



Building the Futuristic **Blockchain** Ecosystem

SECURITY AUDIT REPORT

DogeX

TOKEN OVERVIEW

Risk Findings

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

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	
17	Testnet Version	
19	Manual Review	
24	About Expelee	
25	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 With High Risk
KYC Verification	-
Audit Date	25 Aug 2023

CONTRACT DETAILS

Token Name: DogeX

Symbol: DogeX

Network: Binance Smart Chain

Language: Solidity

Contract Address:

0xA967fff8002219e0cb772634d59327207e6591d4

Total Supply: 1,000,000,000,000

Owner's Wallet:

0x9CeCecb9AaDD8f960956c5D19d584D19A51543F1

Deployer's Wallet:

0x9CeCecb9AaDD8f960956c5D19d584D19A51543F1

Testnet.

<https://testnet.bscscan.com/address/0x2dde7E00389bD8FeB596cCF3ba8158a43e792CA0>

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 Zeppelin 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.

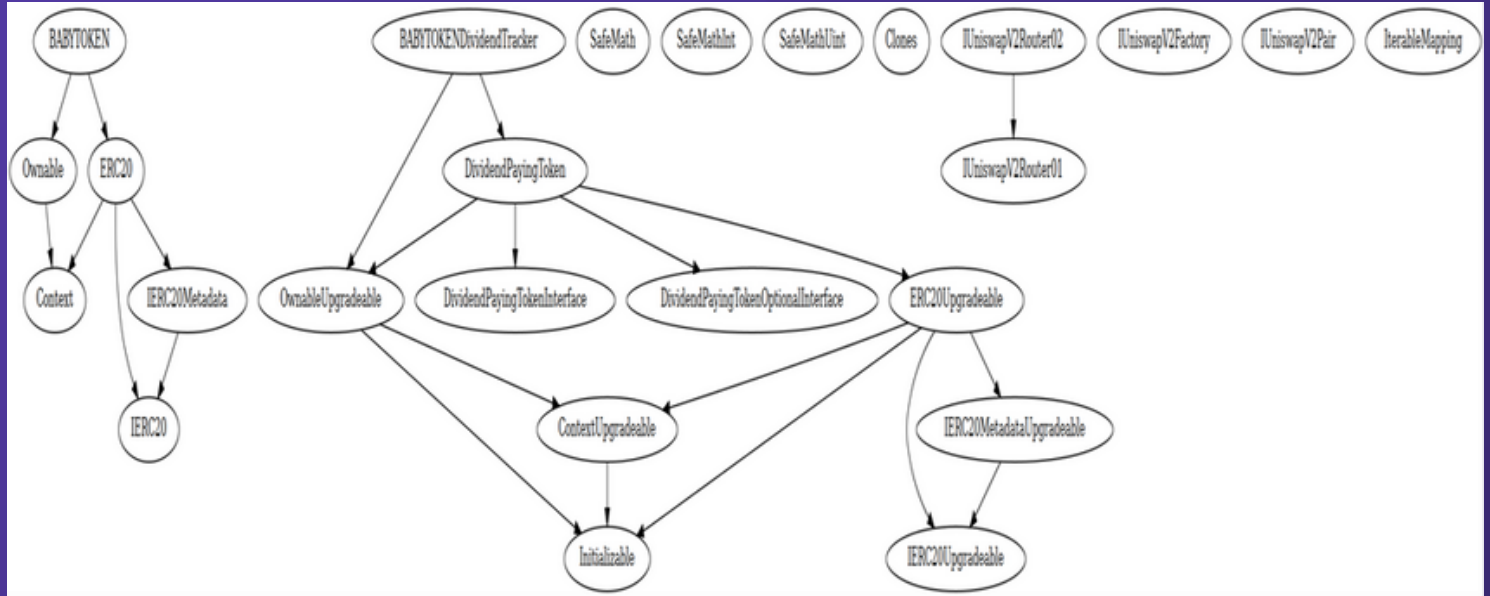
Low Risk

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 | 🔒 | | |
| L | _msgData | Internal | 🔒 | | |
|
|
| **Ownable** | Implementation | Context ||| |
| L | <Constructor> | Public | ! | ● | NO ! |
| L | owner | Public | ! | | NO ! |
| L | renounceOwnership | Public | ! | ● | onlyOwner |
| L | transferOwnership | Public | ! | ● | onlyOwner |
| L | getTime | Public | ! | | NO ! |
| L | _transferOwnership | Internal | 🔒 | ● | |
|
|
| **Initializable** | Implementation | |||
|
|
| **ContextUpgradeable** | Implementation | Initializable ||| |
| L | __Context_init | Internal | 🔒 | ● | initializer |
| L | __Context_init_unchained | Internal | 🔒 | ● | initializer |
| L | _msgSender | Internal | 🔒 | | |
| L | _msgData | Internal | 🔒 | | |
|
|
| **OwnableUpgradeable** | Implementation | Initializable, ContextUpgradeable ||| |
| L | __Ownable_init | Internal | 🔒 | ● | initializer |
| L | __Ownable_init_unchained | Internal | 🔒 | ● | initializer |
| L | owner | Public | ! | | NO ! |
