

expellee

Building the Futuristic **Blockchain Ecosystem**

SECURITY AUDIT REPORT



BABY TUBBIES

TABLE OF CONTENTS

02 Table of Contents

03 Overview

04 Project Description

05 Social Media Profiles

06 Contract Details

07 Owner Privileges

08 Audit Methodology

09 Vulnerabilities Checklist

10 Risk Classification

11 Inheritance Tree

12 Function Details

19 Manual Review

20 Findings

25 About Expelee

26 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	24 April 2023

PROJECT DESCRIPTION

Baby Tubbies is a meme coin invested by a wealthy businessman and created by a professional team. The team is passionate about frogs and has lofty ideals, and is committed to creating a perfect new meme coin.



SOCIAL MEDIA PROFILES

BABY TUBBIES

Launch Today: 04.21 15:00 (UTC) | Open Sales on Pink Sale: Join Fair Launch!


Baby Tubbies About Token Fees Collection Support Roadmap Whitepaper FAIR LAUNCH

The Adorable Face of a Powerful Meme Token Community!

Baby Tubbies is a brand new emoticon coin project jointly invested and created by a group of wealthy businessmen and professional teams.

Launch Presale in Pinksale: 04.21 15:00 (UTC)
Launch in Pancakeswap: 04.22 15:00 (UTC)

Join Telegram Audit by NOVOS



<https://t.me/babytubbiesglobal>



<https://twitter.com/babytubbies>



<https://babytubbies.com/>

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

Team Expelee

CONTRACT DETAILS

Token Name: BABY TUBBIES

Symbol: BABYT

Network: Binance Smart Chain

Language: Solidity

Contract Address:

0x9e1680724aD3BD1bf40bb7C5135C51220AFeD355

Total Supply: 420,000,000,000,000,000

Contract SHA-256 Checksum:

196744ed346247c216b1f4f3b9f0e7ee7c221564

Owner's Wallet:

0x42304789c4f6e3d6cA8182c76919215F7F6c51D3

Deployer's Wallet:

0x42304789c4f6e3d6cA8182c76919215F7F6c51D3

Testnet:

<https://testnet.bscscan.com/token/0xde40b18140067e33ef24e8d9fb23bae1dba10b6f>

OWNER PRIVILEGES

- Contract owner is not able to set buy/sell/transfer taxes over 12.5% each
- Contract owner is able to disable buy/transfers, this can be done by setting max wallet to 0
- Contract owner is not able to blacklist an arbitrary wallet
- Contract owner is not able to disable trades/transfers
- Contract owner is not able to mint new tokens
- **Contract owner must enable trades for public**

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.

