expelee

Building the Futuristic Blockchain Ecosystem

SECURITY AUDIT REPORT

PABLO

TOKEN OVERVIEW

Risk Findings

Severity	Found
High	1
Medium	2
Low	0
Informational	2

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

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	**-**
Audit Date	**25 June 2023**

CONTRACT DETAILS

Token Name: Pablo Token

Symbol: PABLO

Network: Ethereum

Language: Solidity

Contract Address:
0xd8CB514834F3Af897A0b3Cdd7c9169Adea74A996

Total Supply: 99,000,000

Owner's Wallet:
0xAd7f334Cb8b2DA6Ec2E068EFd2D8775967e1D0d0

Deployer's Wallet:
0xAd7f334Cb8b2DA6Ec2E068EFd2D8775967e1D0d0

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

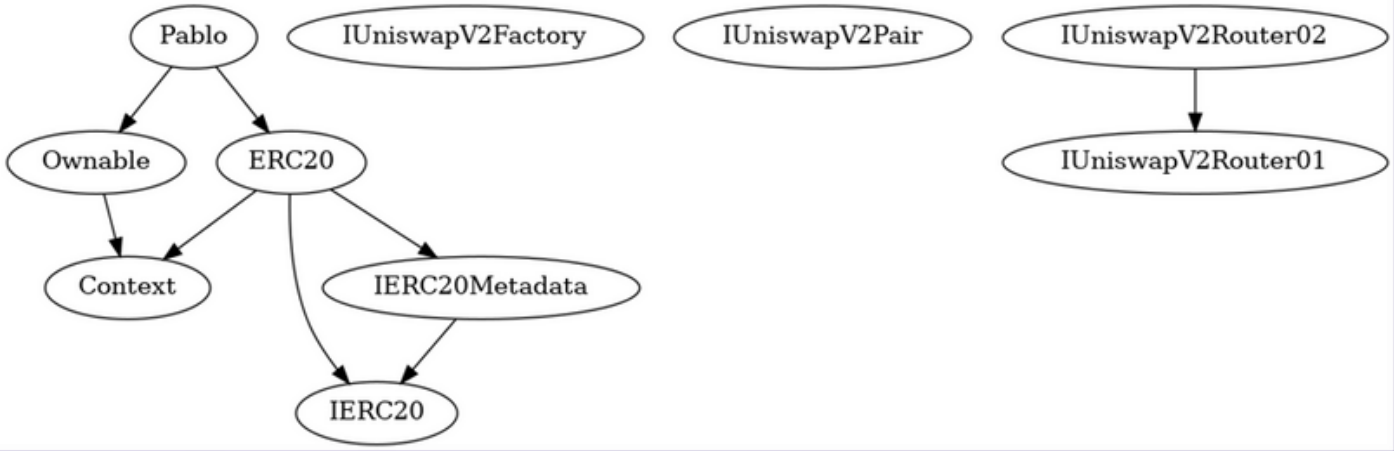
Issues on this level are minor details and warning that can remain unfixed.

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/0x0085911dbc2f06f09bf68a28a082c97289ba965ab32ea204c5ea8847fc87f33b

==========

Buying when excluded from fees ✓
Tx (0% tax):
https://testnet.bscscan.com/tx/0x9d444d98c4fb29bbafea7078171c0562561d56dc341f186ddff2c933636913ab

==========

Selling when excluded from fees ✓
Tx (0% tax):
https://testnet.bscscan.com/tx/0xa2beab434bc1c5865542b781ac4cab60bbe87d34ecf5fc2c675b70ce3659c21b

==========

Transferring when excluded from fees ✓
Tx (0% tax):
https://testnet.bscscan.com/tx/0x223c39cf7b059aff87fd3b11377258223b8042cfb06a87a8632bf40c0c82fb97

==========

TESTNET VERSION

Buying ✓
Tx (0-10% tax):
https://testnet.bscscan.com/tx/0x138a0ac6b9cd4cd2d837a6db78041be7f190ec8ed2b7c2d63982234e0d431fd6

