expelee

Building the Futuristic Blockchain Ecosystem

SECURITY AUDIT REPORT

Donut Doughnut Trump

TOKEN OVERVIEW

Risk Findings

Severity	Found
High	0
Medium	1
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

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	**11 Aug 2023**

CONTRACT DETAILS

Token Name: Donut Doughnut Trump

Symbol: DDT

Network: BSC

Language: Solidity

Contract Address:
0x2af1e69fC340500292d7977bF3788095e1589500

Total Supply: 1,000,000

Owner's Wallet:
0x9420089f9B87eD68E55130321e87271726B8d8Be

Deployer's Wallet:
0x9420089f9B87eD68E55130321e87271726B8d8Be

Testnet.
https://goerli.etherscan.io/token/0x6bF21e3507540e650503063175748 73A479440B3

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.

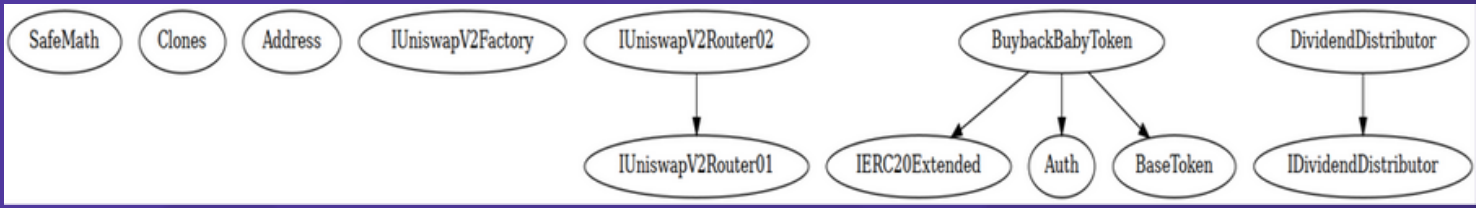
Low Risk

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.

INHERITANCE TREES



expelee

FUNCTION DETAILS

Contract	Type	Bases		
└	**Function Name**	**Visibility**	**Mutability**	**Modifiers**
SafeMath	Library			
└	tryAdd	Internal 🔒		
└	trySub	Internal 🔒		
└	tryMul	Internal 🔒		
└	tryDiv	Internal 🔒		
└	tryMod	Internal 🔒		
└	add	Internal 🔒		
└	sub	Internal 🔒		
└	mul	Internal 🔒		
└	div	Internal 🔒		
└	mod	Internal 🔒		
└	sub	Internal 🔒		
└	div	Internal 🔒		
└	mod	Internal 🔒		
Clones	Library			
└	clone	Internal 🔒	🔴	
└	cloneDeterministic	Internal 🔒	🔴	
└	predictDeterministicAddress	Internal 🔒		
└	predictDeterministicAddress	Internal 🔒		
Address	Library			
└	isContract	Internal 🔒		
└	sendValue	Internal 🔒	🔴	
└	functionCall	Internal 🔒	🔴	
└	functionCall	Internal 🔒	🔴	
└	functionCallWithValue	Internal 🔒	🔴	
└	functionCallWithValue	Internal 🔒	🔴	
└	functionStaticCall	Internal 🔒		
└	functionStaticCall	Internal 🔒		
└	functionDelegateCall	Internal 🔒	🔴	
└	functionDelegateCall	Internal 🔒	🔴	
└	verifyCallResult	Internal 🔒		
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 ❗	

expelee

FUNCTION DETAILS

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 ❗	
IERC20Extended	Interface			
└	totalSupply	External ❗		|NO ❗	
└	decimals	External ❗		|NO ❗	
└	symbol	External ❗		|NO ❗	
└	name	External ❗		|NO ❗	
└	balanceOf	External ❗		|NO ❗	
└	transfer	External ❗	🔴	NO ❗	
└	allowance	External ❗		|NO ❗	
└	approve	External ❗	🔴	NO ❗	
└	transferFrom	External ❗	🔴	NO ❗	
Auth	Implementation			
└	<Constructor>	Public ❗	🔴	NO ❗	
└	authorize	Public ❗	🔴	onlyOwner
└	unauthorize	Public ❗	🔴	onlyOwner
└	isOwner	Public ❗		|NO ❗	