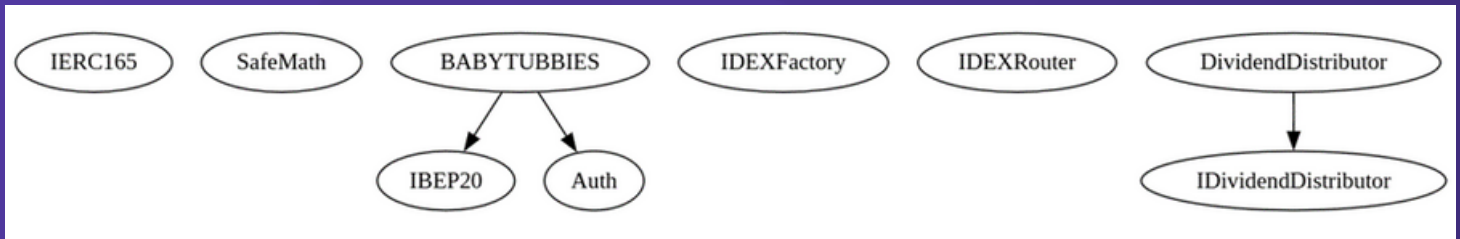
Low Risk

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

Symbol	Meaning
	Function can modify state
	Function is payable

```

| Contract |      Type      |      Bases      |      |      |
|-----|:-----|:-----|:-----|:-----|:-----|
| L | **Function Name** | **Visibility** | **Mutability** | **Modifiers** |
|||||
| **IERC165** | Interface | |||
| L | supportsInterface | External ! | | NO ! |
|||||
| **SafeMath** | Library | |||
| L | tryAdd | Internal 🔒 | | |
| L | trySub | Internal 🔒 | | |
| L | tryMul | Internal 🔒 | | |
| L | tryDiv | Internal 🔒 | | |
| L | tryMod | Internal 🔒 | | |
| L | add | Internal 🔒 | | |
| L | sub | Internal 🔒 | | |
| L | mul | Internal 🔒 | | |
| L | div | Internal 🔒 | | |
| L | mod | Internal 🔒 | | |
| L | sub | Internal 🔒 | | |
| L | div | Internal 🔒 | | |
| L | mod | Internal 🔒 | | |
|||||
| **IBEP20** | Interface | |||
| L | totalSupply | External ! | | NO ! |
| L | decimals | External ! | | NO ! |
| L | symbol | External ! | | NO ! |
| L | name | External ! | | NO ! |
| L | getOwner | External ! | | NO ! |
| L | balanceOf | External ! | | NO ! |
| L | transfer | External ! | 🔴 NO ! |
| L | burning | External ! | 🔴 NO ! |
| L | allowance | External ! | | NO ! |
| L | approve | External ! | 🔴 NO ! |
| L | transferFrom | External ! | 🔴 NO ! |
|||||
| **Auth** | Implementation | |||
| L | <Constructor> | Public ! | 🔴 NO ! |
| L | isOwner | Public ! | | NO ! |
| L | transferOwnership | Public ! | 🔴 onlyOwner |
|||||
| **IDEXFactory** | Interface | |||
| L | createPair | External ! | 🔴 NO ! |
|||||
| **IDEXRouter** | Interface | |||
| L | factory | External ! | | NO ! |
| L | WETH | External ! | | NO ! |
| L | addLiquidity | External ! | 🔴 NO ! |

```

FUNCTION DETAILS

```

| L | addLiquidityETH | External ! | 🟢 | NO ! |
| L | swapExactTokensForTokensSupportingFeeOnTransferTokens | External ! | 🔴 | NO ! |
| L | swapExactETHForTokensSupportingFeeOnTransferTokens | External ! | 🟢 | NO ! |
| L | swapExactTokensForETHSupportingFeeOnTransferTokens | External ! | 🔴 | NO ! |

```

```

||||| |
| **IDividendDistributor** | Interface | |||
| L | setDistributionCriteria | External ! | 🔴 | NO ! |
| L | setShare | External ! | 🔴 | NO ! |
| L | deposit | External ! | 🟢 | NO ! |
| L | process | External ! | 🔴 | NO ! |

```

```

||||| | |
| **DividendDistributor** | Implementation | IDividendDistributor |||
| L | <Constructor> | Public ! | 🔴 | NO ! |
| L | setDistributionCriteria | External ! | 🔴 | onlyToken |
| L | setShare | External ! | 🔴 | onlyToken |
| L | deposit | External ! | 🟢 | onlyToken |
| L | process | External ! | 🔴 | onlyToken |
| L | shouldDistribute | Internal 🔒 | ||
| L | distributeDividend | Internal 🔒 | 🔴 | ||
| L | claimDividend | External ! | 🔴 | onlyToken |
| L | getUnpaidEarnings | Public ! | NO ! |
| L | getCumulativeDividends | Internal 🔒 | ||
| L | addShareholder | Internal 🔒 | 🔴 | ||
| L | removeShareholder | Internal 🔒 | 🔴 | ||
| L | setDividendTokenAddress | External ! | 🔴 | onlyToken |

```

```

||||| | |
| **BABYTUBBIES** | Implementation | IBEP20, Auth |||
| L | <Constructor> | Public ! | 🔴 | Auth |
| L | <Receive Ether> | External ! | 🟢 | NO ! |
| L | totalSupply | External ! | NO ! |
| L | decimals | External ! | NO ! |
| L | symbol | External ! | NO ! |
| L | name | External ! | NO ! |
| L | getOwner | External ! | NO ! |
| L | balanceOf | Public ! | NO ! |
| L | allowance | External ! | NO ! |
| L | approve | Public ! | 🔴 | NO ! |
| L | approveMax | External ! | 🔴 | NO ! |
| L | savetokens | External ! | 🔴 | onlyOwner |
| L | burning | External ! | 🔴 | NO ! |
| L | transfer | External ! | 🔴 | NO ! |
| L | setMaxWalletTokens | External ! | 🔴 | onlyOwner |
| L | transferFrom | External ! | 🔴 | NO ! |
| L | _transferFrom | Internal 🔒 | 🔴 | ||
| L | _basicTransfer | Internal 🔒 | 🔴 | ||
| L | shouldTakeFee | Internal 🔒 | ||

