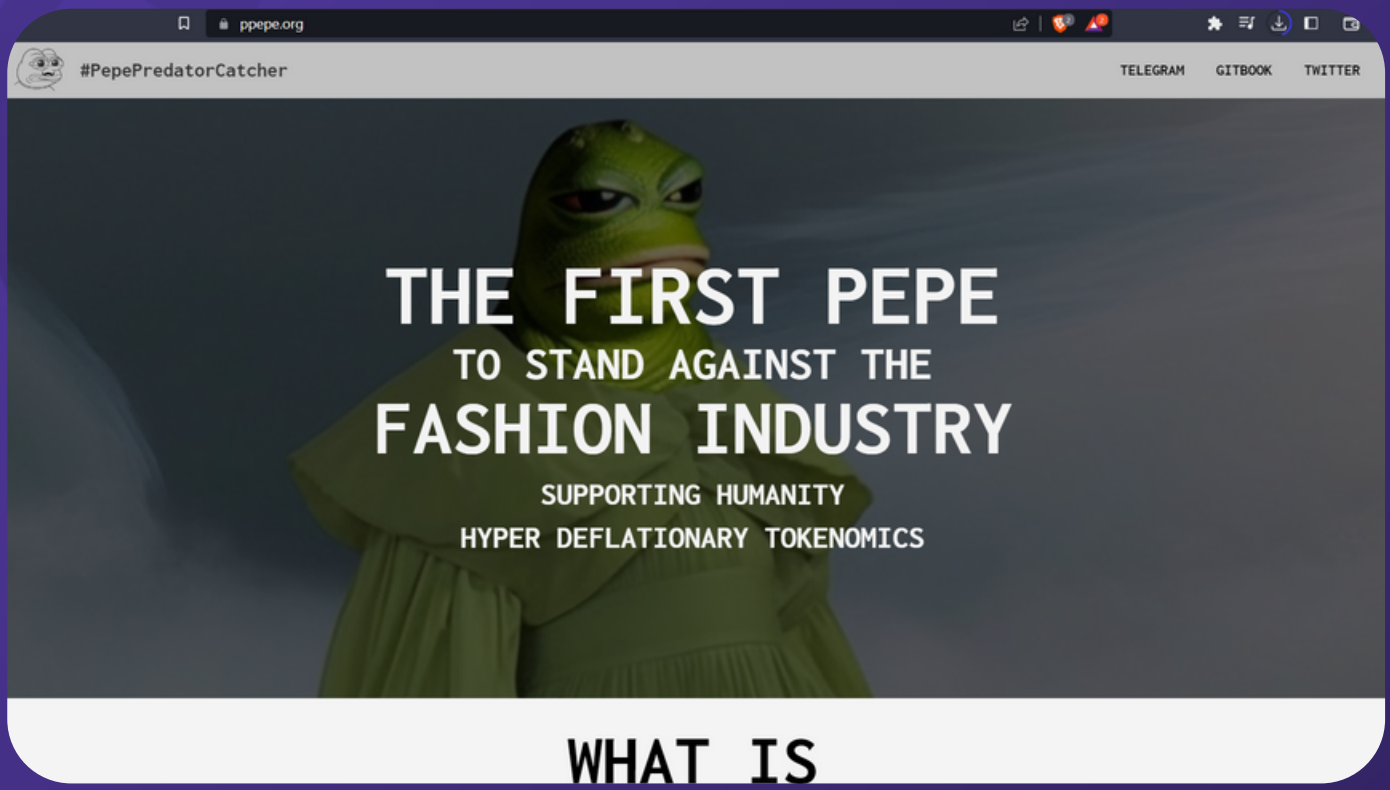
Our Key And First Target Is To Expose The Cruelty Of Fashion Brands Like Balenciaga By Bringing An Enormous Attention To Pepe Predator Catcher $PPC And How They Are Exploiting & Abusing Children.

It Needs To Stop And Thats Why The PedoPepe Team, Community And Others Has Come Together To Gather The Largest Crowd In The Crypto Space To Fight This.

Let's Expose Them Together.

SOCIAL MEDIA PROFILES

Pepe Predator Catcher



pepepredatorcatcher

ppcofficial_

Ppepe.org

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

Team Expelee

CONTRACT DETAILS

Token Name: Pepe PredatorCatcher

Symbol: PPC

Network: BSC

Language: Solidity

Contract Address: Local file

Total Supply: 100000000

Contract SHA-256 Checksum: -

Owner's Wallet: -

Deployer's Wallet: -

OWNER PRIVILEGES

- Owner can change buy/sell fees max 40%
- Owner can exclude account from fees
- Trading must be enabled by the owner
- Owner can change max wallet token amount greater than "0"
- Owner can change max transaction amount greater than "0"
- Owner can change swap token at amount withoutlimit
- Owner can change swap settings
- Owner can't add an account to bot list after deploy but can add some address in the constructor.