| L | renounceOwnership | Public | ! | ● | onlyOwner |
| L | transferOwnership | Public | ! | ● | onlyOwner |
| L | _setOwner | Private | 🔒 | ● | |
|
|
| **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 ! |
|
|
| **IERC20Metadata** | Interface | IERC20 ||| |
| L | name | External | ! | | NO ! |
| L | symbol | External | ! | | NO ! |
| L | decimals | External | ! | | NO ! |
|
|
| **SafeMath** | Library | ||| |
| L | add | Internal | 🔒 | | |
| L | sub | Internal | 🔒 | | |

```

FUNCTION DETAILS

```

| L | sub | Internal | 🔒 | | | |
| L | mul | Internal | 🔒 | | |
| L | div | Internal | 🔒 | | |
| L | div | Internal | 🔒 | | |
| L | mod | Internal | 🔒 | | |
| L | mod | Internal | 🔒 | | |
|||||
| **SafeMathInt** | Library | |||
| L | mul | Internal | 🔒 | | |
| L | div | Internal | 🔒 | | |
| L | sub | Internal | 🔒 | | |
| L | add | Internal | 🔒 | | |
| L | abs | Internal | 🔒 | | |
| L | toUint256Safe | Internal | 🔒 | | |
|||||
| **SafeMathUint** | Library | |||
| L | toInt256Safe | Internal | 🔒 | | |
|||||
| **Clones** | Library | |||
| L | clone | Internal | 🔒 | 🔴 | |
| L | cloneDeterministic | Internal | 🔒 | 🔴 | |
| L | predictDeterministicAddress | Internal | 🔒 | | |
| L | predictDeterministicAddress | Internal | 🔒 | | |
|||||
| **ERC20** | Implementation | Context, IERC20, IERC20Metadata |||
| 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 | 🔴 |
| L | allowance | Public | 🔴 | | NO | 🔴 |
| L | approve | Public | 🔴 | 🔴 | NO | 🔴 |
| L | transferFrom | Public | 🔴 | 🔴 | NO | 🔴 |
| L | increaseAllowance | Public | 🔴 | 🔴 | NO | 🔴 |
| L | decreaseAllowance | Public | 🔴 | 🔴 | NO | 🔴 |
| L | _transfer | Internal | 🔒 | 🔴 | |
| L | _mint | Internal | 🔒 | 🔴 | |
| L | _burn | Internal | 🔒 | 🔴 | |
| L | _approve | Internal | 🔒 | 🔴 | |
| L | _beforeTokenTransfer | Internal | 🔒 | 🔴 | |
|||||
| **IUniswapV2Router01** | Interface | |||
| L | factory | External | 🔴 | | NO | 🔴 |
| L | WETH | External | 🔴 | | NO | 🔴 |

```

FUNCTION DETAILS

