expelee

Building the Futuristic Blockchain Ecosystem

SECURITY AUDIT REPORT

ELRICOPEPE

TABLE OF CONTENTS

OVERVIEW

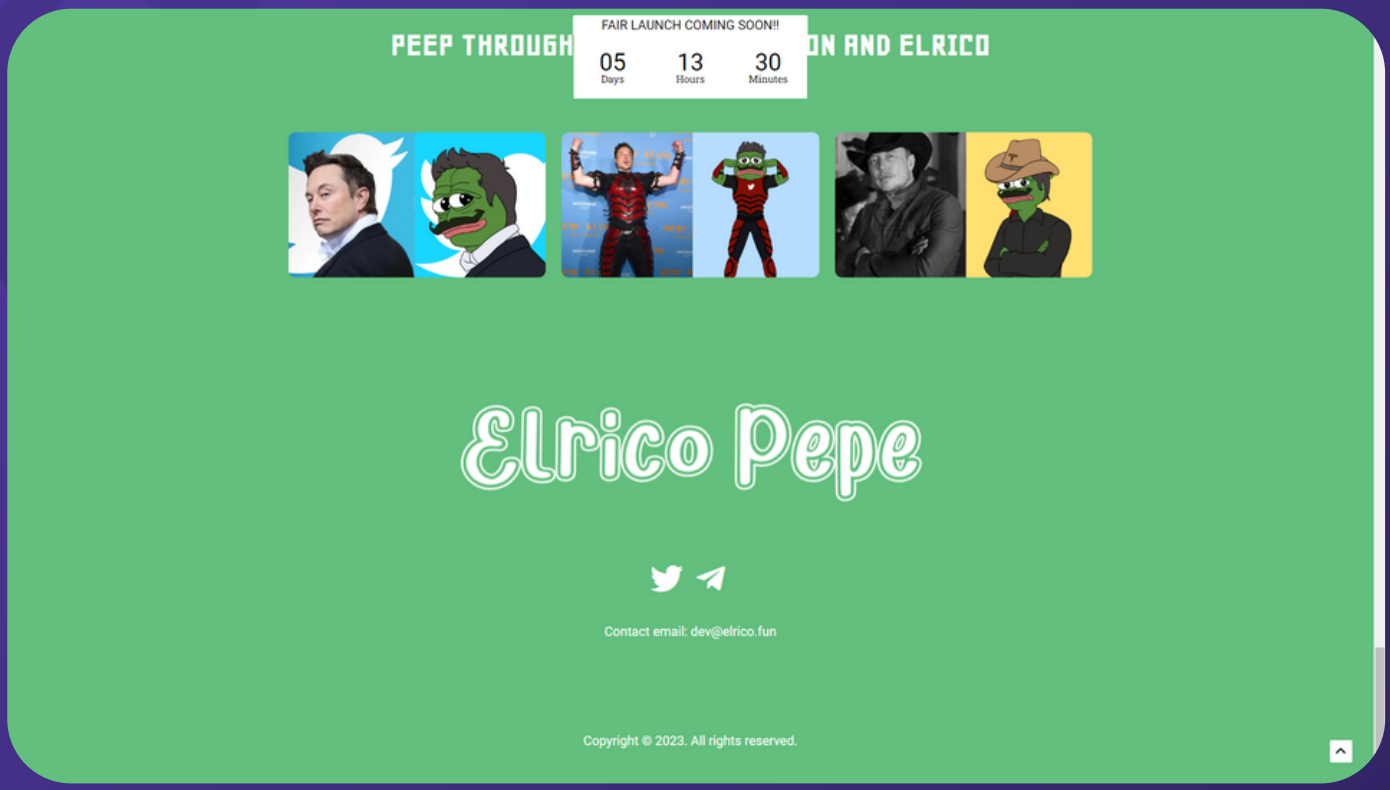
The Expelee team has performed a line-by-line manual analysis and automated review of the smart contract. The smart contract was analysed mainly for common smart contract vulnerabilities, exploits, and manipulation hacks. According to the smart contract audit:

Audit Result	**Passed with High Risk**
KYC Verification	**Done**
Audit Date	**30 May 2023**

PROJECT DESCRIPTION

Elrico Pepe embodies the quintessence of an ideal meme token by offering a trifecta of attributes: amusement, engagement, and entertainment for all. Its goal is to encapsulate the fundamental features of a meme token, while concurrently harnessing the prowess of web3 and blockchain technology to function as a viable medium of exchange for goods and services within a growing ecosystem.