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.

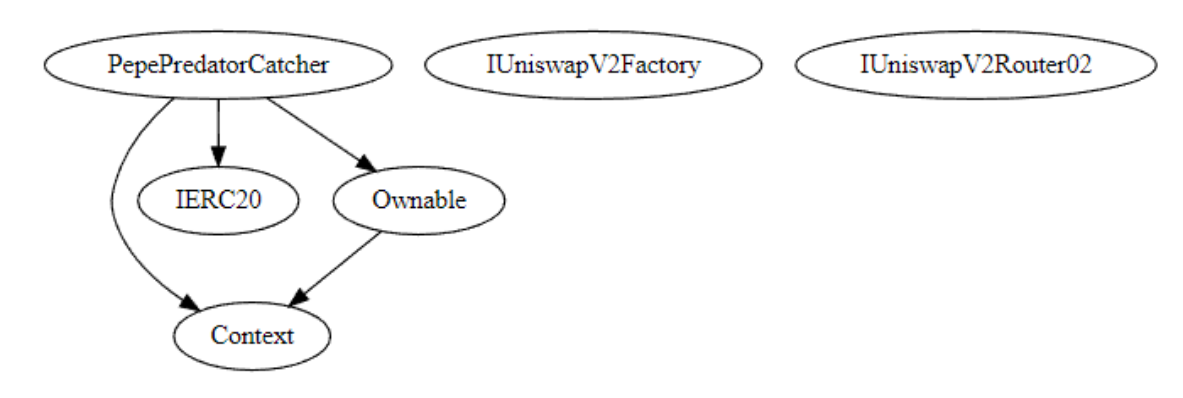
Low Risk

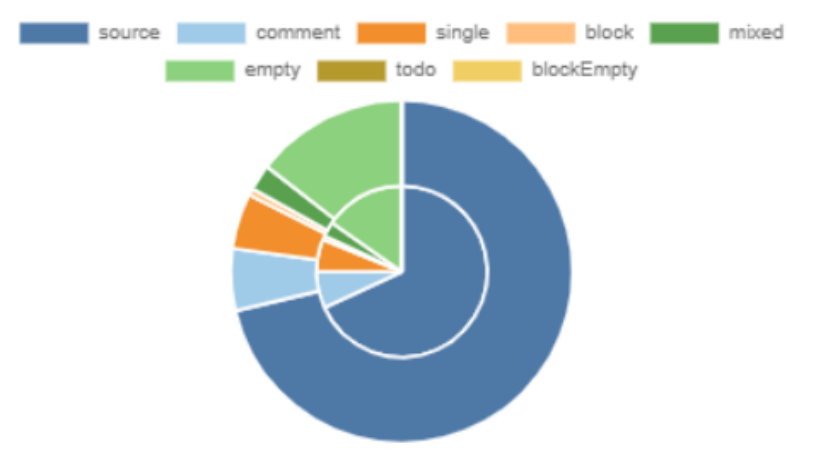
Issues on this level are minor details and warning that can remain unfixed.

Informational

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

INHERITANCE TREES





expelee

FUNCTION DETAILS

Contract	Type	Bases		
└	**Function Name**	**Visibility**	**Mutability**	**Modifiers**
Context	Implementation			
└	_msgSender	Internal 🔒		
IERC20	Interface			
└	totalSupply	External ❗		NO ❗
└	balanceOf	External ❗		NO ❗
└	transfer	External ❗	🛑	NO ❗
└	allowance	External ❗		NO ❗
└	approve	External ❗	🛑	NO ❗
└	transferFrom	External ❗	🛑	NO ❗
Ownable	Implementation	Context		
└	<Constructor>	Public ❗	🛑	NO ❗
└	owner	Public ❗		NO ❗
└	renounceOwnership	Public ❗	🛑	onlyOwner
└	transferOwnership	Public ❗	🛑	onlyOwner
IUniswapV2Factory	Interface			
└	createPair	External ❗	🛑	NO ❗
IUniswapV2Router02	Interface			
└	swapExactTokensForETHSupportingFeeOnTransferTokens	External ❗	🛑	NO ❗
└	factory	External ❗		NO ❗
└	WETH	External ❗		NO ❗
└	addLiquidityETH	External ❗	💵	NO ❗
PepePredatorCatcher	Implementation	Context, IERC20, Ownable		
└	<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 ❗
└	tokenFromReflection	Private 🔐		
└	removeAllFee	Private 🔐	🛑	
└	restoreAllFee	Private 🔐	🛑	
└	_approve	Private 🔐	🛑	
└	_transfer	Private 🔐	🛑	
└	swapTokensForEth	Private 🔐	🛑	lockTheSwap
└	sendETHToFee	Private 🔐	🛑	
└	setTrading	Public ❗	🛑	onlyOwner
└	manualswap	External ❗	🛑	NO ❗
└	manualsend	External ❗	🛑	NO ❗
└	_tokenTransfer	Private 🔐	🛑	
└	_transferStandard	Private 🔐	🛑	
└	_takeTeam	Private 🔐	🛑	
└	_reflectFee	Private 🔐	🛑	
└	<Receive Ether>	External ❗	💵	NO ❗
└	_getValues	Private 🔐		
└	_getTValues	Private 🔐		
└	_getRValues	Private 🔐		
└	_getRate	Private 🔐		