```

| L | addLiquidity | External ! | ● | NO ! |
| L | addLiquidityETH | External ! | 🟢 | NO ! |
| L | removeLiquidity | External ! | ● | NO ! |
| L | removeLiquidityETH | External ! | ● | NO ! |
| L | removeLiquidityWithPermit | External ! | ● | NO ! |
| L | removeLiquidityETHWithPermit | External ! | ● | NO ! |
| L | swapExactTokensForTokens | External ! | ● | NO ! |
| L | swapTokensForExactTokens | External ! | ● | NO ! |
| L | swapExactETHForTokens | External ! | 🟢 | NO ! |
| L | swapTokensForExactETH | External ! | ● | NO ! |
| L | swapExactTokensForETH | External ! | ● | NO ! |
| L | swapETHForExactTokens | External ! | 🟢 | NO ! |
| L | quote | External ! | | NO ! |
| L | getAmountOut | External ! | | NO ! |
| L | getAmountIn | External ! | | NO ! |
| L | getAmountsOut | External ! | | NO ! |
| L | getAmountsIn | External ! | | NO ! |
|||||
| **IUniswapV2Router02** | Interface | IUniswapV2Router01 |||
| L | removeLiquidityETHSupportingFeeOnTransferTokens | External ! | ● | NO ! |
| L | removeLiquidityETHWithPermitSupportingFeeOnTransferTokens | External ! | ● | NO ! |
|
| L | swapExactTokensForTokensSupportingFeeOnTransferTokens | External ! | ● | NO ! |
| L | swapExactETHForTokensSupportingFeeOnTransferTokens | External ! | 🟢 | NO ! |
| L | swapExactTokensForETHSupportingFeeOnTransferTokens | External ! | ● | NO ! |
|||||
| **IUniswapV2Factory** | Interface | |||
| L | feeTo | External ! | | NO ! |
| L | feeToSetter | External ! | | NO ! |
| L | getPair | External ! | | NO ! |
| L | allPairs | External ! | | NO ! |
| L | allPairsLength | External ! | | NO ! |
| L | createPair | External ! | ● | NO ! |
| L | setFeeTo | External ! | ● | NO ! |
| L | setFeeToSetter | External ! | ● | NO ! |
|||||
| **IUniswapV2Pair** | Interface | |||
| L | name | External ! | | NO ! |
| L | symbol | External ! | | NO ! |
| L | decimals | External ! | | NO ! |
| L | totalSupply | External ! | | NO ! |
| L | balanceOf | External ! | | NO ! |
| L | allowance | External ! | | NO ! |
| L | approve | External ! | ● | NO ! |
| L | transfer | External ! | ● | NO ! |
| L | transferFrom | External ! | ● | NO ! |

```

FUNCTION DETAILS

```

| L | DOMAIN_SEPARATOR | External ! | [NO ! | |
| L | PERMIT_TYPEHASH | External ! | [NO ! |
| L | nonces | External ! | [NO ! |
| L | permit | External ! | ● [NO ! |
| L | MINIMUM_LIQUIDITY | External ! | [NO ! |
| L | factory | External ! | [NO ! |
| L | token0 | External ! | [NO ! |
| L | token1 | External ! | [NO ! |
| L | getReserves | External ! | [NO ! |
| L | price0CumulativeLast | External ! | [NO ! |
| L | price1CumulativeLast | External ! | [NO ! |
| L | kLast | External ! | [NO ! |
| L | mint | External ! | ● [NO ! |
| L | burn | External ! | ● [NO ! |
| L | swap | External ! | ● [NO ! |
| L | skim | External ! | ● [NO ! |
| L | sync | External ! | ● [NO ! |
| L | initialize | External ! | ● [NO ! |
|||||
| **IterableMapping** | Library | |||
| L | get | Internal 🔒 | ||
| L | getIndexOfKey | Internal 🔒 | ||
| L | getKeyAtIndex | Internal 🔒 | ||
| L | size | Internal 🔒 | ||
| L | set | Internal 🔒 | ● ||
| L | remove | Internal 🔒 | ● ||
|||||
| **DividendPayingTokenInterface** | Interface | |||
| L | dividendOf | External ! | [NO ! |
| L | withdrawDividend | External ! | ● [NO ! |
|||||
| **DividendPayingTokenOptionalInterface** | Interface | |||
| L | withdrawableDividendOf | External ! | [NO ! |
| L | withdrawnDividendOf | External ! | [NO ! |
| L | accumulativeDividendOf | External ! | [NO ! |
|||||
| **IERC20Upgradeable** | 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 ! |
|||||
| **IERC20MetadataUpgradeable** | Interface | IERC20Upgradeable |||
| L | name | External ! | [NO ! |

```


FUNCTION DETAILS

