



Building the Futuristic **Blockchain Ecosystem**

SECURITY AUDIT REPORT

Twoge Inu 2.0

TOKEN OVERVIEW

Risk Findings

Severity	Found
● High	6
● Medium	1
● Low	0
● Informational	0

Centralization Risks

Owner Privileges	Description
● Can Owner Set Taxes >25% ?	Detected
● Owner needs to enable trading ?	Yes, owner needs to enable trades
● 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	
14	Testnet Version	
16	Manual Review	
17	High Risk Finding	
22	Medium Risk Finding	
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	Failed
KYC Verification	-
Audit Date	24 July 2023

CONTRACT DETAILS

Token Name: Twoge Inu 2.0

Symbol: TWOGE2

Network: Ethereum

Language: Solidity

Contract Address:

0x69Eb55a7578EBEdDCf26E12254BAbe37530444CC

Total Supply: 694,200,000,000

Owner's Wallet:

0xAF04923371F85DA69eb3d119b90A5E48CF39bB78

Deployer's Wallet:

0xAF04923371F85DA69eb3d119b90A5E48CF39bB78

Testnet.

<https://testnet.bscscan.com/token/0xd318907De00C6904540aF2499D9889537BF9C3dB>

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.

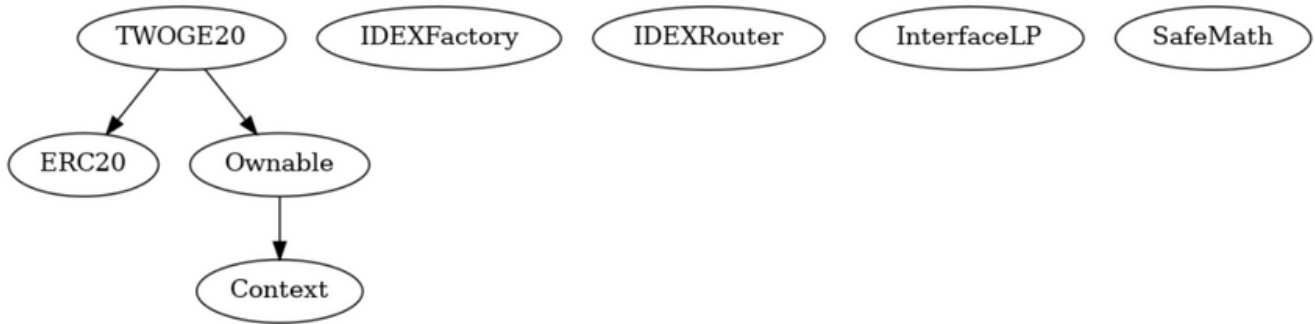
Low Risk

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**	
ERC20 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	allowance	External	!	NO !	
L	approve	External	!	NO !	
L	transferFrom	External	!	NO !	
Context Implementation					
L	_msgSender	Internal	🔒		
L	_msgData	Internal	🔒		
IDEXFactory Interface					
L	createPair	External	!	NO !	
IDEXRouter Interface					
L	factory	External	!	NO !	
L	WETH	External	!	NO !	
L	addLiquidity	External	!	NO !	
L	addLiquidityETH	External	!	NO !	
L	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	!	NO !	
L	swapExactETHForTokensSupportingFeeOnTransferTokens	External	!	NO !	
L	swapExactTokensForETHSupportingFeeOnTransferTokens	External	!	NO !	
InterfaceLP Interface					
L	sync	External	!	NO !	
Ownable Implementation Context					
L	<Constructor>	Public	!	NO !	
L	owner	Public	!	NO !	
L	renounceOwnership	Public	!	onlyOwner	

FUNCTION DETAILS

```

|  ↳ | transferOwnership | Public  !  |  🚫  | onlyOwner |
|||||
|  **SafeMath** | Library | |||
|  ↳ | add | Internal  🔒  |  |  |
|  ↳ | sub | Internal  🔒  |  |  |
|  ↳ | sub | Internal  🔒  |  |  |
|  ↳ | mul | Internal  🔒  |  |  |
|  ↳ | div | Internal  🔒  |  |  |
|  ↳ | div | Internal  🔒  |  |  |
|||||
|  **TWOGE20** | Implementation | Ownable, ERC20 |||
|  ↳ | <Constructor> | Public  !  |  🚫  | NO  !  |
|  ↳ | <Receive Ether> | External  !  |  🚫  | NO  !  |
|  ↳ | totalSupply | External  !  |  | NO  !  |
|  ↳ | decimals | External  !  |  | NO  !  |
|  ↳ | symbol | External  !  |  | NO  !  |
|  ↳ | name | External  !  |  | NO  !  |
|  ↳ | getOwner | External  !  |  | NO  !  |
|  ↳ | balanceOf | Public  !  |  | NO  !  |
|  ↳ | allowance | External  !  |  | NO  !  |
|  ↳ | approve | Public  !  |  🚫  | NO  !  |
|  ↳ | approveMax | External  !  |  🚫  | NO  !  |
|  ↳ | transfer | External  !  |  🚫  | NO  !  |
|  ↳ | transferFrom | External  !  |  🚫  | NO  !  |
|  ↳ | maxWalletRule | External  !  |  🚫  | onlyOwner |
|  ↳ | removeLimits | External  !  |  🚫  | onlyOwner |
|  ↳ | _transferFrom | Internal  🔒  |  🚫  |  |
|  ↳ | _basicTransfer | Internal  🔒  |  🚫  |  |
|  ↳ | checkTxLimit | Internal  🔒  |  |  |
|  ↳ | shouldTakeFee | Internal  🔒  |  |  |
|  ↳ | takeFee | Internal  🔒  |  🚫  |  |
|  ↳ | shouldSwapBack | Internal  🔒  |  |  |
|  ↳ | manualSend | External  !  |  🚫  | NO  !  |
|  ↳ | clearStuckToken | External  !  |  🚫  | NO  !  |
|  ↳ | setStructure | External  !  |  🚫  | onlyOwner |
|  ↳ | startTrading | Public  !  |  🚫  | onlyOwner |

