expelee

Building the Futuristic Blockchain Ecosystem

Audit Report FOR

Mastodon 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 with Medium Risk
KYC Verification	Done
Audit Date	16 Nov 2022

- Manual Review (Passed)
- Tested on Forked Pancakeswap v2 on local testnet (Passed)

-Team Expelee

PROJECT DESCRIPTION

Mastodon Inu

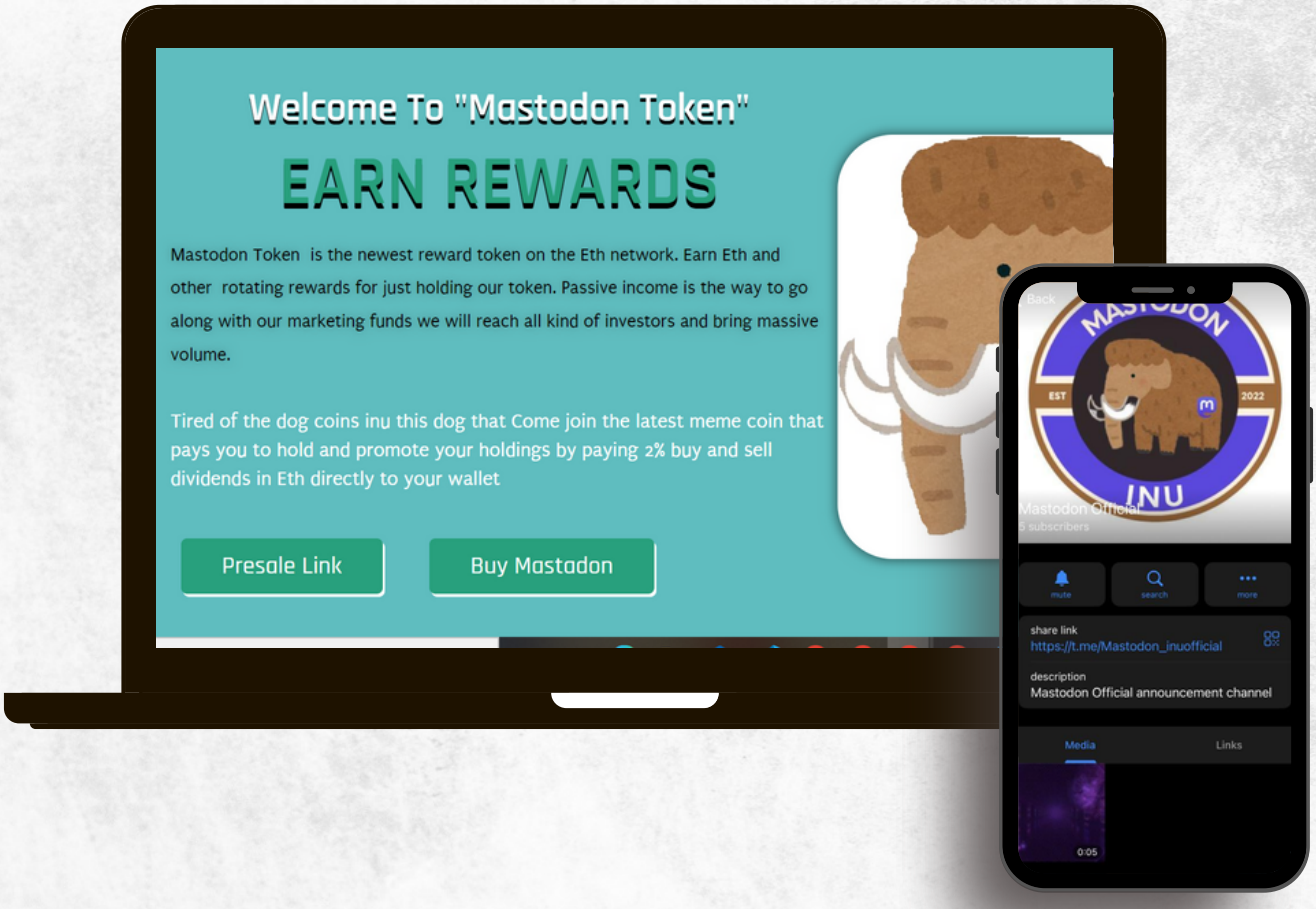
Mastodon Token is the newest reward token on the Eth network. Earn Eth and other rotating rewards for just holding our token. Passive income is the way to go along with our marketing funds we will reach all kind of investors and bring massive volume.

expelee

expelee.com

Social Media Profiles

Mastodon Inu



Https://mastodoninu.com

https://t.me/Mastodon_inuofficial

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

-Team Expelee

CONTRACT DETAILS

Token Name
Mastodon

Symbol
MAS

Network
ETH

Language
Solidity

Contract Address (Verified)
0xb98D03422e977ec1b6c3344Bfa44764304143 59B

Token Type
ERC 20

Decimals
18

Compiler
v0.8.9+commit.e5eed63a

Total Supply
1,000,000,000

Contract SHA-256 Checksum:
be65e827bbb774e0556c42d3fbb183a70e509b5f30fbec9b30b2e6de13eafa05

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

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.