SOCIAL MEDIA PROFILES



https://t.me/elricopepe

https://twitter.com/elricopepe

https://elrico.fun/

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

Team Expelee

CONTRACT DETAILS

Token Name: ELRICO PEPE

Symbol: ELRICO

Network: Binance Smart Chain

Language: Solidity

Contract Address:
0xD11f43d008d3918A5edAbf83173C8719f7620d30

Total Supply: 420690000000

Owner's Wallet:
0x91c8f2B451cdDdFb62fA00AE9fe989d73f595Af8

Deployer's Wallet:
0xE65eC409eedaaA6fD0011e1737dcBd0E30278175

Testnet Link:
https://testnet.bscscan.com/address/0xD70ba52c28191dD64fD05188FD97A7e386347B63

OWNER PRIVILEGES

- Owner can change fees up to 100%
- Owner can exclude account from fees

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

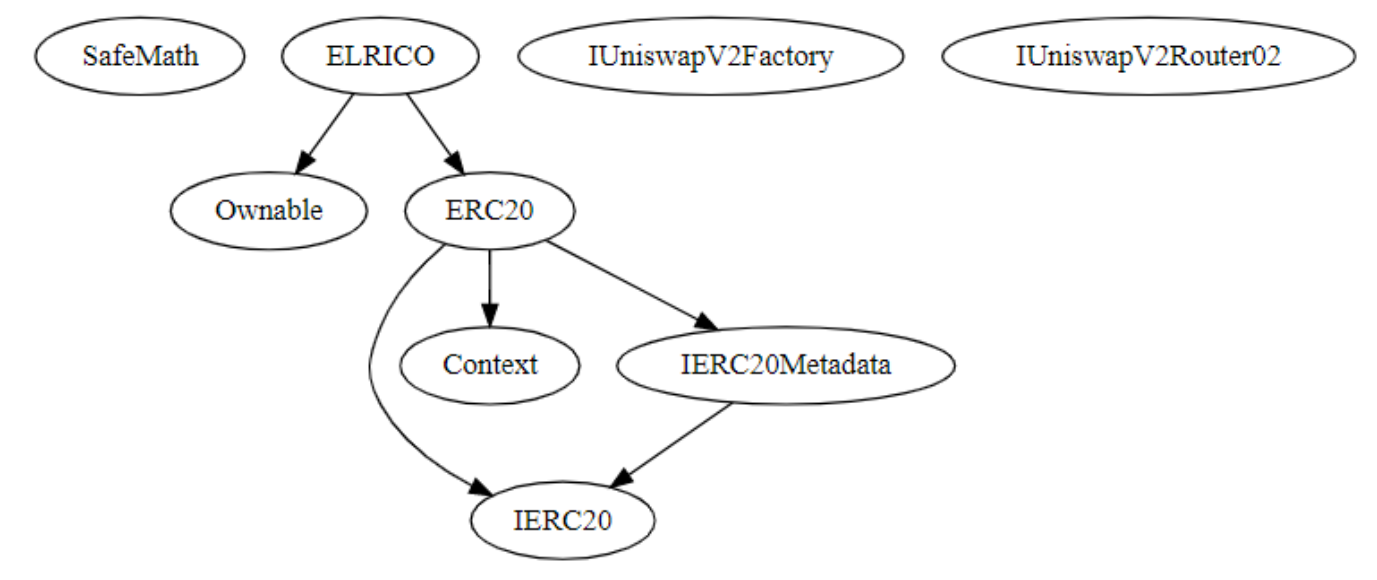
Issues on this level are minor details and warning that can remain unfixed.

