expelee

Building the Futuristic Blockchain Ecosystem

# SECURITY AUDIT REPORT

## DOODIE

# TOKEN OVERVIEW

## Risk Findings

| Severity | Found |
|---|---|
| High | 4 |
| Medium | 2 |
| Low | 0 |
| Informational | 0 |

## Centralization Risks

| Owner Privileges | Description |
|---|---|
| Can Owner Set Taxes >25% ? | 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 |

expelee

# TABLE OF CONTENTS

# 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** | **26 July 2023** |

# CONTRACT DETAILS

Token Name: DoodieMan

Symbol: DOODIE

Network: Binance Smart Chain

Language: Solidity

Contract Address:
0x6f9F662b9A25eFD411F13f8fE8E71E9A9C56aD42

Total Supply: 10,000,000,000

Owner's Wallet:
0x9e84d30c449889500710F436eAeBC8F732ac459d

Deployer's Wallet:
0x0000000000000000000000000000000000000000

Testnet.
https://testnet.bscscan.com/token/0x2B25eCe78368419b10e6E2978d4d5250954e5Cb9

# 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 acces 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

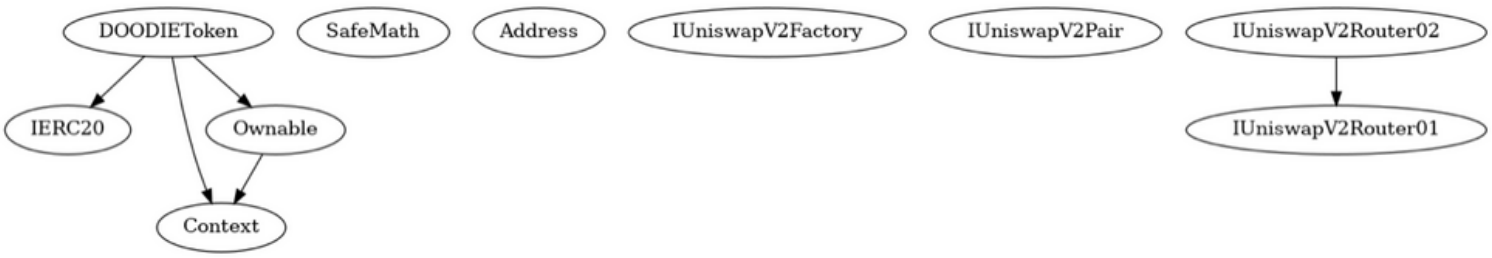
Issues on this level are minor details and warning that can remain unfixed.

## Informational

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

expelee

# INHERITANCE TREES



expelee

# FUNCTION DETAILS

| Contract | Type | Bases | | |
|:---|:---|:---|:---|:---|
| └ | **Function Name** | **Visibility** | **Mutability** | **Modifiers** |
| | | | | |
| **IERC20** | Interface | | | |
| └ | totalSupply | External ❗ | | NO ❗ |
| └ | balanceOf | External ❗ | | NO ❗ |
| └ | transfer | External ❗ | 🛑 | NO ❗ |
| └ | allowance | External ❗ | | NO ❗ |
| └ | approve | External ❗ | 🛑 | NO ❗ |
| └ | transferFrom | External ❗ | 🛑 | NO ❗ |
| | | | | |
| **SafeMath** | Library | | | |
| └ | add | Internal 🔒 | | |
| └ | sub | Internal 🔒 | | |
| └ | sub | Internal 🔒 | | |
| └ | mul | Internal 🔒 | | |
| └ | div | Internal 🔒 | | |
| └ | div | Internal 🔒 | | |
| └ | mod | Internal 🔒 | | |
| └ | mod | Internal 🔒 | | |
| | | | | |
| **Context** | Implementation | | | |
| └ | _msgSender | Internal 🔒 | | |
| └ | _msgData | Internal 🔒 | | |
| | | | | |
| **Address** | Library | | | |
| └ | isContract | Internal 🔒 | | |
| └ | sendValue | Internal 🔒 | 🛑 | |
| └ | functionCall | Internal 🔒 | 🛑 | |
| └ | functionCall | Internal 🔒 | 🛑 | |
| └ | functionCallWithValue | Internal 🔒 | 🛑 | |
| └ | functionCallWithValue | Internal 🔒 | 🛑 | |
| └ | _functionCallWithValue | Private 🔐 | 🛑 | |
| | | | | |
| **Ownable** | Implementation | Context | | |
| └ | <Constructor> | Public ❗ | 🛑 | NO ❗ |
| └ | owner | Public ❗ | | NO ❗ |
| └ | renounceOwnership | Public ❗ | 🛑 | onlyOwner |