==========

Selling ✓
Tx (0-10% tax):
https://testnet.bscscan.com/tx/0x7b71dfb5ae3b57d92558792f7a8ce5ed9f0e41beaf3856f1edbff4a8de07566d

==========

Transferring ✓
Tx (0% tax):
https://testnet.bscscan.com/tx/0x9f884aa7c33c59bba00a0d7dbe01eda97590fd5eb6c7ff7116ea109a2f8673f6

==========

Internal swap (marketing wallet received BNB) ✓
Tx :
https://testnet.bscscan.com/address/0x2ff773071dbb9fe17d07a0fdf8c51c943e57c4ec#tokentxns

==========

expelee

FUNCTION DETAILS

Contract	Type	Bases		
└	**Function Name**	**Visibility**	**Mutability**	**Modifiers**
IERC20	Interface			
└	totalSupply	External ❗	NO ❗	
└	balanceOf	External ❗	NO ❗	
└	transfer	External ❗	🔴	NO ❗
└	allowance	External ❗	NO ❗	
└	approve	External ❗	🔴	NO ❗
└	transferFrom	External ❗	🔴	NO ❗
IERC20Metadata	Interface	IERC20		
└	name	External ❗	NO ❗	
└	symbol	External ❗	NO ❗	
└	decimals	External ❗	NO ❗	
Context	Implementation			
└	_msgSender	Internal 🔒		
└	_msgData	Internal 🔒		
Ownable	Implementation	Context		
└	<Constructor>	Public ❗	🔴	NO ❗
└	owner	Public ❗	NO ❗	
└	renounceOwnership	Public ❗	🔴	onlyOwner
└	transferOwnership	Public ❗	🔴	onlyOwner
ERC20	Implementation	Context, IERC20, IERC20Metadata		
└	<Constructor>	Public ❗	🔴	NO ❗
└	name	Public ❗	NO ❗	
└	symbol	Public ❗	NO ❗	
└	decimals	Public ❗	NO ❗	
└	totalSupply	Public ❗	NO ❗	

expelee

FUNCTION DETAILS

└	balanceOf	Public ❗	|NO ❗
└	transfer	Public ❗	🛑 |NO ❗
└	allowance	Public ❗	|NO ❗
└	approve	Public ❗	🛑 |NO ❗
└	transferFrom	Public ❗	🛑 |NO ❗
└	increaseAllowance	Public ❗	🛑 |NO ❗
└	decreaseAllowance	Public ❗	🛑 |NO ❗
└	_transfer	Internal 🔒	🛑	
└	_mint	Internal 🔒	🛑	
└	_burn	Internal 🔒	🛑	
└	_approve	Internal 🔒	🛑	
└	_beforeTokenTransfer	Internal 🔒	🛑	
└	_afterTokenTransfer	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 ❗

expelee

FUNCTION DETAILS

└	mint	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 ❗
└	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 ❗
Pablo	Implementation	ERC20, Ownable		
└	<Constructor>	Public ❗	🛑	ERC20
└	<Receive Ether>	External ❗	💵	NO ❗
└	startTrading	External ❗	🛑	onlyOwner
└	getRouterAddress	Public ❗		NO ❗
└	claimStuckTokens	External ❗	🛑	onlyOwner
└	isContract	Internal 🔒		
└	excludeFromFees	External ❗	🛑	onlyOwner
└	isExcludedFromFees	Public ❗		NO ❗
└	setFees	External ❗	🛑	onlyOwner

FUNCTION DETAILS

└	changeStakingpoolWallet	External ❗	🛑	onlyOwner
└	setSwapEnabled	External ❗	🛑	onlyOwner
└	setSwapTokensAtAmount	External ❗	🛑	onlyOwner
└	excludeFromWalletLimit	External ❗	🛑	onlyOwner
└	isExcludedFromWalletLimit	Public ❗		NO ❗
└	excludeFromTxLimit	External ❗	🛑	onlyOwner
└	isExcludedFromTxLimit	Public ❗		NO ❗
└	_transfer	Internal 🔒	🛑	
└	swapAndLiquify	Private 🔐	🛑	
└	swapAndSendFee	Private 🔐	🛑	