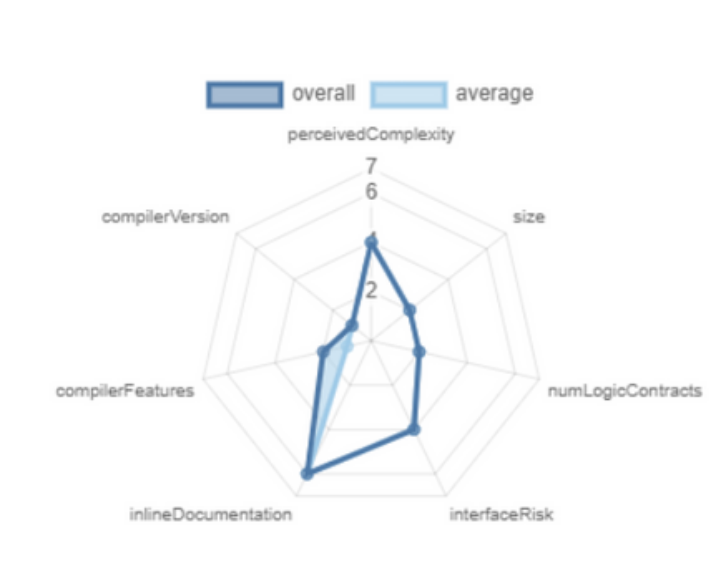
Informational

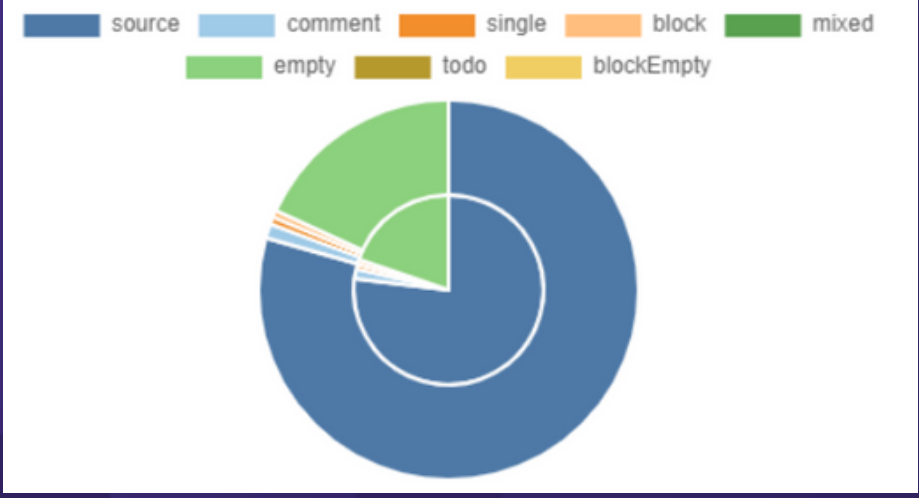
Issues on this level are minor details and warning that can remain unfixed.

expelee

INHERITANCE TREES







expelee

FUNCTION DETAILS

Context	Implementation				
└	_msgSender	Internal 🔒			
└	_msgData	Internal 🔒			
Ownable	Implementation				
└	<Constructor>	Public ❗	🔴	NO ❗	
└	owner	Public ❗		NO ❗	
└	isOwner	Public ❗		NO ❗	
└	renounceOwnership	Public ❗	🔴	onlyOwner	
└	transferOwnership	Public ❗	🔴	onlyOwner	
└	_transferOwnership	Internal 🔒	🔴		
IERC20Metadata	Interface	IERC20			
└	name	External ❗		NO ❗	
└	symbol	External ❗		NO ❗	
└	decimals	External ❗		NO ❗	
ERC20	Implementation	Context, IERC20, IERC20Metadata			
└	<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 ❗	
└	_transfer	Internal 🔒	🔴		
└	_mint	Internal 🔒	🔴		
└	_burn	Internal 🔒	🔴		
└	_approve	Internal 🔒	🔴		
└	_beforeTokenTransfer	Internal 🔒	🔴		
└	_afterTokenTransfer	Internal 🔒	🔴		
ELRICO	Implementation	ERC20, Ownable			
└	<Constructor>	Public ❗	🔴	ERC20	
└	shouldSwapBack	Internal 🔒			
└	_transfer	Internal 🔒	🔴		
└	swapBack	Internal 🔒	🔴	swapping	
└	setFees	External ❗	🔴	onlyOwner	
└	excludeFromFee	External ❗	🔴	onlyOwner	
└	<Receive *Ether*>	External ❗	💵	NO ❗	

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	1
Medium Risk	Medium	0
Low Risk	Low	1
Suggestion & discussion	Informational	0
Gas Optimizations	Gas Opt.	0

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. Owner can change fees without any limit.

```
function setFees(uint256 _buyFee, uint256 _sellFee) external onlyOwner {
    buyFee = _buyFee;
    sellFee = _sellFee;
}
```

Recommendation

Owner shouldn't harm the investor (without limit of fees) to empowered roles. Fees shoudn't be arbitrary limits. The team should carefully manage the private keys of the owner's account. We strongly recommend a powerful security mechanism that will prevent a single user from accessing the contract admin functions.

LOW RISK FINDING

Owner can exclude accounts from fees

Severity : Low

Overview

Excludes/Includes an address from the collection of fees

```
0 references | Control flow graph | 0x84d8fe | ftrace | funcSig
function excludeFromFee(address account↑, bool status↑) external onlyOwner {
    require(account↑ != address(0), "ELRICO: zero address");
    isExcludedFromFee[account↑] = status↑;
}
```

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.

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.