expelee

# FUNCTION DETAILS

| └ | transferOwnership | Public ❗ | 🛑 | onlyOwner |
|||||||
| **IUniswapV2Factory** | Interface | |||
| └ | feeTo | External ❗ | | NO ❗ |
| └ | feeToSetter | External ❗ | | NO ❗ |
| └ | getPair | External ❗ | | NO ❗ |
| └ | allPairs | External ❗ | | NO ❗ |
| └ | allPairsLength | External ❗ | | NO ❗ |
| └ | createPair | External ❗ | 🛑 | NO ❗ |
| └ | setFeeTo | External ❗ | 🛑 | NO ❗ |
| └ | setFeeToSetter | External ❗ | 🛑 | NO ❗ |
|||||||
| **IUniswapV2Pair** | Interface | |||
| └ | name | External ❗ | | NO ❗ |
| └ | symbol | External ❗ | | NO ❗ |
| └ | decimals | External ❗ | | NO ❗ |
| └ | totalSupply | External ❗ | | NO ❗ |
| └ | balanceOf | External ❗ | | NO ❗ |
| └ | allowance | External ❗ | | NO ❗ |
| └ | approve | External ❗ | 🛑 | NO ❗ |
| └ | transfer | External ❗ | 🛑 | NO ❗ |
| └ | transferFrom | External ❗ | 🛑 | NO ❗ |
| └ | DOMAIN_SEPARATOR | External ❗ | | NO ❗ |
| └ | PERMIT_TYPEHASH | External ❗ | | NO ❗ |
| └ | nonces | External ❗ | | NO ❗ |
| └ | permit | External ❗ | 🛑 | NO ❗ |
| └ | MINIMUM_LIQUIDITY | External ❗ | | NO ❗ |
| └ | factory | External ❗ | | NO ❗ |
| └ | token0 | External ❗ | | NO ❗ |
| └ | token1 | External ❗ | | NO ❗ |
| └ | getReserves | External ❗ | | NO ❗ |
| └ | price0CumulativeLast | External ❗ | | NO ❗ |
| └ | price1CumulativeLast | External ❗ | | NO ❗ |
| └ | kLast | External ❗ | | NO ❗ |
| └ | burn | External ❗ | 🛑 | NO ❗ |
| └ | swap | External ❗ | 🛑 | NO ❗ |