expelee

FUNCTION DETAILS

└	isAuthorized	Public ❗		NO ❗
└	transferOwnership	Public ❗	🔴	onlyOwner
IDividendDistributor	Interface			
└	setDistributionCriteria	External ❗	🔴	NO ❗
└	setShare	External ❗	🔴	NO ❗
└	deposit	External ❗	💵	NO ❗
└	process	External ❗	🔴	NO ❗
DividendDistributor	Implementation	IDividendDistributor		
└	<Constructor>	Public ❗	🔴	NO ❗
└	setDistributionCriteria	External ❗	🔴	onlyToken
└	setShare	External ❗	🔴	onlyToken
└	deposit	External ❗	💵	onlyToken
└	process	External ❗	🔴	onlyToken
└	shouldDistribute	Internal 🔒		
└	distributeDividend	Internal 🔒	🔴	
└	claimDividend	External ❗	🔴	NO ❗
└	getUnpaidEarnings	Public ❗		NO ❗
└	getCumulativeDividends	Internal 🔒		
└	addShareholder	Internal 🔒	🔴	
└	removeShareholder	Internal 🔒	🔴	
BaseToken	Implementation			
BuybackBabyToken	Implementation	IERC20Extended, Auth, BaseToken		
└	<Constructor>	Public ❗	💵	Auth
└	_initializeFees	Internal 🔒	🔴	
└	_initializeLiquidityBuyBack	Internal 🔒	🔴	
└	<Receive Ether>	External ❗	💵	NO ❗
└	totalSupply	External ❗		NO ❗
└	decimals	External ❗		NO ❗
└	symbol	External ❗		NO ❗
└	name	External ❗		NO ❗
└	balanceOf	Public ❗		NO ❗
└	allowance	External ❗		NO ❗
└	approve	Public ❗	🔴	NO ❗
└	approveMax	External ❗	🔴	NO ❗
└	transfer	External ❗	🔴	NO ❗
└	transferFrom	External ❗	🔴	NO ❗
└	_transferFrom	Internal 🔒	🔴	
└	_basicTransfer	Internal 🔒	🔴	
└	shouldTakeFee	Internal 🔒		
└	getTotalFee	Public ❗		NO ❗
└	getMultipliedFee	Public ❗		NO ❗
└	takeFee	Internal 🔒	🔴	

FUNCTION DETAILS

└	shouldSwapBack	Internal 🔒		
└	swapBack	Internal 🔒	🔴	swapping
└	shouldAutoBuyback	Internal 🔒		
└	triggerZeusBuyback	External ❗	🔴	authorized
└	clearBuybackMultiplier	External ❗	🔴	authorized
└	triggerAutoBuyback	Internal 🔒	🔴	
└	buyTokens	Internal 🔒	🔴	swapping
└	setAutoBuybackSettings	External ❗	🔴	authorized
└	setBuybackMultiplierSettings	External ❗	🔴	authorized
└	setIsDividendExempt	External ❗	🔴	authorized
└	setIsFeeExempt	External ❗	🔴	authorized
└	setBuyBacker	External ❗	🔴	authorized
└	setFees	Public ❗	🔴	authorized
└	_setFees	Internal 🔒	🔴	
└	setFeeReceivers	External ❗	🔴	authorized
└	setSwapBackSettings	External ❗	🔴	authorized
└	setTargetLiquidity	External ❗	🔴	authorized
└	setDistributionCriteria	External ❗	🔴	onlyOwner
└	setDistributorSettings	External ❗	🔴	authorized
└	getCirculatingSupply	Public ❗		NO ❗
└	getLiquidityBacking	Public ❗		NO ❗
└	isOverLiquified	Public ❗		NO ❗

Legend

Symbol	Meaning
🔴	Function can modify state
💵	Function is payable

TESTNET VERSION

Adding Liquidity ✅
Tx:
https://goerli.etherscan.io/tx/0x7735dcb0859fd52a385d61eef93b688ba0cab25b9ede8ef468d487edff7ec1a3

==========

Buying when excluded from fees ✅
Tx (0-25% tax):
https://goerli.etherscan.io/tx/0x2023f1cf17e2b0648bc0a25a1826c71aa4112afb5367adc655945c2b022c4a05

==========

Selling when excluded from fees ✅
Tx (0% tax):
https://goerli.etherscan.io/tx/0x2023f1cf17e2b0648bc0a25a1826c71aa4112afb5367adc655945c2b022c4a05

==========

Transferring when excluded from fees ✅
Tx (0% tax):
https://goerli.etherscan.io/tx/0xc47d0307d1881d52235d570b7d254557df71a848027d22b42c3e62a81e8cf380

==========

Buying ✅
Tx (0-25% tax):
https://goerli.etherscan.io/tx/0xb3c76647776b6b42d0c62cedcf708e6b8ff37ba20b89288f8e9cb4aea8f03561

TESTNET VERSION

Selling ✅
Tx (0-25% tax):
https://goerli.etherscan.io/tx/0x51aae3484903640379 03ff4a38f62b3ae1e859f49db45ea158f6dc639bb0fb6b

==========

Transferring ✅
Tx (0-25% tax):
https://goerli.etherscan.io/tx/0xfb0e318ef3a7be59bbaaca2a269a8ed80559dd0a44dcf04b75405759f6f93c8c

==========

Internal swap (Reward distribution, auto-liquidity, buyback, reward distribution) ✅
Tx:
https://goerli.etherscan.io/tx/0x8a67664910b509ed2c6cbe1ee9a7e3bf140d27b7fcaed725b93afcbf02154af5

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			

MEDIUM RISK FINDING

Up to 25% fee

Status: Open
Impact: Medium

Overview:
Owner is able to update buy/sel/transfer fees within 0-25 percent. The upper bound might be considered a high amount of tax for investors.

```
function setFees(
    uint256 _liquidityFee,
    uint256 _buybackFee,
    uint256 _reflectionFee,
    uint256 _marketingFee,
    uint256 _feeDenominator
) public authorized {
    _setFees(
        _liquidityFee,
        _buybackFee,
        _reflectionFee,
        _marketingFee,
        _feeDenominator
    );
}
```

Suggestion:
Its suggested to declare a more reasonable upper bound for fees. This upper bound is suggested to be 10% according to pinksale safu cirteria

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

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.

Building the Futuristic Blockchain Ecosystem