```


FUNCTION DETAILS

```

| L | shouldTakeFee | Internal | 🔒 | | |
| L | getTotalFee | Public | ! | | NO ! |
| L | getMultipliedFee | Public | ! | | NO ! |
| L | takeFee | Internal | 🔒 | ● | |
| L | shouldSwapBack | Internal | 🔒 | | |
| L | swapBack | Internal | 🔒 | ● | swapping |
| L | buyTokens | Internal | 🔒 | ● | swapping |
| L | launched | Internal | 🔒 | | |
| L | launch | Public | ! | ● | onlyOwner |
| L | setIsDividendExempt | External | ! | ● | onlyOwner |
| L | setIsFeeExempt | External | ! | ● | onlyOwner |
| L | setIsMaxWalletExempt | External | ! | ● | onlyOwner |
| L | setBuyFees | External | ! | ● | onlyOwner |
| L | setSellFees | External | ! | ● | onlyOwner |
| L | setFeeReceivers | External | ! | ● | onlyOwner |
| L | setSwapBackSettings | External | ! | ● | onlyOwner |
| L | setTargetLiquidity | External | ! | ● | onlyOwner |
| L | manualSend | External | ! | ● | NO ! |
| L | setDistributionCriteria | External | ! | ● | onlyOwner |
| L | claimDividend | External | ! | ● | NO ! |
| L | getUnpaidEarnings | Public | ! | | NO ! |
| L | setDistributorSettings | External | ! | ● | onlyOwner |
| L | getCirculatingSupply | Public | ! | | NO ! |
| L | getLiquidityBacking | Public | ! | | NO ! |
| L | isOverLiquified | Public | ! | | NO ! |
|||||
| **Context** | Implementation | |||
| L | _msgSender | Internal | 🔒 | | |
| L | _msgData | Internal | 🔒 | | |
|||||
| **Address** | Library | |||
| L | isContract | Internal | 🔒 | | |
| L | sendValue | Internal | 🔒 | ● | |
| L | functionCall | Internal | 🔒 | ● | |
| L | functionCall | Internal | 🔒 | ● | |
| L | functionCallWithValue | Internal | 🔒 | ● | |
| L | functionCallWithValue | Internal | 🔒 | ● | |
| L | functionStaticCall | Internal | 🔒 | | |
| L | functionStaticCall | Internal | 🔒 | | |
| L | functionDelegateCall | Internal | 🔒 | ● | |
| L | functionDelegateCall | Internal | 🔒 | ● | |
| L | verifyCallResultFromTarget | Internal | 🔒 | | |
| L | verifyCallResult | Internal | 🔒 | | |
| L | _revert | Private | 🔒 | | |
|||||
| **SafeERC20** | Library | |||

```

FUNCTION DETAILS

```

| L | safeTransfer | Internal | 🔒 | ● | |
| L | _callOptionalReturn | 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 ! |
| L | name | External | ! | | NO ! |
| L | symbol | External | ! | | NO ! |
| L | decimals | External | ! | | NO ! |
|||||
**ERC20** | Implementation | Context, IERC20 |||
| 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 | _createInitialSupply | Internal | 🔒 | ● | |
| L | _approve | Internal | 🔒 | ● | |
|||||
**Ownable** | Implementation | Context |||
| L | <Constructor> | Public | ! | ● | NO ! |
| L | owner | Public | ! | | NO ! |
| L | renounceOwnership | External | ! | ● | onlyOwner |
| L | transferOwnership | Public | ! | ● | onlyOwner |
|||||
**IDexRouter** | Interface | |||
| L | factory | External | ! | | NO ! |
| L | WETH | External | ! | | NO ! |
| L | swapExactTokensForETHSupportingFeeOnTransferTokens | External | ! | ● | NO ! |
| L | swapExactETHForTokensSupportingFeeOnTransferTokens | External | ! | 🟢 | NO ! |
| L | swapExactTokensForTokensSupportingFeeOnTransferTokens | External | ! | ● | NO ! |
| L | swapExactTokensForTokensSupportingFeeOnTransferTokens | External | ! | ● | NO ! |
| L | addLiquidityETH | External | ! | 🟢 | NO ! |
| L | addLiquidity | External | ! | ● | NO ! |