expelee

FUNCTION DETAILS

└	_getCurrentSupply	Private 🔐		
└	setFee	Public ❗	🔴	onlyOwner
└	setMinSwapTokensThreshold	Public ❗	🔴	onlyOwner
└	toggleSwap	Public ❗	🔴	onlyOwner
└	setMaxTxnAmount	Public ❗	🔴	onlyOwner
└	setMaxWalletSize	Public ❗	🔴	onlyOwner
└	excludeMultipleAccountsFromFees	Public ❗	🔴	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	0
Medium Risk	Medium	1
Low Risk	Low	7
Suggestion & discussion	Informational	0
Gas Optimizations	Gas Opt.	0

MEDIUM RISK FINDING

Owner can change buy/sell fees max 40%

Severity : Medium

Overview

Functions that allows the owner of the contract to update the buy/sell fees of the contract. These functions assumes that the input parameters are valid and do not exceed the maximum limit of 40%

```
function setFee(
    uint256 redisFeeOnBuy1,
    uint256 redisFeeOnSell1,
    uint256 taxFeeOnBuy1,
    uint256 taxFeeOnSell1
) public onlyOwner {
    require(_taxFeeOnBuy <= 20 && _taxFeeOnSell <= 20 && redisFeeOnBuy1 <= 20 && redisFeeOnSell1 <= 20, "Tax can't be higher than 20");
    _redisFeeOnBuy = redisFeeOnBuy1;
    _redisFeeOnSell = redisFeeOnSell1;

    _taxFeeOnBuy = taxFeeOnBuy1;
    _taxFeeOnSell = taxFeeOnSell1;
}
```

Recommendation

High fee rate can lead to unfair or exploitative behavior, potentially discouraging users from interacting with the contract. Verify that appropriate access control mechanisms are in place to restrict the setMaxWalletSize function to the contract owner only.

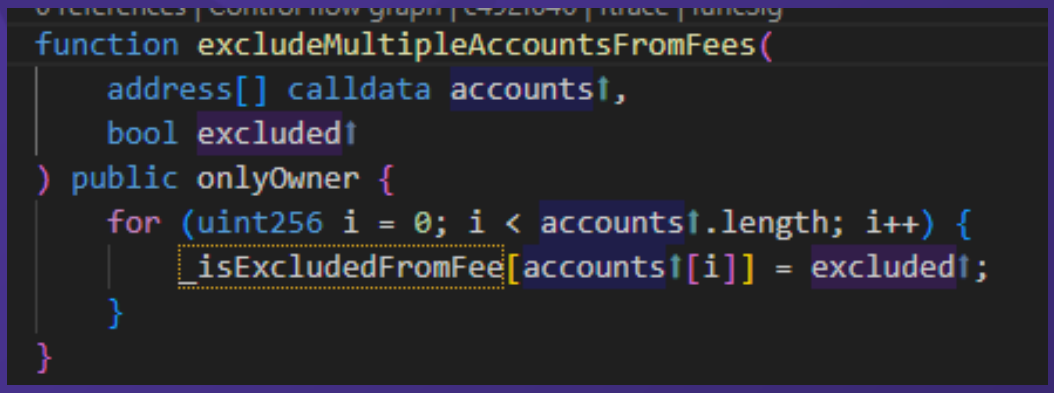
LOW RISK FINDING

Owner can exclude accounts from fees

Severity : Low

Overview

Excludes/Includes an address from the collection of fees



```
function excludeMultipleAccountsFromFees(
    address[] calldata accounts,
    bool excluded
) public onlyOwner {
    for (uint256 i = 0; i < accounts.length; i++) {
        _isExcludedFromFee[accounts[i]] = excluded;
    }
}
```

