expelee

Building the Futuristic Blockchain Ecosystem

SECURITY AUDIT REPORT

UNICORN

TOKEN OVERVIEW

Risk Findings

Severity	Found
High	1
Medium	0
Low	0
Informational	1

Centralization Risks

Owner Privileges	Description
Can Owner Set Taxes >25% ?	Not 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

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	**23 August 2023**

CONTRACT DETAILS

Token Address:
0xE134452bdE626424a3443C2e2e28EC445F759165

Name: Unicorn

Symbol: UNICORN

Decimals: 18

Network: Ethereum

Token Type: ERC20

Owner: 0x8c0093Ae2705e98D65122C18404f0F469388b05f

Deployer: 0xfe85CDe37f126a5017cD5803897CBF89292bbcf3

Token Supply: 150,000,000,000,000

Checksum:
f3799cf290a83cd0d4e6a5a112c92d2fcb63f178

Testnet version:
The tests conducted were performed on the contract deployed on the Binance Smart Chain (BSC) Testnet.
https://testnet.bscscan.com/address/0x1baA2733ce3b83B721fF69209554ce6745e8563C

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

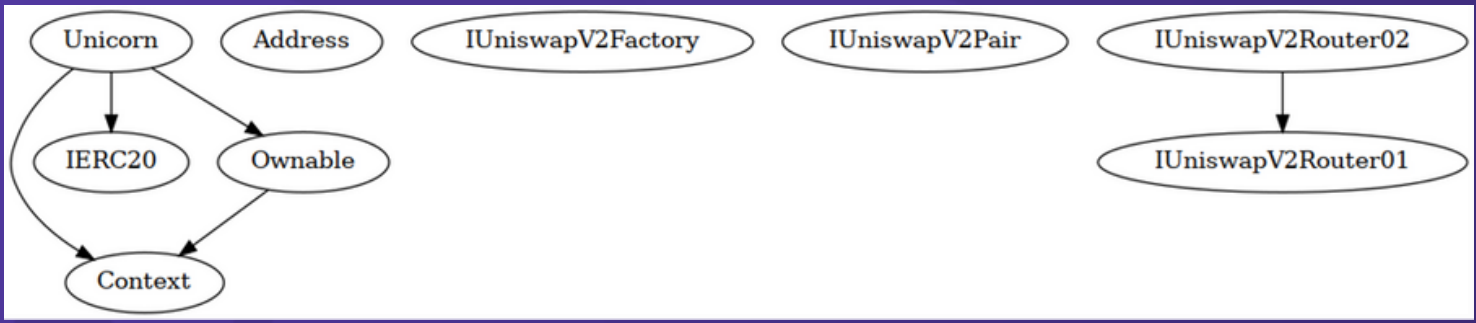
Issues on this level are minor details and warning that can remain unfixed.

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**
Context	Implementation				
└	_msgSender	Internal 🔒		
IERC20	Interface				
└	totalSupply	External ❗	|NO ❗	
└	balanceOf	External ❗	|NO ❗	
└	transfer	External ❗	🛑 |NO ❗	
└	allowance	External ❗	|NO ❗	
└	approve	External ❗	🛑 |NO ❗	
└	transferFrom	External ❗	🛑 |NO ❗	
Address	Library				
└	isContract	Internal 🔒		
└	sendValue	Internal 🔒	🛑	
└	functionCall	Internal 🔒	🛑	
└	functionCall	Internal 🔒	🛑	
└	functionCallWithValue	Internal 🔒	🛑	
└	functionCallWithValue	Internal 🔒	🛑	
└	functionStaticCall	Internal 🔒		
└	functionStaticCall	Internal 🔒		
└	functionDelegateCall	Internal 🔒	🛑	
└	functionDelegateCall	Internal 🔒	🛑	
└	verifyCallResultFromTarget	Internal 🔒		
└	verifyCallResult	Internal 🔒		
└	_revert	Private 🔐		
IUniswapV2Factory	Interface				
└	feeTo	External ❗	|NO ❗	
└	feeToSetter	External ❗	|NO ❗	
└	getPair	External ❗	|NO ❗	
└	allPairs	External ❗	|NO ❗	
└	allPairsLength	External ❗	|NO ❗	

FUNCTION DETAILS

└	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❗
└	getReserves	External ❗		NO❗
└	burn	External ❗	🛑	NO❗
└	swap	External ❗	🛑	NO❗
└	skim	External ❗	🛑	NO❗
└	sync	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❗
└	price0CumulativeLast	External ❗		NO❗
└	price1CumulativeLast	External ❗		NO❗
└	kLast	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❗

