



Building the Futuristic **Blockchain Ecosystem**

SECURITY AUDIT REPORT

Lust

TOKEN OVERVIEW

Risk Findings

Severity	Found
● High	0
● Medium	2
● Low	0
● Informational	1

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

02	Token Overview	_____
03	Table of Contents	_____
04	Overview	_____
05	Contract Details	_____
06	Audit Methodology	_____
07	Vulnerabilities Checklist	_____
08	Risk Classification	_____
09	Inheritance Trees	_____
10	Function Details	_____
12	Testnet Version	_____
14	Manual Review	_____
15	Medium Risk Finding	_____
17	Informational Risk Finding	_____
18	About Expelee	_____
19	Disclaimer	_____

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	27 July 2023

CONTRACT DETAILS

Token Name: Lust

Symbol: \$Lust

Network: Ethereum

Language: Solidity

Contract Address:

0x6f9F662b9A25eFD411F13f8fE8E71E9A9C56aD42

Total Supply:69,000,000

Owner's Wallet:

0x7d44dA450Ff624de9EC0781FDC1383983387aB7f

Deployer's Wallet:

0x5213778DE1278789E0e99Ca98a4e47E2EdCE4695

Testnet.

<https://testnet.bscscan.com/token/0xa8dF728D0DF77e749216C9440370d8F95d8c3031>

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 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.

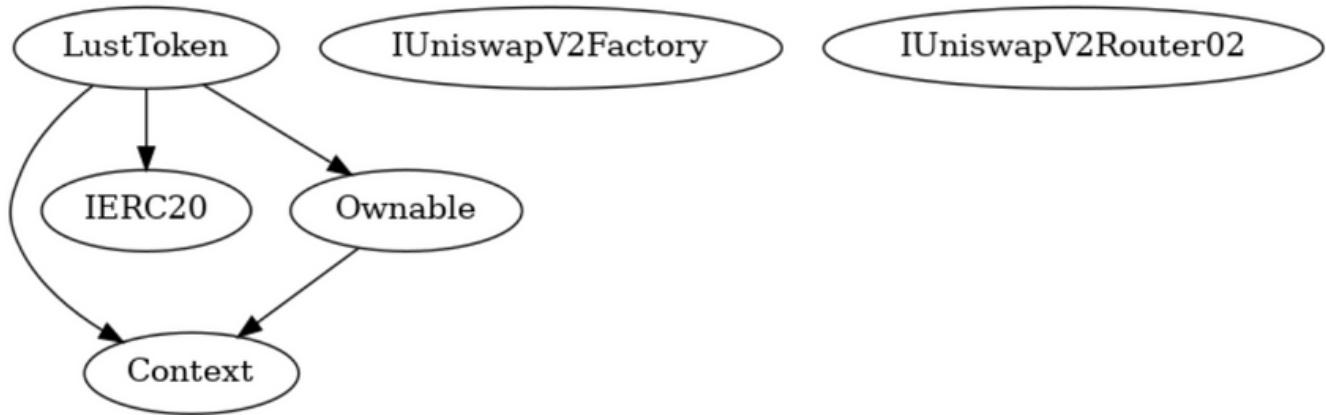
Low Risk

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

Informational

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

INHERITANCE TREES



FUNCTION DETAILS

```

| Contract |      Type      |      Bases      |      |      |
|-----|-----|-----|-----|-----|
|  L  | **Function Name** | **Visibility** | **Mutability** | **Modifiers** |
|||||
| **Context** | Implementation | |||
|  L  | _msgSender | Internal  | ||
|||||
| **IERC20** | Interface | |||
|  L  | totalSupply | External  | |NO  |
|  L  | balanceOf | External  | |NO  |
|  L  | transfer | External  |  |NO  |
|  L  | allowance | External  | |NO  |
|  L  | approve | External  |  |NO  |
|  L  | transferFrom | External  |  |NO  |
|||||
| **Ownable** | Implementation | Context |||
|  L  | <Constructor> | Public  |  |NO  |
|  L  | owner | Public  | |NO  |
|  L  | transferOwnership | Public  |  |onlyOwner |
|  L  | _transferOwnership | Internal  |  ||
|  L  | renounceOwnership | Public  |  |onlyOwner |
|||||
| **IUniswapV2Factory** | Interface | |||
|  L  | createPair | External  |  |NO  |
|||||
| **IUniswapV2Router02** | Interface | |||
|  L  | swapExactTokensForETHSupportingFeeOnTransferTokens | External  |  |NO  |
|  L  | factory | External  | |NO  |
|  L  | WETH | External  | |NO  |
|||||
| **LustToken** | Implementation | Context, IERC20, Ownable |||
|  L  | <Constructor> | Public  |  |NO  |
|  L  | name | Public  | |NO  |
|  L  | symbol | Public  | |NO  |
|  L  | decimals | Public  | |NO  |
|  L  | totalSupply | Public  | |NO  |
|  L  | balanceOf | Public  | |NO  |
|  L  | transfer | Public  |  |NO  |