```

| L | symbol | External ! | | NO ! |
| L | decimals | External ! | | NO ! |
|||||
| **ERC20Upgradeable** | Implementation | Initializable, ContextUpgradeable,
IERC20Upgradeable, IERC20MetadataUpgradeable |||
| L | __ERC20_init | Internal 🔒 | ● | initializer |
| L | __ERC20_init_unchained | Internal 🔒 | ● | initializer |
| L | name | Public ! | | NO ! |
| L | symbol | Public ! | | NO ! |
| L | decimals | Public ! | | NO ! |
| L | totalSupply | Public ! | | NO ! |
| L | balanceOf | Public ! | | NO ! |
| L | transfer | Public ! | ● | NO ! |
| L | allowance | Public ! | | NO ! |
| L | approve | Public ! | ● | NO ! |
| L | transferFrom | Public ! | ● | NO ! |
| L | increaseAllowance | Public ! | ● | NO ! |
| L | decreaseAllowance | Public ! | ● | NO ! |
| L | _transfer | Internal 🔒 | ● | |
| L | _mint | Internal 🔒 | ● | |
| L | _burn | Internal 🔒 | ● | |
| L | _approve | Internal 🔒 | ● | |
| L | _beforeTokenTransfer | Internal 🔒 | ● | |
| L | _afterTokenTransfer | Internal 🔒 | ● | |
|||||
| **DividendPayingToken** | Implementation | ERC20Upgradeable, OwnableUpgradeable,
DividendPayingTokenInterface, DividendPayingTokenOptionalInterface |||
| L | __DividendPayingToken_init | Internal 🔒 | ● | initializer |
| L | distributeCAKEDividends | Public ! | ● | onlyOwner |
| L | withdrawDividend | Public ! | ● | NO ! |
| L | _withdrawDividendOfUser | Internal 🔒 | ● | |
| L | dividendOf | Public ! | | NO ! |
| L | withdrawableDividendOf | Public ! | | NO ! |
| L | withdrawnDividendOf | Public ! | | NO ! |
| L | accumulativeDividendOf | Public ! | | NO ! |
| L | _transfer | Internal 🔒 | ● | |
| L | _mint | Internal 🔒 | ● | |
| L | _burn | Internal 🔒 | ● | |
| L | _setBalance | Internal 🔒 | ● | |
|||||
| **BABYTOKENDividendTracker** | Implementation | OwnableUpgradeable,
DividendPayingToken |||
| L | initialize | External ! | ● | initializer |
| L | _transfer | Internal 🔒 | | |
| L | withdrawDividend | Public ! | | NO ! |
| L | excludeFromDividends | External ! | ● | onlyOwner |

```

FUNCTION DETAILS