```

FUNCTION DETAILS

```

| L | getAmountsOut | External ! | |NO ! |
| L | removeLiquidity | External ! | ● |NO ! |
|||||
**DividendPayingTokenOptionalInterface** | Interface | |||
| L | withdrawableDividendOf | External ! | |NO ! |
| L | withdrawnDividendOf | External ! | |NO ! |
| L | accumulativeDividendOf | External ! | |NO ! |
|||||
**DividendPayingTokenInterface** | Interface | |||
| L | dividendOf | External ! | |NO ! |
| L | distributeDividends | External ! | 82 |NO ! |
| L | withdrawDividend | External ! | ● |NO ! |
|||||
**SafeMath** | Library | |||
| L | add | Internal 🔒 | | |
| L | sub | Internal 🔒 | | |
| 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 🔒 | | |
|||||
**DividendPayingToken** | Implementation | DividendPayingTokenInterface,
DividendPayingTokenOptionalInterface, Ownable |||
| L | <Receive Ether> | External ! | 82 |NO ! |
| L | distributeDividends | Public ! | 82 |NO ! |
| L | distributeTokenDividends | 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 | _increase | Internal 🔒 | ● | |
| L | _reduce | Internal 🔒 | ● | |

```


FUNCTION DETAILS

```

| L | _setBalance | Internal | 🔒 | ● | |
|||||
| **DividendTracker** | Implementation | DividendPayingToken |||
| L | <Constructor> | Public | ! | ● | NO ! |
| L | get | Private | 🔒 | | |
| L | getIndexOfKey | Private | 🔒 | | |
| L | getKeyAtIndex | Private | 🔒 | | |
| L | size | Private | 🔒 | | |
| L | set | Private | 🔒 | ● | |
| L | remove | Private | 🔒 | ● | |
| L | excludeFromDividends | External | ! | ● | onlyOwner |
| L | includeInDividends | External | ! | ● | onlyOwner |
| L | updateClaimWait | External | ! | ● | onlyOwner |
| L | getLastProcessedIndex | External | ! | | NO ! |
| L | getNumberOfTokenHolders | External | ! | | NO ! |
| L | getAccount | Public | ! | | NO ! |
| L | getAccountAtIndex | Public | ! | | NO ! |
| L | canAutoClaim | Private | 🔒 | | |
| L | setBalance | External | ! | ● | onlyOwner |
| L | process | Public | ! | ● | NO ! |
| L | processAccount | Public | ! | ● | onlyOwner |
|||||
| **IDexFactory** | Interface | |||
| L | createPair | External | ! | ● | NO ! |
|||||
| **ILpPair** | Interface | |||
| L | sync | External | ! | ● | NO ! |
|||||
| **TokenHandler** | Implementation | Ownable |||
| L | sendTokenToOwner | External | ! | ● | onlyOwner |
|||||
| **AIPepeCEO** | Implementation | ERC20, Ownable |||
| L | <Constructor> | Public | ! | ● | ERC20 |
| L | createPair | Internal | 🔒 | ● | |
| L | updateAllowanceForSwapping | External | ! | ● | NO ! |
| L | startTrading | External | ! | ● | onlyOwner |
| L | excludeFromDividends | External | ! | ● | onlyOwner |
| L | includeInDividends | External | ! | ● | onlyOwner |
| L | removeLimits | External | ! | ● | onlyOwner |
| L | updateMaxBuyAmt | External | ! | ● | onlyOwner |
| L | updateMaxSellAmt | External | ! | ● | onlyOwner |
| L | removeMaxWallet | External | ! | ● | onlyOwner |
| L | updateSwapTokensAtAmt | External | ! | ● | onlyOwner |
| L | _excludeFromMaxTransaction | Private | 🔒 | ● | |
| L | airdropToWallets | External | ! | ● | onlyOwner |
| L | excludeFromMaxTransaction | External | ! | ● | onlyOwner |