```

FUNCTION DETAILS

```

|  ↳ | allowance | Public  !  | | NO  !  |
|  ↳ | approve | Public  !  | 🔴 | NO  !  |
|  ↳ | transferFrom | Public  !  | 🔴 | NO  !  |
|  ↳ | _approve | Private 🗝️ | 🔴 | |
|  ↳ | ExcludeWalletFromFees | External  !  | 🔴 | onlyOwner |
|  ↳ | IncludeWalletinFees | External  !  | 🔴 | onlyOwner |
|  ↳ | ChangeTax | External  !  | 🔴 | onlyOwner |
|  ↳ | ChangeMinSwap | External  !  | 🔴 | onlyOwner |
|  ↳ | airdropTokens | External  !  | 🔴 | NO  !  |
|  ↳ | ChangeMarketingWalletAddress | External  !  | 🔴 | onlyOwner |
|  ↳ | _transfer | Private 🗝️ | 🔴 | |
|  ↳ | <Receive Ether> | External  !  | 💵 | NO  !  |

```

Legend

```

| Symbol | Meaning |
|:-----:|:-----|
| 🔴      | Function can modify state |
| 💵      | Function is payable |

```

TESTNET VERSION

Adding Liquidity 

Tx:

<https://testnet.bscscan.com/tx/0x702aa56d0098502712a5c62d5f463bfce0d61d826c0c78383f5d01b499178812>

=====

Buying when excluded from fees 

Tx (0% tax):

<https://testnet.bscscan.com/tx/0x736c5e54833716b273763a70365d9fa97dae45589cc3971307653a2aa3aef94d>

=====

Selling when excluded from fees 

Tx (0% tax):

<https://testnet.bscscan.com/tx/0x5e479dedf2d2a0bc9ed9627de02c2e4dafb2157a5764d478ed461c82e9722048>

=====

Transferring when excluded from fees 

Tx (0% tax):

<https://testnet.bscscan.com/tx/0x0201866bd5d4f1a3bf95b140e514a20a8b0d663e86a684a2a81bc6daa19a7e9f>

=====

Buying 

Tx (0-2% tax):

<https://testnet.bscscan.com/tx/0x0201866bd5d4f1a3bf95b140e514a20a8b0d663e86a684a2a81bc6daa19a7e9f>

TESTNET VERSION

Selling ✓

Tx (0-2% tax):

<https://testnet.bscscan.com/tx/0xf1ba6e27acad6046628e34842e6a6f86d8c2dda43dee11cb97a2e33a1f943282>

=====

Transferring ✓

Tx(0% tax):

<https://testnet.bscscan.com/tx/0xd9d4c40fd55a6b9372a6d462828381db92c2ee7ad9d4f2ae1898cb38c02f533d>

=====

Internal swap (BNB to marketing wallet | reward token to dividend tracker | reward distribution) ✓

Tx:

<https://testnet.bscscan.com/address/0xd9db2810bc786eb788af46b52a7fbef0476433d7#internaltx>

MANUAL REVIEW

Severity Criteria

Expelee assesses the severity of disclosed vulnerabilities according to methodology based on OWASP standards.

Vulnerabilities are divided into three primary risk categories:

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			

MEDIUM RISK FINDING

Can't include wallets in fee

Category: Logical

Status: Open

Impact: Medium

Overview:

includeWalletinFees should be used in order to include a wallet in fees. However this function is excluding "wallet" from fees.

```
function IncludeWalletinFees(address wallet) external onlyOwner {  
    _isExcludedFromFees[wallet] = true;  
}
```

Suggestion:

the function should be implemented in this way:

```
function IncludeWalletinFees(address wallet) external onlyOwner {  
    _isExcludedFromFees[wallet] = false;  
}
```

MEDIUM RISK FINDING

Marketing wallet rejecting Ethereum

Category: Centralization

Status: Open

Impact: Medium

Overview:

Marketing wallet could be set to a contract that rejects receiving ethereum. In this case internal swaps will be reverted since marketing wallet (or ETH destination) is not accepting ether.

Suggestion:

- Receive Ether in the contract and use a low-level call for sending ethereum to the marketing wallet.
- Use a try-catch block in order to handle potential reverts upon receiving ETH

INFORMATINAL RISK FINDING

Stuck ethereum and ERC20 tokens

Category: Missing Logic

Status: Open

Impact: Medium

Overview:

Marketing wallet could be set to a contract that rejects receiving etehreum. In this case internal swaps will be reverted since marketing wallet (or ETH destination) is not accepting ether.

Suggestion:

- Receive Ether in the contract and use a low-level call for sending ethereum to the marketing wallet.
- Use a try-catch block in order to handle potential reverts upon receiving ETH

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.

The logo for Expelee, featuring the word "expelee" in a stylized font. The "ex" is in white, and "pelee" is in orange. The letters are bold and modern.

Building the Futuristic **Blockchain Ecosystem**