expelee

# FUNCTION DETAILS

| └| skim | External ❗ | 🛑 |NO ❗ |
| └| sync | External ❗ | 🛑 |NO ❗ |
| └| initialize | External ❗ | 🛑 |NO ❗ |
||||||
| **IUniswapV2Router01** | Interface | |||
| └| factory | External ❗ | |NO ❗ |
| └| WETH | External ❗ | |NO ❗ |
| └| addLiquidity | External ❗ | 🛑 |NO ❗ |
| └| addLiquidityETH | External ❗ | 💵 |NO ❗ |
| └| removeLiquidity | External ❗ | 🛑 |NO ❗ |
| └| removeLiquidityETH | External ❗ | 🛑 |NO ❗ |
| └| removeLiquidityWithPermit | External ❗ | 🛑 |NO ❗ |
| └| removeLiquidityETHWithPermit | External ❗ | 🛑 |NO ❗ |
| └| swapExactTokensForTokens | External ❗ | 🛑 |NO ❗ |
| └| swapTokensForExactTokens | External ❗ | 🛑 |NO ❗ |
| └| swapExactETHForTokens | External ❗ | 💵 |NO ❗ |
| └| swapTokensForExactETH | External ❗ | 🛑 |NO ❗ |
| └| swapExactTokensForETH | External ❗ | 🛑 |NO ❗ |
| └| swapETHForExactTokens | External ❗ | 💵 |NO ❗ |
| └| quote | External ❗ | |NO ❗ |
| └| getAmountOut | External ❗ | |NO ❗ |
| └| getAmountIn | External ❗ | |NO ❗ |
| └| getAmountsOut | External ❗ | |NO ❗ |
| └| getAmountsIn | External ❗ | |NO ❗ |
||||||
| **IUniswapV2Router02** | Interface | IUniswapV2Router01 |||
| └| removeLiquidityETHSupportingFeeOnTransferTokens | External ❗ | 🛑 |NO ❗ |
| └| removeLiquidityETHWithPermitSupportingFeeOnTransferTokens | External ❗ | 🛑 |NO ❗ |
|
| └| swapExactTokensForTokensSupportingFeeOnTransferTokens | External ❗ | 🛑 |NO ❗ |
| └| swapExactETHForTokensSupportingFeeOnTransferTokens | External ❗ | 💵 |NO ❗ |
| └| swapExactTokensForETHSupportingFeeOnTransferTokens | External ❗ | 🛑 |NO ❗ |
||||||
| **DOODIEToken** | Implementation | Context, IERC20, Ownable |||
| └| <Constructor> | Public ❗ | 🛑 |NO ❗ |
| └| name | Public ❗ | |NO ❗ |
| └| symbol | Public ❗ | |NO ❗ |

# FUNCTION DETAILS

| └ | decimals | Public ❗ | | NO ❗ |
| --- | --- | --- | --- | --- |
| └ | totalSupply | Public ❗ | | NO ❗ |
| └ | balanceOf | Public ❗ | | NO ❗ |
| └ | transfer | Public ❗ | 🛑 | NO ❗ |
| └ | allowance | Public ❗ | | NO ❗ |
| └ | approve | Public ❗ | 🛑 | NO ❗ |
| └ | transferFrom | Public ❗ | 🛑 | NO ❗ |
| └ | increaseAllowance | Public ❗ | 🛑 | NO ❗ |
| └ | decreaseAllowance | Public ❗ | 🛑 | NO ❗ |
| └ | isExcludedFromReward | Public ❗ | | NO ❗ |
| └ | totalFees | Public ❗ | | NO ❗ |
| └ | deliver | Public ❗ | 🛑 | NO ❗ |
| └ | reflectionFromToken | Public ❗ | | NO ❗ |
| └ | tokenFromReflection | Public ❗ | | NO ❗ |
| └ | excludeFromReward | Public ❗ | 🛑 | onlyOwner |
| └ | includeInReward | External ❗ | 🛑 | onlyOwner |
| └ | _transferBothExcluded | Private 🔐 | 🛑 | |
| └ | <Receive Ether> | External ❗ | 💵 | NO ❗ |
| └ | _reflectFee | Private 🔐 | 🛑 | |
| └ | _getValues | Private 🔐 | | |
| └ | _getTValues | Private 🔐 | | |
| └ | _getRValues | Private 🔐 | | |
| └ | _getRate | Private 🔐 | | |
| └ | _getCurrentSupply | Private 🔐 | | |
| └ | _takeLiquidity | Private 🔐 | 🛑 | |
| └ | calculateRewardFee | Private 🔐 | | |
| └ | calculateLiquidityFee | Private 🔐 | | |
| └ | removeAllFee | Private 🔐 | 🛑 | |
| └ | restoreAllFee | Private 🔐 | 🛑 | |
| └ | isExcludedFromFee | Public ❗ | | NO ❗ |
| └ | _approve | Private 🔐 | 🛑 | |
| └ | _transfer | Private 🔐 | 🛑 | |
| └ | swapAndLiquify | Private 🔐 | 🛑 | lockTheSwap |
| └ | swapTokensForEth | Private 🔐 | 🛑 | |
| └ | addLiquidity | Private 🔐 | 🛑 | |