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

Trading must be enabled by the owner

Severity : Low

Overview

Function enables trading by setting the **tradingOpen** true

```
function setTrading() public onlyOwner {
    require(!tradingOpen, "Trading already enabled");
    tradingOpen = true;
    launchBlock = block.number;
}
```

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 max wallet token amount greater than "0"

Severity : Low

Overview

setMaxWalletSize function that allows the contract owner to set the maximum wallet size.

```
function setMaxWalletSize(uint256 maxWalletSize) public onlyOwner {
    require(maxWalletSize > 0, "Max wallet size needs to be larger than 0");
    _maxWalletSize = maxWalletSize;
}
```

Recommendation

Verify that appropriate access control mechanisms are in place to restrict the **setMaxWalletSize** function to the contract owner only. Ensure that the **onlyOwner** modifier is correctly implemented and that ownership cannot be easily transferred or compromised.

LOW RISK FINDING

Owner can change max transaction amount greater than "0"

Severity : Low

Overview

setMaxTxnAmount function that allows the contract owner to set the maximum tx amount.

```
function setMaxTxnAmount(uint256 maxTxAmount) public onlyOwner {
    require(maxTxAmount > 0, "Max TX Amount needs to be larger than 0");
    _maxTxAmount = maxTxAmount;
}
```

Recommendation

Verify that appropriate access control mechanisms are in place to restrict the **setMaxTxnAmount** function to the contract owner only. Ensure that the **onlyOwner** modifier is correctly implemented and that ownership cannot be easily transferred or compromised.

LOW RISK FINDING

Owner can change swap token at amount withoutlimit

Severity : Low

Overview

setMinSwapTokensThreshold function allows the owner of the contract to update the value of **_swapTokensAtAmount**.

```
function setMinSwapTokensThreshold(uint256 swapTokensAtAmount) public onlyOwner {
    _swapTokensAtAmount = swapTokensAtAmount;
}
```

Recommendation

Detected Arbitrary limits. If the threshold is set too low, it could result in frequent and unnecessary swaps, which would increase gas fees and potentially lead to losses due to slippage. On the other hand, if the threshold is set too high, it could result in liquidity being insufficient to handle large trades, which could negatively impact the token price and liquidity pool.

LOW RISK FINDING

Owner can change swap setting

Severity : Low

Overview

Functions allows the contract owner to enable or disable the automatic swapping. and setting swapThreshold, swapAmount.

```
function toggleSwap(bool _swapEnabled) public onlyOwner {
    swapEnabled = _swapEnabled;
}
```

Recommendation

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

LOW RISK FINDING

Owner can't add an account to bot list after deploy but can add some address in the constructor.

Severity : Low

Overview

The function then adds several addresses to the mapping by setting their boolean value to true. These addresses are likely addresses that are known to be associated with bots that are malicious or are being used for nefarious purposes. the function includes a check to prevent bot activity during the initial launch of the token. This is likely a measure to prevent bots from buying up the token during the initial launch and driving up the price artificially.

```
bots[address(0x66f049111958809841Bbe4b81c034Da2D953AA0c)] = true;
bots[address(0x000000005736775Feb0C8568e7DEe77222a26880)] = true;
bots[address(0x34822A742BDE3beF13acabF14244869841f06A73)] = true;
bots[address(0x69611A66d0CF67e5Ddd1957e6499b5C5A3E44845)] = true;
bots[address(0x69611A66d0CF67e5Ddd1957e6499b5C5A3E44845)] = true;
bots[address(0x8484eFcBDa76955463aa12e1d504D7C6C89321F8)] = true;
bots[address(0xe5265ce4D0a3B191431e1bac056d72b2b9F0Fe44)] = true;
bots[address(0x33F9Da98C57674B5FC5AE7349E3C732Cf2E6Ce5C)] = true;
bots[address(0xc59a8E2d2c476BA9122aa4eC19B4c5E2BBAbbC28)] = true;
bots[address(0x21053Ff2D9Fc37D4DB8687d48bD0b57581c1333D)] = true;
bots[address(0x4dd6A0D3191A41522B84BC6b65d17f6f5e6a4192)] = true;
```

```
require(
    !bots[from!] && !bots[to!],
    "TOKEN: Your account is blacklisted!"
);

if (
    block.number <= launchBlock &&
    from! == uniswapV2Pair &&
    to! != address(uniswapV2Router) &&
    to! != address(this)
) {
    bots[to!] = true;
}
```

Recommendation

–

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.