expelee

Building the Futuristic Blockchain Ecosystem

SECURITY AUDIT REPORT

BENDER INU

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	**18 July 2023**

CONTRACT DETAILS

Token Name: BENDER INU

Symbol: BENDERINU

Network: Binance Smart Chain

Language: Solidity

Contract Address:
0xC0064020C72B5e7341d8e4b6f84df7FB3bd460aB

Total Supply: 420,699,000,000,000

Owner's Wallet:
0xB1E24b35f125d3304B4e0D9ED428ddc82B072514

Deployer's Wallet:
0xB1E24b35f125d3304B4e0D9ED428ddc82B072514

Testnet.
https://testnet.bscscan.com/token/0xc12bF0e13d5c20F218fEDe9A9ac9034c1B1Ac6EA

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.

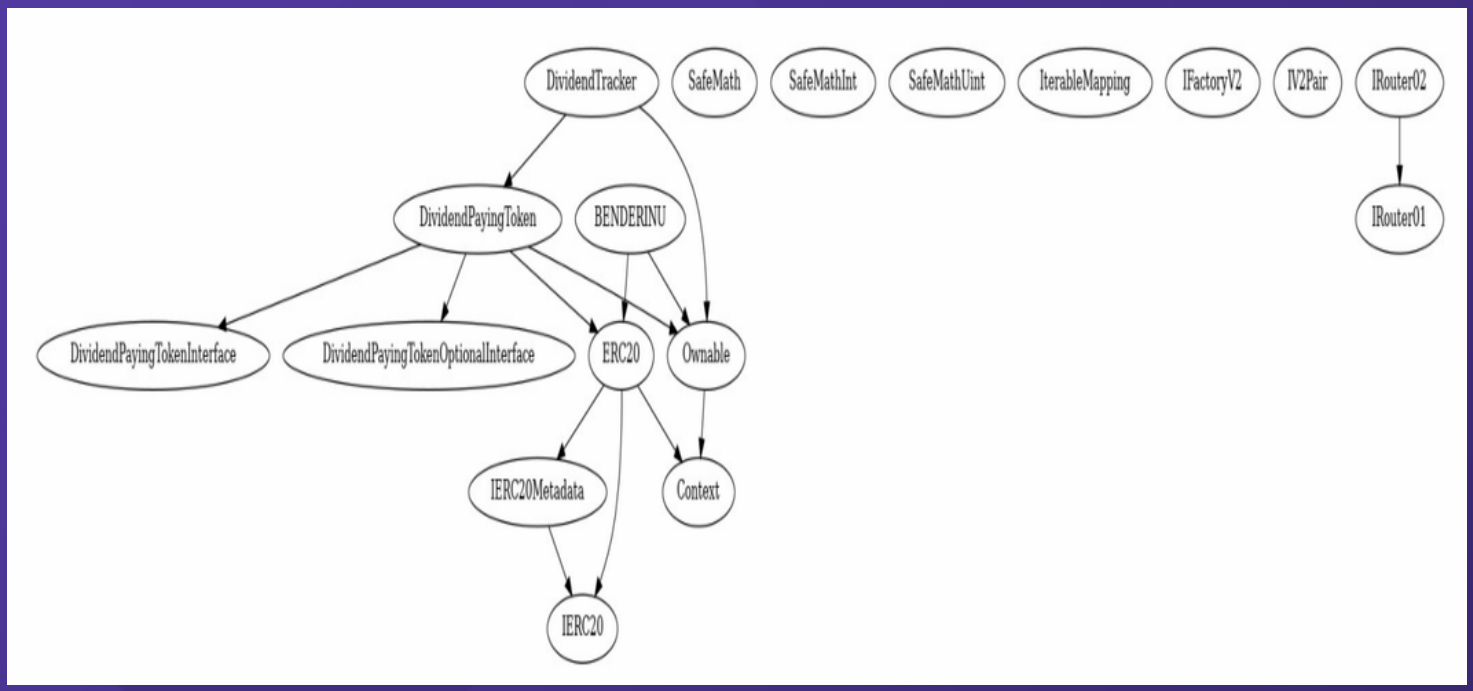
Low Risk

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**
Context	Implementation			
└	<Constructor>	Public ❗	🛑	NO ❗
└	_msgSender	Internal 🔒		
└	_msgData	Internal 🔒		
Ownable	Implementation	Context		
└	<Constructor>	Public ❗	🛑	NO ❗
└	owner	Public ❗		NO ❗
└	renounceOwnership	Public ❗	🛑	onlyOwner
└	transferOwnership	Public ❗	🛑	onlyOwner
└	_setOwner	Private 🔐	🛑	
SafeMath	Library			
└	add	Internal 🔒		
└	sub	Internal 🔒		
└	sub	Internal 🔒		
└	mul	Internal 🔒		
└	div	Internal 🔒		
└	div	Internal 🔒		
└	mod	Internal 🔒		
└	mod	Internal 🔒		
SafeMathInt	Library			
└	mul	Internal 🔒		
└	div	Internal 🔒		
└	sub	Internal 🔒		
└	add	Internal 🔒		
└	abs	Internal 🔒		
└	toUint256Safe	Internal 🔒		
SafeMathUint	Library			
└	toInt256Safe	Internal 🔒		
IterableMapping	Library			