# FUNCTION DETAILS

| └ | _tokenTransfer | Private 🔐 | 🛑 | | |
| └ | _transferStandard | Private 🔐 | 🛑 | | |
| └ | _transferToExcluded | Private 🔐 | 🛑 | | |
| └ | _transferFromExcluded | Private 🔐 | 🛑 | | |
| └ | excludeFromFee | Public ❗ | 🛑 | onlyOwner |
| └ | includeInFee | Public ❗ | 🛑 | onlyOwner |
| └ | setmarketingWallet | External ❗ | 🛑 | onlyOwner |
| └ | setRewardFeePercent | External ❗ | 🛑 | onlyOwner |
| └ | setLiquidityFeePercent | External ❗ | 🛑 | onlyOwner |
| └ | setmarketingFeePercent | External ❗ | 🛑 | onlyOwner |
| └ | setRouterAddress | Public ❗ | 🛑 | onlyOwner |
| └ | setSwapAndLiquifyEnabled | Public ❗ | 🛑 | onlyOwner |

### Legend

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

expelee

# TESTNET VERSION

Adding Liquidity ✅
Tx:
https://testnet.bscscan.com/tx/0xc8f8394c792f3c328d7f5583ae93eebcaefdca75e1aa05709333bb24f8616558

==========

Buying when excluded from fees ✅
Tx (0% tax):
https://testnet.bscscan.com/tx/0x9ceec6c91014ce91656077c7a028e52fb8d03ff929c71ac294863a9167482879

==========

Selling when excluded from fees ✅
Tx (0% tax):
https://testnet.bscscan.com/tx/0x205f9c7812836b34a9994b7ff3e687065ff899e8df6580381f73dc5ae6181c2f

==========

Transferring when excluded from fees ✅
Tx (0% tax):
https://testnet.bscscan.com/tx/0xf988f6f95bb54ca04fc947c1ee727aa30a3e6f3a656a19d6aeb9b234b5eb131a

==========

Buying ✅
Tx (0-100% tax):
https://testnet.bscscan.com/tx/0x8ac952ac57e9553916709f9e7c3f0bd9cf4a8d5de68f1c6b4773254ce09054cc

# TESTNET VERSION

Selling
Tx (0-100% tax):
https://testnet.bscscan.com/tx/0xb67cede54eabe48dd42ee69340949855b8709d8604419bcf45dbacb810885aac

==========

Transferring
Tx(0% tax):
https://testnet.bscscan.com/tx/0x58012d66dae6728290ed88e7157507a60dfbc1b7ebd396b4a62ec9b96be3de02

==========

Internal swap (BNB to marketing wallet | reward token to dividend tracker | reward distribution)
Tx:
https://testnet.bscscan.com/tx/0x58012d66dae6728290ed88e7157507a60dfbc1b7ebd396b4a62ec9b96be3de02

# MANUAL REVIEW

## Severity Criteria

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

Vulnerabilities are dividend into three primary risk categroies:
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

## Unbounded fees

**Category: Centralization**
**Status: Open**
**Impact: High**

**Overview:**
The contract owner is able to set buy/sell/transfer fees up to 100%

```
function setRewardFeePercent(uint256 RewardFee) external onlyOwner {
 _RewardFee = RewardFee;
}

function setLiquidityFeePercent(uint256 liquidityFee) external onlyOwner {
 _liquidityFee = liquidityFee;
}

function setmarketingFeePercent(uint256 marketingFee) external onlyOwner
{
 _marketingFee = marketingFee;
}
```

# HIGH RISK FINDING

**Suggestion:**
Ensure that fees are within a reasonable range. Ussualy 0-10% is suggested.

```
function setRewardFeePercent(uint256 RewardFee) external onlyOwner {
  require(RewardFee <= 10, “Fees should not be higher than 10%’);
 _RewardFee = RewardFee;
}

function setLiquidityFeePercent(uint256 liquidityFee) external onlyOwner {
  require(RewardFee <= 10, “Fees should not be higher than 10%’);
 _liquidityFee = liquidityFee;
}

function setmarketingFeePercent(uint256 marketingFee) external onlyOwner
{
  require(RewardFee <= 10, “Fees should not be higher than 10%’);
 _marketingFee = marketingFee;
}
```