```

FUNCTION DETAILS

```

| L | transferOwnership | Public ! | 🚫 | onlyOwner |
|||||
| **SafeMath** | Library | |||
| L | add | Internal 🗝️ | | |
| L | sub | Internal 🗝️ | | |
| L | sub | Internal 🗝️ | | |
| L | mul | Internal 🗝️ | | |
| L | div | Internal 🗝️ | | |
| L | div | Internal 🗝️ | | |
|||||
| **TWOGE20** | Implementation | Ownable, ERC20 |||
| L | <Constructor> | Public ! | 🚫 | NO ! |
| 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 | transfer | External ! | 🚫 | NO ! |
| L | transferFrom | External ! | 🚫 | NO ! |
| L | maxWalletRule | External ! | 🚫 | onlyOwner |
| L | removeLimits | External ! | 🚫 | onlyOwner |
| L | _transferFrom | Internal 🗝️ | 🚫 | |
| L | _basicTransfer | Internal 🗝️ | 🚫 | |
| L | checkTxLimit | Internal 🗝️ | | |
| L | shouldTakeFee | Internal 🗝️ | | |
| L | takeFee | Internal 🗝️ | 🚫 | |
| L | shouldSwapBack | Internal 🗝️ | | |
| L | manualSend | External ! | 🚫 | NO ! |
| L | clearStuckToken | External ! | 🚫 | NO ! |
| L | setStructure | External ! | 🚫 | onlyOwner |
| L | startTrading | Public ! | 🚫 | onlyOwner |

```

FUNCTION DETAILS

```

|  L | reduceFee | Public  !  |  🔴  | onlyOwner | |
|  L | swapBack  | Internal 🔒  |  🔴  | swapping |
|  L | set_fees  | Internal 🔒  |  🔴  | ||
|  L | setParameters | External  !  |  🔴  | onlyOwner |
|  L | setWallets | External  !  |  🔴  | onlyOwner |
|  L | setSwapBackSettings | External  !  |  🔴  | onlyOwner |
|  L | checkRatio | Public  !  | |NO  !  |
|  L | showBacking | Public  !  | |NO  !  |
|  L | showSupply | Public  !  | |NO  !  |

```

Legend

```

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

```

TESTNET VERSION

Adding Liquidity ✓

Tx:

<https://testnet.bscscan.com/tx/0x46d42befa30cdc9d734ba7681f8c325c00f4adf10e6a28acd30a67157592f9ca>

=====

Buying when excluded from fees ✓

Tx (0% tax):

<https://testnet.bscscan.com/tx/0x3348991f45a8cbd3e2cc32fc3d2dbb8de31664cc74ae1d559b254fae16cfcd85>

=====

Selling when excluded from fees ✓

Tx (0% tax):

<https://testnet.bscscan.com/tx/0x382ba1907cbb1a1600fda56972a4e6294764a8532a5bc0d70fbbaee5cdb19252>

=====

Transferring when excluded from fees ✓

Tx (0% tax):

<https://testnet.bscscan.com/tx/0xef2ed534c54bc8cf5adcfb7e920d4c1b9e987feecbd6c3ccdc0370a22689cbbe>

=====

Buying ✓

Tx (0-100% tax):

<https://testnet.bscscan.com/tx/0x8320e681cd18b9d312ae6066e4d77360d31fec1aa94057dd32c3ccdcfd855517>

TESTNET VERSION

Selling ✓

Tx (0-100% tax):

<https://testnet.bscscan.com/tx/0xb79f9673754b79a98944434af68285e9104227a7b75f29ed8ac29dc2519e2ea9>

=====

Transferring ✓

Tx (0-100% tax):

<https://testnet.bscscan.com/tx/0xc97f45fcb8dc98a7d94f6c9b17177f8f87d455fa86317261402dcc493321dd0f>