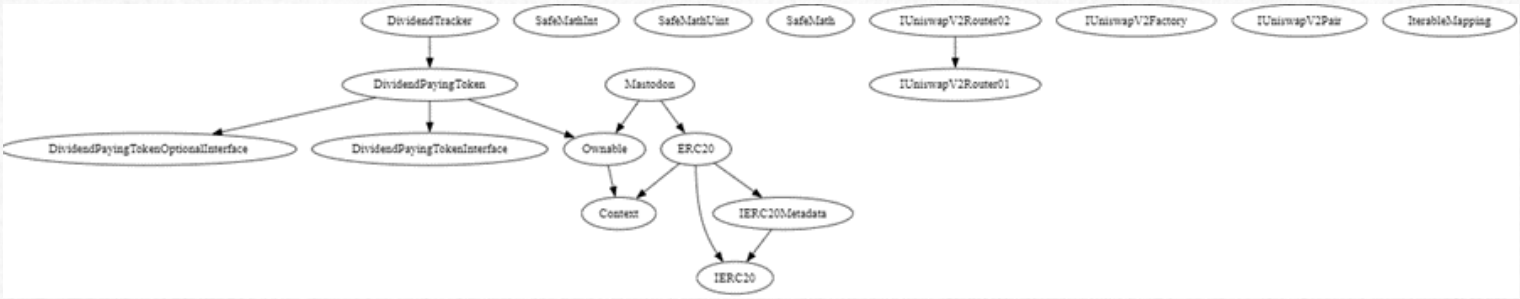
AUDIT SUMMARY

Ownership:

Deployer - 0xC03e47F61F5C8F143c870584D9A92cd170447558

Contracts & Inheritance Tree:

Mastodon token is inherited from this contracts



Summary

- A Reward Paying Token, You Can Choose Your Costume Reward Token Between Allowed Tokens (ETH, USDC, Doge, XRP, BTC, USDT, BUSD)
- Owner is not able to set buy / sell taxes each more than 20% (40% max in total buy + sell)
- You Can Use Up To Your Earned ETH Reward's For Buying MAS Tokens With 0 Fee
- Anti-Sniper Implementation, which doesnt allow snipers to spend more than a limited amount of gas price
- Owner is able to set max buy/holding amount but not less than 1% of total supply.
- - Owne is able to set a max buy/sell/transfer amount, but not less than 0.5% of total supply
- Owner is not able to mint new tokens
- Owner is not able to disable trading

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:4
- Low Risk Findings:1
- Suggestions & discussion: 0
- Gas Optimizations : 1

Medium Risk Findings

Centralization – Owner is able to set a max amount for buying/holding amount. this amount can not be less than 1% of total supply.

```
 function updateMaxWalletAmount(uint256 newNum) external onlyOwner {
      require(newNum > (totalSupply() * 1 / 95)/1e18, "Cannot set maxWallet lower
than 1%");
      maxWallet = newNum * (10**18);
}
```

Centralization - Owner is able to set a max amount for buying/selling/transferring amount. this amount can't be less than 0.5% of total supply.

```
 function updateMaxAmount(uint256 newNum) external onlyOwner {
      require(newNum > (totalSupply() * 5 / 1000)/1e18, "Cannot set
maxTransactionAmount  lower than 0.5%");
      maxTransactionAmount = newNum * (10**18);


}
```

Centralization – Liquidity tokens received from auto-liquidity are sent to an EOA (externally owned account), a malicious owner or hacker who got access to this account pk can use those tokens to remove a portion of liquidity.

Suggestions: burn generated LP tokens

Logical - - router address is remaining same in _transfer function even if its updated (_transfer):

```
if(!tocustomtest){
        try dividendTracker.setRewardToken(to, defaultToken,
 address(0x7a250d5630B4cF539739dF2C5dAcb4c659F2488D)) {} catch {}
}
```

Suggestions: use the variable that stores router address

Low Risk Findings

Centralization - Owner is able to set buy fees and sell fees up to 20%. this means up to maximum 40% on a buy & sell if all taxes are set to max

Gas Optimizations

- High Impact: too many read from storage at _transfer function, buyTotalFees & sellTotalFees are 6 times from storage, this costs 12,600 gas, while savign them to a memory variable and accessing the later only costs 30 gas

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.