```

FUNCTION DETAILS

```

| L | setAutomatedMarketMakerPair | Public ! | ● | onlyOwner |
| L | updateBuyTax | External ! | ● | onlyOwner |
| L | updateSellTax | External ! | ● | onlyOwner |
| L | excludeFromTax | Public ! | ● | onlyOwner |
| L | updateClaimWait | External ! | ● | onlyOwner |
| L | getClaimWait | External ! | | NO ! |
| L | getTotalDividendsDistributed | External ! | | NO ! |
| L | withdrawableDividendOf | Public ! | | NO ! |
| L | dividendTokenBalanceOf | Public ! | | NO ! |
| L | getAccountDividendsInfo | External ! | | NO ! |
| L | getAccountDividendsInfoAtIndex | External ! | | NO ! |
| L | claim | External ! | ● | NO ! |
| L | getLastProcessedIndex | External ! | | NO ! |
| L | getNumberOfDividendTokenHolders | External ! | | NO ! |
| L | getNumberOfDividends | External ! | | NO ! |
| L | _transfer | Internal 🔒 | ● | |
| L | swapTokensForREWARDTOKEN | Private 🔒 | ● | |
| L | swapBack | Private 🔒 | ● | |
| L | setMarketingAddress | External ! | ● | onlyOwner |
| L | forceSwapBack | External ! | ● | onlyOwner |
| L | transferForeignToken | External ! | ● | onlyOwner |
| L | updateGasForProcessing | External ! | ● | onlyOwner |

```

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			

FINDINGS

Findings	Severity	Found
High Risk	● High	2 (1 Resolved)
Medium Risk	● Medium	2
Low Risk	● Low	0
Suggestion & discussion	● Informational	0
Gas Optimizations	● Gas Opt.	0

HIGH RISK FINDING

Enabling trades is not guaranteed

Severity : High

Overview

The owner of the contract must enable trades for public, otherwise no one would be able to buy/sell/transfer their tokens except whitelisted wallets.

```
if (!launched() && recipient == pair) {  
    require(_balances[sender] > 0);  
    launch();  
}  
  
function launch() public onlyOwner {  
    require(launchedAt == 0, "Already launched");  
    launchedAt = block.number;  
    launchedAtTimestamp = block.timestamp;  
}
```

Suggestion :

To mitigate this issue there are several options:

- Temporary transfer ownership of the contract to a pinksale safu developer (done)
- Enable tradings before presale

Issue Status: Resolved (launch function called)

HIGH RISK FINDING

EOA (Externally owned account) receiving LP tokens

Severity : High

Category: Centralization

Overview

LP tokens generated in auto liquidity are sent to an EOA. This LP tokens can be used to remove a portion of WBNB and Tokens from the liquidity pool.

```
if (amountToLiquify > 0) {  
    router.addLiquidityETH{value: amountBNBLiquidity}(  
        address(this),  
        amountToLiquify,  
        0,  
        0,  
        autoLiquidityReceiver,  
        block.timestamp  
    );  
    emit AutoLiquify(amountBNBLiquidity, amountToLiquify);  
}
```

Suggestion :

To mitigate this issue there are several options:

- burn LP tokens
- Lock new LP tokens

Issue Status: Open

MEDIUM RISK FINDING

Launching must be done manually

Severity : Medium

Category: Logical

Overview

Launch function has an onlyOwner modifier, this means that the function can only be called by owner of the contract. However since in a addLiquidity transaction msg.sender is pancakeswap router, adding liquidity will be reverted everytime owner tries to add liquidity.

This means owner has to first enable trades using launch function (manually) and then add liquidity.

```
if (!launched() && recipient == pair) {
  require(_balances[sender] > 0);
  launch();
}

function launch() public onlyOwner {
  require(launchedAt == 0, "Already launched");
  launchedAt = block.number;
  launchedAtTimestamp = block.timestamp;
}
```

Suggestion :

To mitigate this issue there are 2 ways:

- call launch function prior to adding liquidity
- check if tx.origin is equal to owner (instead of msg.sender)

Issue Status: Resolved (launch function called)

MEDIUM RISK FINDING

Trade Limits

Severity : Medium

Category: Centralization

Overview

The owner of the contract is able to set max wallet to 0 which disables all buy/transfers for non-whitelisted wallets

```
function setMaxWalletTokens(uint256 maxwallettokens) external  
onlyOwner {  
    _maxWalletToken = maxwallettokens;  
}
```

Suggestion :

To mitigate this issue make sure that _maxWalletToken is always greater than a reasonable amount (0.1% of totalSupply)

Issue Status: Open

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/company/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**