expelee

Building the Futuristic **Blockchain Ecosystem**

Audit Report FOR

ETH Moon

OVERVIEW

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	14 Sep 2022

Be aware that smart contracts deployed on the blockchain aren't resistant to internal exploit, external vulnerability, or hack. For a detailed understanding of risk severity, source code vulnerability, functional hack, and audit disclaimer, kindly refer to the audit.

- Team Expelee

PROJECT DESCRIPTION

ETH Moon

LOW TAX 5%, SAFU token created by Pinksale,
No Team Tokens only 10% for airdrop, No Private Sale
Future partnership with big projects, Ama's on major
channels, like Gollums, Venom, Caesar and
WhaleCoinTalk, Free to earn, P2E, PVP, Staking, Hero
Upgrade, LAND, Marketplace
on eth CMC & CG

ethmoontoken.com

ethmoonchat

ethmoon2022

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

- Team Expelee

CONTRACT DETAILS

Contract Name

LiquidityGeneratorToken

Optimization

Yes with 200 runs

Contract Address

0xD68cAb56DEA416a80F00277d429592106aeF1baC

Network

BSC

Language

Solidity

Total Supply

100,000,000,000 ethm

Decimals

9

Compiler

v0.8.4+commit.c7e474f2

License

MIT License

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
Creator Address	0xee058d35d0f1687a8cf3179beb850962a9ff769f
Creator Balance	100,000,000,000 ethm
Owner Address	0xee058d35d0f1687a8cf3179beb850962a9ff769f
Mint	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.

MANUAL AUDIT

Simple & Quick Overview:

- Contract is forked from **Safemoon** with small
- changesOwner is **not** able to set taxes over 25%
- Reflection token, gives **ethm** reflections
- Owner is not able to blacklist an arbitrary address
- Owner is not able to pause trades
- Fair Launch on Pinksale with minimum 5BNB contribution
- **11% of tokens are unlocked at the time of writing this report**

Centralization Risks:

- **Medium** - Owner is able to set taxes up to 25%, using **setCharityFeePercent**, **setLiquidityFeePercent** and **setTaxFeePercent** functions

Logical Issues

- **Low:** wrong error message at this require statement:

```
function includeInReward(address account) external onlyOwner {
        require(_isExcluded[account], "Account is already excluded");
```

is account is not excluded, then error message should be something like:
"Account is already included"

Gas Optimizations

[GO-1] : The conditior`! isExcluded[sender]  &&  ! isExcluded[recipient]` can be included in else

Recommendation:

the following code can be
removed

```
} else if (!_isExcluded[sender] && !_isExcluded[recipient]) {
            _transferStandard(sender, recipient, amount);
```

[GO-2] : Variables _tTotal , numTokensSellToAddToLiquidity , _name , _symbol and _decimals could bedeclaredas constant since these state variables are never to be changed

Suggestions:

[S-1] : if `contractTokenBalance > numTokensSellToAddToLiquidity` then assigning **contractTokenBalance** to **numTokensSellToAddToLiquidity** again may cause some of the tokens tobe stucked in contract

```
if (overMinTokenBalance &&
            !inSwapAndLiquify &&
            from != uniswapV2Pair &&
            swapAndLiquifyEnabled
        ) {
            contractTokenBalance = numTokensSellToAddToLiquidity;
            //add liquidity
            swapAndLiquify(contractTokenBalance);
        }
```

Recommendation:

delete `contractTokenBalance = numTokensSellToAddToLiquidity;`

[S-2] : The return values of function addLiquidityETH are not properly handled.

```
uniswapV2Router.addLiquidityETH{value: ethAmount}(
            address(this),
            tokenAmount,
            0, // slippage is unavoidable
            0, // slippage is unavoidable
            address(0xdead),
            block.timestamp
);
```

Recommendation:

We recommend using variables to receive the return value of the functions mentioned above and handle both success and failure cases if needed by the business logic

[S-3] : In contract , there are a bunch of functions that can change state variables. However, these function do not emit event to pass the changes out of chain

[S-4] : typo at this event:

```
event SwapAndLiquify(
        uint256 tokensSwapped,
        uint256 ethReceived,
        uint256 tokensIntoLiqudity
);
```

tokensIntoLiqudity must be **tokensIntoLiquidity**

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.