expelee

Building the Futuristic Blockchain Ecosystem

Audit Report FOR

CS Twitter Inu

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	Not Done
Audit Date	17 Oct 2022

-Team Expelee

PROJECT DESCRIPTION

CZ Inu

Welcome to CZ Twitter Inu, a decentralized meme token evolving into a vibrant ecosystem. $CZINU aims to restore integrity to meme coins by focusing on real world utility, philanthropic causes and holder rewards.

expelee

expelee.com

Social Media Profiles

CZ Inu

https://czinubsc.com/

https://t.me/CZTwitterInu

https://twitter.com/czinubnb/

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

-Team Expelee

CONTRACT DETAILS

Contract Name
CZINU

Symbol
CZINU

Network
BSC

Language
Solidity

Contract Address (Verified)
0x898bf7E556EEbd76eDFea9b362635fF4f5D02820

Token Type
ERC 20

Total Supply
500,000,000

Compiler
v0.8.9+commit.e5eed63a

License
default

Contract SHA-256 Checksum:
097E79293CEA2458D0F29CF51F0B9870C68800019051AEB411AE62480EA9D53

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
- Complier
- Hardhat

FUNCTION OVERVIEW

Can Take Back Ownership	Not Detected
Owner Change Balance	Not Detected
Blacklist	Not Detected
Modify Fees	Detected
Proxy	Not Detected
Whitelisted	Not Detected
Anti Whale	Not Detected
Trading Cooldown	Not Detected
Transfer Pausable	Not Detected
Cannot Sell All	Not Detected
Hidden Owner	Not Detected
Mint	Not Detected

VULNERABILITY CHECKLIST

Design Logic	Passed
Compiler warnings.	Passed
Private user data leaks	Passed
Timestamp 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 Zeppelin module	Passed
Fallback function security	Passed

RISK CLASSIFICATION

When performing smart contract audits, our specialists look for known vulnerabilities as well as logical and access 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

Information level is to offer suggestions for improvement of efficacy or security for features with a risk free factor.

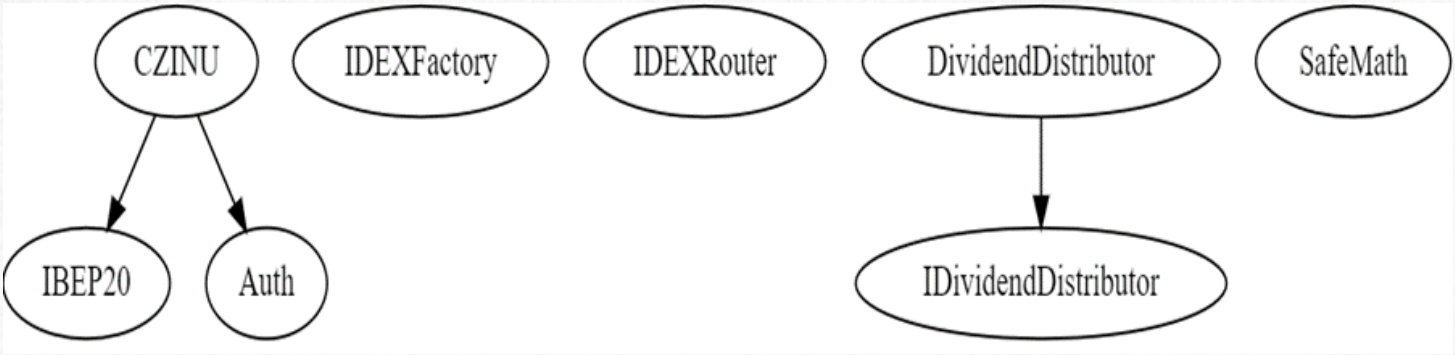
AUDIT SUMMARY

Ownership:

0x98927bf5bd609382af9460f83d36cd3f10e679cf, is the owner of the contract

Contracts & Inheritance Tree:

all of below contracts are in this audit scope



Summary

- A Simple AutoLiquidity ERC20 token, there is currently 10% tax on buy, 10% on sells and 10% on transfers
- Taxes can not exceed 20%, this means 40% tax on buys and sells max
- Owner is able to change max tax and max wallet (max holding) amounts

MANUAL AUDIT

Severity Criteria

Expelee assesses the severity of disclosed vulnerabilities according to a methodology based on
OWASP standards.
Vulnerabilities are divided into three primary risk categories: high, medium, and low.

High-level considerations for vulnerabilities span the followingkey 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 Summary

- High Risk Findings:0
- Medium Risk Findings:2
- Low Risk Findings:0
- Suggestions & discussion: 1
- Gas Optimizations : 0

Medium Risk Findings

Centralization - owner is able change max buy/sell/transfer amount as well as max holding amount.

```
function setMaxWalletPercent(uint256 maxWallPercent)
external onlyOwner() {
        _maxWalletToken = (_totalSupply * maxWallPercent ) /
100;
    }
function setTxLimit(uint256 amount) external authorized {
        _maxTxAmount = amount;
    }
```

Medium Risk Findings

Centralization -owner is able change buy/sell/transfer taxes up to 20% each

```
function setFees(uint256 _liquidityFee, uint256
_reflectionFee, uint256 _marketingFee, uint256 _charityFee,
uint256 _feeDenominator) external authorized {
        liquidityFee = _liquidityFee;
        reflectionFee = _reflectionFee;
        marketingFee = _marketingFee;
        charityFee = _charityFee;
        totalFee =
_liquidityFee.add(_reflectionFee).add(_marketingFee).add
(_charityFee);
        feeDenominator = _feeDenominator;
        require(totalFee < feeDenominator / 5);
}
```

Suggestions

- **using unnecessary library SafeMath: since compiler version is > 0.8.0, all overflow/underflow errors are handled internally by the compiler and using SafeMath only increases the gas usage.**

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 guarantees against the sale of team tokens or the removal of liquidity by the project audited in this document.

Always Do your own research and protect 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. Always 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.