```

| | isExcludedFromDividends | Public | ! | [NO !] | |
| | updateClaimWait | External | ! | ● | onlyOwner |
| | updateMinimumTokenBalanceForDividends | External | ! | ● | onlyOwner |
| | getLastProcessedIndex | External | ! | [NO !] |
| | getNumberOfTokenHolders | External | ! | [NO !] |
| | getAccount | Public | ! | [NO !] |
| | getAccountAtIndex | Public | ! | [NO !] |
| | canAutoClaim | Private | 🔒 | | |
| | setBalance | External | ! | ● | onlyOwner |
| | process | Public | ! | ● | [NO !] |
| | processAccount | Public | ! | ● | onlyOwner |
|||||
| | **BABYTOKEN** | Implementation | ERC20, Ownable |||
| | <Constructor> | Public | ! | 🟢 | ERC20 |
| | <Receive Ether> | External | ! | 🟢 | [NO !] |
| | updateMinimumTokenBalanceForDividends | Public | ! | ● | onlyOwner |
| | getMinimumTokenBalanceForDividends | External | ! | [NO !] |
| | updateUniswapV2Router | Public | ! | ● | onlyOwner |
| | excludeFromFees | Public | ! | ● | onlyOwner |
| | excludeMultipleAccountsFromFees | Public | ! | ● | onlyOwner |
| | setAirdropNumbs | Public | ! | ● | onlyOwner |
| | setKing | Public | ! | ● | onlyOwner |
| | setMarketingWallet | External | ! | ● | onlyOwner |
| | setAutomatedMarketMakerPair | Public | ! | ● | onlyOwner |
| | _setAutomatedMarketMakerPair | Private | 🔒 | ● | |
| | updateGasForProcessing | Public | ! | ● | onlyOwner |
| | updateClaimWait | External | ! | ● | onlyOwner |
| | getClaimWait | External | ! | [NO !] |
| | getTotalDividendsDistributed | External | ! | [NO !] |
| | isExcludedFromFees | Public | ! | [NO !] |
| | withdrawableDividendOf | Public | ! | [NO !] |
| | dividendTokenBalanceOf | Public | ! | [NO !] |
| | excludeFromDividends | External | ! | ● | onlyOwner |
| | isExcludedFromDividends | Public | ! | [NO !] |
| | getAccountDividendsInfo | External | ! | [NO !] |
| | getAccountDividendsInfoAtIndex | External | ! | [NO !] |
| | processDividendTracker | External | ! | ● | [NO !] |
| | claim | External | ! | ● | [NO !] |
| | getLastProcessedIndex | External | ! | [NO !] |
| | getNumberOfDividendTokenHolders | External | ! | [NO !] |
| | swapManual | Public | ! | ● | onlyOwner |
| | setSwapAndLiquifyEnabled | Public | ! | ● | onlyOwner |
| | setSwapTokensAtAmount | Public | ! | ● | onlyOwner |
| | setDeadWallet | Public | ! | ● | onlyOwner |
| | setBuyTaxes | External | ! | ● | onlyOwner |

```

FUNCTION DETAILS

```

| L | setSelTaxes | External ! | ● | onlyOwner |
| L | _transfer | Internal 🔒 | ● | |
| L | swapAndSendToFee | Private 🔒 | ● | |
| L | swapAndLiquify | Private 🔒 | ● | |
| L | swapTokensForEth | Private 🔒 | ● | |
| L | swapTokensForCake | Private 🔒 | ● | |
| L | addLiquidity | Private 🔒 | ● | |
| L | swapAndSendDividends | Private 🔒 | ● | |

```

Legend

Symbol	Meaning
●	Function can modify state
🔒	Function is payable

TESTNET VERSION

Adding Liquidity ✓

Tx:

<https://testnet.bscscan.com/tx/0xc2c3b0fc8647b76094a65760b4ca2e6dbbee2af76dd483dbbdb4ae3243f91f04>

=====

Buying when excluded from fees ✓

Tx (0% tax):

<https://testnet.bscscan.com/tx/0xe4f0246ca1361e0e4d86d26826bee64edd4582cc78d3c42976ad000f64f715cb>

=====

Selling when excluded from fees ✓

Tx (0% tax):

<https://testnet.bscscan.com/tx/0xab6347cd703385191bb95dea144261534d506180719f2ee4f844ae0ae207381>

=====

Transferring when excluded from fees ✓

Tx (0% tax):

<https://testnet.bscscan.com/tx/0x81233d1924b627a0ab65e3b3fa0ef469cb32f2c93ae8f3ca552affb30d343edc>

=====

Buying ✓

Tx (0-25% tax):

<https://testnet.bscscan.com/tx/0x29eba251fc08c158e990bf9622c26f26ebedaab67121bf2c8b8ae1a515bf25d5>

=====

Selling ✓

Tx (0-25% tax):

<https://testnet.bscscan.com/tx/0x9a6ec5de5cb5ce7cbb9a04a6a406ca621036ef7260df009e6f8d4fb21700c95e>

TESTNET VERSION

Transferring ✓

Tx (25% tax):

<https://testnet.bscscan.com/tx/0xf12533e733ca171c5e63a73bf3f08f2295d8c99161573f254b943989795b512a>

=====

Internal swap (BUSD sent to marketing wallet | Auto-liquidity | Burning | rewards) ✓

Tx:

<https://testnet.bscscan.com/tx/0xf12533e733ca171c5e63a73bf3f08f2295d8c99161573f254b943989795b512a>

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			

HIGH RISK FINDING

Owner is able to update router

Category: **Centralization**

Status: Open

Severity: **High**

Overview:

The contract owner is able to update router contract which is used for performing internal swap. Updating router to a malicious contract could cause errors when performing internal swap / auto-liquidity / reward distribution