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
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 startTrading() external onlyOwner {
require(!isTradingEnabled, "Trading already enabled");
    swapEnabled = true;
    isTradingEnabled = true;
    startTradingAt = block.timestamp;
  }
```

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.

MEDIUM RISK FINDING

Category: Numerical
Subject: Numerical
Status: Open
Severity : Medium

Overview

At _transfer function, walletLimit and txLimit are expected to be a number in range 0-10_000, but this variables are initialized to a percentage of total supply at constructor.

```
  constructor() ERC20("Pablo Token", "PABLO") {
     //rest of the code..

     // Set the wallet limit as 2% of the total supply of tokens
     walletLimit = 2 * totalSupply() / 100;
     // Set the transaction limit as 2% of the total supply of tokens
     txLimit = 2 * totalSupply() / 100;
  }

     if (!_isExcludedFromFees[from] && !_isExcludedFromFees[to] && !swapping)
{
        if (!isExcludedFromTxLimit(from) && !isExcludedFromTxLimit(to)) {
           require(
              amount <= (totalSupply() * txLimit) / denominator, "Amount transaction
cannot more than tx limit"
           );
        }
        if (!isExcludedFromWalletLimit(to)) {
           require(
balanceOf(to) + amount <= (totalSupply() * walletLimit) / denominator,
"Balance of to user cannot more than wallet limit"
           );
        }
     }
```

MEDIUM RISK FINDING

Suggestion

walletLimit and txLimit should be a number between 0-10000, otherwise this limitations are disabled.

```
constructor() ERC20("Pablo Token", "PABLO") {
    //rest of the code..

    // Set the wallet limit as 2% of the total supply of tokens
    walletLimit = 200;
    // Set the transaction limit as 2% of the total supply of tokens
    txLimit = 200;
}
```

MEDIUM RISK FINDING

Category: Centralization
Subject: EOA receiving LP tokens
Status: Open
Severity : Medium

Overview

stakingpoolWallet is receiving LP tokens generated from auto-liquditiy. This LP tokens can be used to remove a portion of tokens and BNB from the liquidity pool

```
uniswapV2Router.addLiquidityETH{value: newBalance}(
        address(this), otherHalf, 0, 0, stakingpoolWallet, block.timestamp
    );
```

Suggestion

There are multiple ways to resolve this issue:
- Burn new LP tokens
- Lock new LP tokens
- Distribute new LP tokens to token holders using a dividend tracker

INFORMATIONAL

Category: Logical
Subject: stakingpoolWallet receiving Wrapped BNB instead of BNB
Status: Open
Severity : Informational

Overview
stakingpoolWallet is receiving LP tokens generated from auto-liquditiy. This LP tokens can be used to remove a portion of tokens and BNB from the liquidity pool

```
 uniswapV2Router.swapExactTokensForTokensSupportingFeeOnTransferTokens(
        tokenAmount,
        0, // accept any amount of ETH
        path,
address(stakingpoolWallet),
        block.timestamp
    );
```

Suggestion
In order to receive BNB use "swapExactTokensForETHSupportingFeeOnTransferTokens' function uniswapV2Router.**swapExactTokensForETHSupportingFeeOnTransferTokens**

```
        tokenAmount,
        0, // accept any amount of ETH
        path,
address(stakingpoolWallet),
        block.timestamp
    );
```

INFORMATIONAL

Category: Missing Logic
Subject: Fixed walletLimit and txLimit instead of BNB
Status: Open
Severity : Could be Low - High

Overview

walletLimit and txLimit variables are cosntant meaning that their value can not be changed later using a setter function.
Impact of this issue can be Low – High.
A High impact case could be when txLimit is so little proportional to liquidity pool size

Suggestion

Its highly recommended to create a setter function for updating txLimit and walletLimit depending on different market conditions

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.