=====

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

Tx:

<https://testnet.bscscan.com/tx/0x5d997a5ba8c3852bb7f16aec89811fee2d7874a3f6baf08a2885d4cc37f6135d>

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

Trades are disabled by default

Category: **Centralization**

Status: **Open**

Impact: **High**

Overview:

The contract has been structured such that all trading is disabled by default, necessitating the contract owner's manual intervention to enable trading. This can lead to a situation where, if trades remain disabled, token holders won't be able to buy, sell, or trade their tokens, causing a severe impact on the token's usability and market liquidity.

```
function startTrading() public onlyOwner {  
    TradingOpen = true;  
    buypercent = 1000;  
    sellpercent = 800;  
    transferpercent = 1000;  
}
```

Suggestion:

To mitigate this risk, it is recommended that trading be enabled before the token presale. This can be achieved by invoking the "startTrading" function or by transferring ownership of the contract to a third-party that has established trust with the community, such as a Certified SAFU developer. This reduces the concentration of power and the potential for malicious actions, thereby promoting a more decentralized and fair environment for all participants.

HIGH RISK FINDING

Excessive fees

Category: Centralization

Status: Open

Impact: High

Overview:

The contract owner is able to set up to 100% fee for buy/sell/transfers.

```
function setStructure(uint256 _percentonbuy, uint256 _percentonsell,
uint256 _wallettransfer) external onlyOwner {
    sellpercent = _percentonsell;
    buypercent = _percentonbuy;
    transferpercent = _wallettransfer;
}
```

Suggestion:

Ensure that fees are within a reasonable range, usually 0-10% is suggested as a reasonable tax for investors.

HIGH RISK FINDING

No function for authorizing wallets

Category: Logical

Status: Open

Impact: High

Overview:

Contract allows authorized wallets to be able to transfer tokens when trades are disabled for public but there are not functions in order to authorize new wallets.

This means owner will not be able to authorize certain wallets like presale wallet

```
if (!authorizations[sender] && !authorizations[recipient]) {  
    require(TradingOpen, "Trading not open yet");  
}
```

Suggestion:

Create an onlyOwner function to be able to authorize a new wallet

HIGH RISK FINDING

Underflow error at takeFee

Category: Logical

Status: Open

Impact: High

Overview:

at takeFee function, total fee is calculated in this way:

```
uint256 feeAmount = amount.mul(totalFee).mul(percent).div(feeDenominator * 100);
```

the transaction reverts here if numerator is greater than denominator:

```
amount.mul(totalFee).mul(percent) > feeDenominator * 100
```

Suggestion:

Ensure that numerator is not greater than denominator, this requires adding proper validation at setParameters and setStructure functions.

HIGH RISK FINDING

Setting swapthreshold to 0 can disable sells

Category: Logical

Status: Open

Impact: High

Overview:

“swapThreshold” is the amount of tokens contract uses for performing internal swap. If this amount is set to zero, internal swap reverts, reverting whole transaction.

```
function setSwapBackSettings(bool _enabled, uint256 _amount) external
onlyOwner {
    swapEnabled = _enabled;
    swapThreshold = _amount;
    emit set_SwapBack(swapThreshold, swapEnabled);
}
```

Suggestion:

Ensure that swapThreshold is more than 0:

```
function setSwapBackSettings(bool _enabled, uint256 _amount) external
onlyOwner {
    swapEnabled = _enabled;
    swapThreshold = _amount;
    require(swapThreshold > 10 ** decimals(), “swap threshold must be greater
than one token”);
    emit set_SwapBack(swapThreshold, swapEnabled);
}
```

HIGH RISK FINDING

Setting Fees to 0 could revert internal swap transactions

Category: **Logical**

Status: **Open**

Impact: **High**

Overview:

if totalFee is equal to zero, then a divide by zero will revert the transaction in which internal swap (swapBack) is performing

```
function swapBack() internal swapping {  
    uint256 dynamicLiquidityFee = checkRatio(setRatio, setRatioDenominator) ?  
    0 : liquidityFee;  
    uint256 amountToLiquify =  
    swapThreshold.mul(dynamicLiquidityFee).div(totalFee).div(2);
```

Suggestion:

Ensure that swapThreshold is more than 0:

```
function swapBack() internal swapping {  
    if(totalFee == 0) return;  
    if(swapThreshold == 0) return;  
    uint256 dynamicLiquidityFee = checkRatio(setRatio, setRatioDenominator) ?  
    0 : liquidityFee;  
    uint256 amountToLiquify =  
    swapThreshold.mul(dynamicLiquidityFee).div(totalFee).div(2);
```

MEDIUM RISK FINDING

EOA receiving LP tokens

Category: Centralization

Status: Open

Impact: Medium

Overview:

an EOA wallet (autoLiquidityReceiver) is receiving LP tokens generated from auto-liquidity.

This accumulated LP tokens could be used to remove a portion of liquidity pool.

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

Suggestion:

Its suggested to either 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.

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**