FUNCTION DETAILS

└	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 ❗
Ownable	Implementation	Context		
└	<Constructor>	Public ❗	🛑	NO ❗
└	owner	Public ❗		NO ❗
└	_checkOwner	Internal 🔒		
└	transferOwnership	Public ❗	🛑	onlyOwner
└	_transferOwnership	Internal 🔒	🛑	
Unicorn	Implementation	Context, IERC20, Ownable		
└	<Constructor>	Public ❗	🛑	NO ❗
└	renounceOwnership	Public ❗	🛑	onlyOwner
└	totalSupply	Public ❗		NO ❗
└	balanceOf	Public ❗		NO ❗
└	isExcludedFromFee	External ❗		NO ❗
└	transfer	Public ❗	🛑	NO ❗
└	allowance	Public ❗		NO ❗
└	approve	Public ❗	🛑	NO ❗
└	setMaxTxAmount	External ❗	🛑	onlyOwner
└	setMaxWalletAmount	External ❗	🛑	onlyOwner
└	transferFrom	Public ❗	🛑	NO ❗
└	increaseAllowance	Public ❗	🛑	NO ❗
└	decreaseAllowance	Public ❗	🛑	NO ❗
└	_approve	Private 🔐	🛑	
└	_transfer	Private 🔐	🛑	
└	swapAndLiquify	Public ❗	🛑	lockTheSwap
└	swapTokensForEth	Private 🔐	🛑	
└	_tokenTransfer	Private 🔐	🛑	
└	excludeFromFee	External ❗	🛑	onlyOwner
└	includeInFee	External ❗	🛑	onlyOwner
└	setTokensToSwap	External ❗	🛑	onlyOwner
└	setSwapAndLiquifyEnabled	External ❗	🛑	onlyOwner
└	setMarketingWallet	External ❗	🛑	onlyOwner

FUNCTION DETAILS

└	transferToAddressETH	Private 🔐	🛑	
└	<Receive Ether>	External ❗	💵	NO❗
└	swapETHForTokens	Private 🔐	🛑	
└	recoverETHfromContract	External ❗	🛑	onlyOwner
└	recoverTokensFromContract	External ❗	🛑	onlyOwner
└	enableTrading	External ❗	🛑	onlyOwner

Legend

Symbol	Meaning
🛑	Function can modify state
💵	Function is payable

TESTNET VERSION

Adding Liquidity ✅
Tx:
https://testnet.bscscan.com/tx/0x59c6b75e061174e1322479cde410bf40fc4927f6bb692f5ebe748efcb7bc432f

==========

Buying when excluded from fees ✅
Tx (0% tax):
https://testnet.bscscan.com/tx/0x57184542031a38c5b793a626caf80ae434423d89061f53cdba32a75959418f1b

==========

Selling when excluded from fees ✅
Tx (0% tax):
https://testnet.bscscan.com/tx/0x9ce22ac60ec0efe855f7df65b13894e1e67c068b9842c2d8129ff908727bc2c3

==========

Transferring when excluded from fees ✅
Tx (0% tax):
https://testnet.bscscan.com/tx/0x61ab8836bd0637b4ad783aa500621e0f676392f8c131e1e3263d84ae6fbb05bf

==========

Buying ✅
Tx (1% tax):
https://testnet.bscscan.com/tx/0x482fb49a7bf6c28eb4693d75c661cc1d750391477b877a0c66347ca790c5fb51

expelee

TESTNET VERSION

Selling ✅
Tx (1% tax):
https://testnet.bscscan.com/tx/0xeca794a937f736fa6c6923ccdd037f41eed0eb836e1b84b6cea61cf175351795

==========

Transferring ✅
Tx (0% tax):
https://testnet.bscscan.com/tx/0xf63d2478515bbabadd46c6f4ae8125e0a45e8317cd420d50ed2e17ec0567fde6

==========

Internal swap (ETH sent to marketing wallet) ✅
Tx:
https://testnet.bscscan.com/address/0x121259519b182bd4230611499a8d05c82aeb6f4f#internaltx

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

Category: Centralization

Subject: Trades are disabled by default

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 enableTrading() external onlyOwner {
require(!tradingOpen, "Trading already enabled.");
      tradingOpen = true;
swapAndLiquifyEnabled = true;
      emit AuditLog(
        "We have Enable Trading and Automatic Swaps:",
        msg.sender
      );
  }
```

Suggestion:
To mitigate this risk, it is recommended that trading be enabled before the token presale. This can be achieved by invoking the "enableTrading" 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.

INFORMATIONAL FINDING

Category: Logical

Subject: Transferring ETH to a contract that rejects ETH

Status: Open

Impact: Informational

Overview:
transferToAddressETH function transfers “amount” of ETH to “recipient” if recipient (or marketing wallet) is a contract that rejects receiving ether, swapAndLiquify could fail due to this issue. (which will revert the whole transfer / sell transactions (in which swap and liquify is performed) for all wallets)

```
function transferToAddressETH(
    address payable recipient,
    uint256 amount
) private {
    (bool succ, ) = recipient.call{value: amount}("");
    require(succ, "Transfer failed.");
}
```

Suggestion:
To mitigate this risk, ignore return value (“success”) of the low-level call

```
function transferToAddressETH(
    address payable recipient,
    uint256 amount
) private {
    (bool succ, ) = recipient.call{value: amount}("");
}
```

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.

Building the Futuristic Blockchain Ecosystem