expelee

Building the Futuristic Blockchain Ecosystem

SECURITY AUDIT REPORT

VaultChain

TOKEN OVERVIEW

Risk Findings

Severity	Found
High	1
Medium	0
Low	0
Informational	0

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	**14 October 2023**

expelee

CONTRACT DETAILS

Token Address: 0x0ab96A5F084228e2EE3F4970C4EBA350e8548AA9

Name: VaultChain

Symbol: VaultChain

Decimals: 18

Network: BSC

Token Type: BEP20

Owner: 0x6e116080d40DbcBa47b8230AC1682395f5d130db

Deployer: 0x6e116080d40DbcBa47b8230AC1682395f5d130db

Token Supply: 1,000,000,000

Checksum:
cb2134035a08d9a9f0030b2f1bc77b3adcf0973d

Testnet version:
The tests conducted were performed on the contract deployed on the Binance Smart Chain (BSC) Testnet.
https://testnet.bscscan.com/token/0x2233FFCFc3CBBe5979c2dC4955E66B509Ac882F9

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

FUNCTION DETAILS

Contract	Type	Bases		
└	**Function Name**	**Visibility**	**Mutability**	**Modifiers**
SafeMathInt	Library			
└	mul	Internal 🔒		
└	div	Internal 🔒		
└	sub	Internal 🔒		
└	add	Internal 🔒		
└	abs	Internal 🔒		
└	toUint256Safe	Internal 🔒		
SafeMathUint	Library			
└	toInt256Safe	Internal 🔒		
IterableMapping	Library			
└	get	Internal 🔒		
└	getIndexOfKey	Internal 🔒		
└	getKeyAtIndex	Internal 🔒		
└	size	Internal 🔒		
└	set	Internal 🔒	🛑	
└	remove	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 ❗
IUniswapV2Pair	Interface			
└	name	External ❗		NO ❗
└	symbol	External ❗		NO ❗
└	decimals	External ❗		NO ❗
└	totalSupply	External ❗		NO ❗
└	balanceOf	External ❗		NO ❗
└	allowance	External ❗		NO ❗

expelee

FUNCTION DETAILS

└	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 ❗
└	mint	External ❗	🛑	NO ❗
└	burn	External ❗	🛑	NO ❗
└	swap	External ❗	🛑	NO ❗
└	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 ❗

FUNCTION DETAILS

└	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External ❗	🛑	NO ❗
└	swapExactTokensForTokensSupportingFeeOnTransferTokens	External ❗	🛑	NO ❗
└	swapExactETHForTokensSupportingFeeOnTransferTokens	External ❗	💵	NO ❗
└	swapExactTokensForETHSupportingFeeOnTransferTokens	External ❗	🛑	NO ❗
DividendPayingTokenInterface	Interface	|||		
└	dividendOf	External ❗	|NO ❗	
└	withdrawDividend	External ❗	🛑 |NO ❗	
DividendPayingTokenOptionalInterface	Interface	|||		
└	withdrawableDividendOf	External ❗	|NO ❗	
└	withdrawnDividendOf	External ❗	|NO ❗	
└	accumulativeDividendOf	External ❗	|NO ❗	
DividendPayingToken	Implementation	ERC20, Ownable, DividendPayingTokenInterface, DividendPayingTokenOptionalInterface |||		
└	<Constructor>	Public ❗	🛑	ERC20	
└	distributeDividends	External ❗	🛑	onlyOwner	
└	withdrawDividend	External ❗	🛑 |NO ❗	
└	_withdrawDividendOfUser	Internal 🔒	🛑		
└	dividendOf	Public ❗	|NO ❗	
└	withdrawableDividendOf	Public ❗	|NO ❗	
└	withdrawnDividendOf	Public ❗	|NO ❗	
└	accumulativeDividendOf	Public ❗	|NO ❗	
└	_transfer	Internal 🔒	🛑		
└	_mint	Internal 🔒	🛑		
└	_burn	Internal 🔒	🛑		
└	_setBalance	Internal 🔒	🛑		
DividendTracker	Implementation	Ownable, DividendPayingToken |||		
└	<Constructor>	Public ❗	🛑	DividendPayingToken	
└	_transfer	Internal 🔒			
└	withdrawDividend	External ❗	|NO ❗	
└	updateMinimumTokenBalanceForDividends	External ❗	🛑	onlyOwner	
└	excludeFromDividends	External ❗	🛑	onlyOwner	
└	updateClaimWait	External ❗	🛑	onlyOwner	
└	setLastProcessedIndex	External ❗	🛑	onlyOwner	
└	getLastProcessedIndex	External ❗	|NO ❗	
└	getNumberOfTokenHolders	External ❗	|NO ❗	
└	getAccount	Public ❗	|NO ❗	
└	getAccountAtIndex	External ❗	|NO ❗	
└	canAutoClaim	Private 🔐			
└	setBalance	External ❗	🛑	onlyOwner	
└	process	External ❗	🛑 |NO ❗	
└	processAccount	Public ❗	🛑	onlyOwner	