# HIGH RISK FINDING

## Changing router

**Category:** Logical
**Status:** Open
**Impact:** High

**Overview:**
Owner is able to update swap router that is used for performing internal swap. Setting router to a malicious contract could revert internal swaps and eventually whole transfer/sell transaction.

```
function setRouterAddress(address newRouter) public onlyOwner {
 IUniswapV2Router02 _newPancakeRouter = IUniswapV2Router02(newRouter);
 uniswapV2Pair = IUniswapV2Factory(_newPancakeRouter.factory()).createPair(
  address(this),
  _newPancakeRouter.WETH()
 );
 uniswapV2Router = _newPancakeRouter;
}
```

**Suggestion:**
Ensure that router is immutable in order to mitigate this logical issue.

# HIGH RISK FINDING

## Invalid routing to transfer functions

**Category:** Logical
**Status:** Open
**Impact:** High

**Overview:**
at _tokenTransfer function, same condition is checked multiple times and the trnansfer is router to an invalid transfer function.

```
if (
 _isExcluded[sender] &&
 !_isExcluded[recipient] &&
 (sender != uniswapV2Pair && recipient != uniswapV2Pair)
) {
 _transferFromExcluded(sender, recipient, (amount.sub(marketingAmt)));
} else if (
 _isExcluded[sender] &&
 !_isExcluded[recipient] &&
 (sender != uniswapV2Pair && recipient != uniswapV2Pair)
) {
 _transferToExcluded(sender, recipient, (amount.sub(marketingAmt)));
} else if (
 _isExcluded[sender] &&
 !_isExcluded[recipient] &&
 (sender != uniswapV2Pair && recipient != uniswapV2Pair)
) {
 _transferStandard(sender, recipient, (amount.sub(marketingAmt)));
} else if (
 _isExcluded[sender] &&
 !_isExcluded[recipient] &&
 (sender != uniswapV2Pair && recipient != uniswapV2Pair)
) {
```

# HIGH RISK FINDING

```
 _transferBothExcluded(sender, recipient, (amount.sub(marketingAmt)));
} else {
 _transferStandard(sender, recipient, (amount.sub(marketingAmt)));
}
```

**Suggestion:**
ensuer to cover all different conditions and route the transfer into correct transfer function.

# HIGH RISK FINDING

## Same condition checked at _transfer and _tokenTransfer

**Category:** Logical
**Status:** Open
**Impact:** High

**Overview:**
same condition for setting fees is checked twice at _transfer and _tokenTransfer functions

```
if (
 (_isExcludedFromFee[sender] || _isExcludedFromFee[recipient]) ||
 (sender != uniswapV2Pair && recipient != uniswapV2Pair)
) {
 removeAllFee();
} else {
 //Set Fee for Buys
 if (sender == uniswapV2Pair && recipient != address(uniswapV2Router)) {
  _RewardFee = _previousRewardFee;
  _marketingFee = _previousmarketingFee;
  _liquidityFee = _previousLiquidityFee;
 }
 //Set Fee for Sells
 if (recipient == uniswapV2Pair && sender != address(uniswapV2Router)) {
  _RewardFee = _previousRewardFee;
  _marketingFee = _previousmarketingFee;
  _liquidityFee = _previousLiquidityFee;
 }
}
```

**Suggestion:**
Remove this conditions from _tokenTransfer function.

# MEDIUM RISK FINDING

## Transferring fees to marketing wallet

**Category:** Logical
**Status:** Open
**Impact:** Medium

**Overview:**
transferStandard function is used for transferring fees to marketing wallet. This means that if marketing wallet is excluded from reflectinos, its balance doesn't update correctly.

```
_transferStandard(sender, marketingWallet, marketingAmt);
```

**Suggestion:**
check if marketing wallet is excluded from reflections before transferring fees to marketing wallet.

# MEDIUM RISK FINDING

## Owner receiving LP shares

**Category:** Centralization
**Status:** Open
**Impact:** Medium

**Overview:**
After each auto-liquidity (internal swap), owner receives the minted LP tokens. This accumulated LP tokens can be used to remove a portion of liquidity pool. The impact could be little to high depending on this LP tokens and total LP tokens which were initialiy minted

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

**Suggestion:**
Its suggested to burn or Lock 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

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.