```
function updateUniswapV2Router(address newAddress)
public onlyOwner {
    require(
        newAddress != address(uniswapV2Router),
        "The router already has that address"
    );
    emit UpdateUniswapV2Router(newAddress,
address(uniswapV2Router));
    uniswapV2Router = IUniswapV2Router02(newAddress);
    address _uniswapV2Pair =
IUniswapV2Factory(uniswapV2Router.factory())
.createPair(address(this), uniswapV2Router.WETH());
    uniswapPair = _uniswapV2Pair;
}
```

Suggestion: Ensure that uniswapV2Router is immutable

HIGH RISK FINDING

Owner is able to update dead blocks

Category: **Centralization**

Status: Open

Severity: **High**

Overview:

The contract owner is able to set dead block to any number, if current block is a dead block, all purchased tokens will be sent to an EOA (100% tax)

```
function setKing(uint256 newValue) public onlyOwner {  
    kill = newValue;  
}
```

Suggestion:

set an upper bound for maximum amount of dead blocks.

```
function setKing(uint256 newValue) public onlyOwner {  
    require(newValue <= 5, "Can not set dead block more  
than 5");  
    kill = newValue;  
}
```

MEDIUM RISK FINDING

Excessive fees

Category: **Centralization**

Status: Open

Severity: **Medium**

Overview:

Owner is able to set up to 25% percent fee on buy / sell transactions

```
function setBuyTaxes(  
    uint256 liquidity,  
    uint256 rewardsFee,  
    uint256 marketingFee,  
    uint256 deadFee  
) external onlyOwner {...}
```

```
function setSelTaxes(  
    uint256 liquidity,  
    uint256 rewardsFee,  
    uint256 marketingFee,  
    uint256 deadFee  
) external onlyOwner {...}
```

Suggestion:

To mitigate this risk, ensure that fees can not exceed 10% on buy/sell/transfers

MEDIUM RISK FINDING

LP tokens sent to EOA

Category: **Centralization**

Status: Open

Severity: **Medium**

Overview: New LP tokens are sent to an EOA (Externally owned account). This LP tokens can be used to remove a portion of liquidity pool.

```
function addLiquidity(uint256 tokenAmount, uint256 ethAmount)
private {
    // approve token transfer to cover all possible scenarios
    _approve(address(this), address(uniswapV2Router), tokenAmount);
    // add the liquidity
    try
    uniswapV2Router.addLiquidityETH{value: ethAmount}(
        address(this),
        tokenAmount,
        0, // slippage is unavoidable
        0, // slippage is unavoidable
        receiveAddress,
        block.timestamp
    )
    {} catch {
        emit Failed_addLiquidityETH();
    }
}
```

Suggestion: It is suggested to lock or burn new LP tokens

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](https://twitter.com/expeleeofficial)

 [expelee](https://medium.com/expelee)

 [Expelee](https://t.me/Expelee)

 [expelee](https://in.linkedin.com/expelee)

 [expelee_official](https://www.instagram.com/expelee_official)

 [expelee-co](https://github.com/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**