FUNCTION DETAILS

VaultChain	Implementation	ERC20, Ownable		
└	<Constructor>	Public ❗	💵	ERC20
└	<Receive Ether>	External ❗	💵	NO❗
└	claimStuckTokens	External ❗	🛑	onlyOwner
└	isContract	Internal 🔒		
└	_setAutomatedMarketMakerPair	Private 🔐	🛑	
└	setWhitelistStatus	Public ❗	🛑	onlyOwner
└	iswhitelisted	External ❗		NO❗
└	updateBuyFees	Public ❗	🛑	onlyOwner
└	updateSellFees	Public ❗	🛑	onlyOwner
└	changeMarketingWallet	External ❗	🛑	onlyOwner
└	changeTreasuryWallet	External ❗	🛑	onlyOwner
└	changeDevelopmentWallet	External ❗	🛑	onlyOwner
└	setMaxWalletPercentage	Public ❗	🛑	onlyOwner
└	enableTrading	External ❗	🛑	onlyOwner
└	_transfer	Internal 🔒	🛑	
└	swapAndLiquify	Private 🔐	🛑	
└	Burn	Private 🔐	🛑	
└	swapAndSendDividends	Private 🔐	🛑	
└	setSwapTokensAtAmount	External ❗	🛑	onlyOwner
└	updateGasForProcessing	External ❗	🛑	onlyOwner
└	updateMinimumBalanceForDividends	External ❗	🛑	onlyOwner
└	updateClaimWait	Public ❗	🛑	onlyOwner
└	getClaimWait	External ❗		NO❗
└	getTotalDividendsDistributed	External ❗		NO❗
└	withdrawableDividendOf	External ❗		NO❗
└	dividendTokenBalanceOf	External ❗		NO❗
└	totalRewardsEarned	External ❗		NO❗
└	excludeFromDividends	External ❗	🛑	onlyOwner
└	getAccountDividendsInfo	External ❗		NO❗
└	getAccountDividendsInfoAtIndex	External ❗		NO❗
└	processDividendTracker	External ❗	🛑	NO❗
└	claim	External ❗	🛑	NO❗
└	claimAddress	External ❗	🛑	onlyOwner
└	getLastProcessedIndex	External ❗		NO❗
└	setLastProcessedIndex	External ❗	🛑	onlyOwner
└	getNumberOfDividendTokenHolders	External ❗		NO❗

Legend

Symbol	Meaning
🛑	Function can modify state
💵	Function is payable

TESTNET VERSION

Adding Liquidity ✅
Tx:
https://testnet.bscscan.com/tx/0x6f8d02d3142f3465108fd89d5e97b280a0e7b8e85bee0fb5fe50646c3073851a

==========

Buying when excluded from fees ✅
Tx (0% tax):
https://testnet.bscscan.com/tx/0xd627fa781cd3a917b458affbc695bce1d3dc9d6beb3e883fe6ad0bdb0e9b8e94

==========

Selling when excluded from fees ✅
Tx (0% tax):
https://testnet.bscscan.com/tx/0x54e487920d6d2bcf1de4bfadb711a7b99d1057409636f5359e619bf81a1784a9

==========

Transferring when excluded from fees ✅
Tx (0% tax):
https://testnet.bscscan.com/tx/0xc6bbbfba86fd216e18fb82c7af36cea656d7d11390ccec0731bb91866f4051ab

==========

Buying ✅
Tx (0-10% tax):
https://testnet.bscscan.com/tx/0x3f3753127af00510f9a39fe918fa3062920d179166f2402bae5e3f60a7ef9a8a

TESTNET VERSION

Selling ✅
Tx (0-10% tax):
https://testnet.bscscan.com/tx/0x9f53f58d9a1e2a10d3143c72f6031c9ee7e7a924ad3216f986b8bf59af9ebabd

==========

Transferring ✅
Tx (0-10% tax):
https://testnet.bscscan.com/tx/0x7ba3d1d5f6910ad423461dde3a6529d17f7bfceced8dfc8955ee30d951d16cc2

==========

Internal swap ✅
Tx:
https://testnet.bscscan.com/tx/0x9f53f58d9a1e2a10d3143c72f6031c9ee7e7a924ad3216f986b8bf59af9ebabd

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(!tradingEnabled, "Trading is already enabled");
      tradingEnabled = true;
      startTradingBlock = block.number;
  }
```

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.

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.