expelee

FUNCTION DETAILS

└	get	Internal 🔒		
└	getIndexOfKey	Internal 🔒		
└	getKeyAtIndex	Internal 🔒		
└	size	Internal 🔒		
└	set	Internal 🔒	🛑	
└	remove	Internal 🔒	🛑	
IFactoryV2	Interface			
└	getPair	External ❗	NO ❗	
└	createPair	External ❗	🛑	NO ❗
IV2Pair	Interface			
└	factory	External ❗	NO ❗	
└	getReserves	External ❗	NO ❗	
└	sync	External ❗	🛑	NO ❗
IRouter01	Interface			
└	factory	External ❗	NO ❗	
└	WETH	External ❗	NO ❗	
└	addLiquidityETH	External ❗	💵	NO ❗
└	addLiquidity	External ❗	🛑	NO ❗
└	swapExactETHForTokens	External ❗	💵	NO ❗
└	getAmountsOut	External ❗	NO ❗	
└	getAmountsIn	External ❗	NO ❗	
IRouter02	Interface	IRouter01		
└	swapExactTokensForETHSupportingFeeOnTransferTokens	External ❗	🛑	NO ❗
└	swapExactETHForTokensSupportingFeeOnTransferTokens	External ❗	💵	NO ❗
└	swapExactTokensForTokensSupportingFeeOnTransferTokens	External ❗	🛑	NO ❗
└	swapExactTokensForTokens	External ❗	🛑	NO ❗
IERC20	Interface			
└	totalSupply	External ❗	NO ❗	
└	balanceOf	External ❗	NO ❗	
└	allowance	External ❗	NO ❗	
└	transfer	External ❗	🛑	NO ❗
└	approve	External ❗	🛑	NO ❗
└	transferFrom	External ❗	🛑	NO ❗

expelee

FUNCTION DETAILS

IERC20Metadata	Interface	IERC20		
└	name	External ❗	|NO ❗
└	symbol	External ❗	|NO ❗
└	decimals	External ❗	|NO ❗
ERC20	Implementation	Context, IERC20, IERC20Metadata		
└	<Constructor>	Public ❗	🛑 |NO ❗
└	name	Public ❗	|NO ❗
└	symbol	Public ❗	|NO ❗
└	decimals	Public ❗	|NO ❗
└	totalSupply	Public ❗	|NO ❗
└	balanceOf	Public ❗	|NO ❗
└	transfer	Public ❗	🛑 |NO ❗
└	allowance	Public ❗	|NO ❗
└	approve	Public ❗	🛑 |NO ❗
└	transferFrom	Public ❗	🛑 |NO ❗
└	increaseAllowance	Public ❗	🛑 |NO ❗
└	decreaseAllowance	Public ❗	🛑 |NO ❗
└	_transfer	Internal 🔒	🛑	
└	_mint	Internal 🔒	🛑	
└	_burn	Internal 🔒	🛑	
└	_approve	Internal 🔒	🛑	
└	_beforeTokenTransfer	Internal 🔒	🛑	
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	Public ❗	🛑	onlyOwner

expelee

FUNCTION DETAILS

└	withdrawDividend	Public ❗	🛑	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	Public ❗		NO ❗
└	updateMinimumTokenBalanceForDividends	External ❗	🛑	onlyOwner
└	excludeFromDividends	External ❗	🛑	onlyOwner
└	updateClaimWait	External ❗	🛑	onlyOwner
└	setLastProcessedIndex	External ❗	🛑	onlyOwner
└	getLastProcessedIndex	External ❗		NO ❗
└	getNumberOfTokenHolders	External ❗		NO ❗
└	getAccount	Public ❗		NO ❗
└	getAccountAtIndex	Public ❗		NO ❗
└	canAutoClaim	Private 🔐	
└	setBalance	External ❗	🛑	onlyOwner
└	process	Public ❗	🛑	NO ❗
└	processAccount	Public ❗	🛑	onlyOwner

BENDERINU	Implementation	ERC20, Ownable
└	<Constructor>	Public ❗	🛑	ERC20
└	<Receive Ether>	External ❗	💵	NO ❗
└	changeWallets	External ❗	🛑	onlyOwner
└	setNoFeeWallet	External ❗	🛑	onlyOwner
└	isNoFeeWallet	Public ❗		NO ❗
└	is_buy	Internal 🔒	
└	is_sell	Internal 🔒	

FUNCTION DETAILS

└	isLimitedAddress	Internal 🔒		
└	enableTrading	External ❗	🛑	onlyOwner
└	_transfer	Internal 🔒	🛑	
└	takeTaxes	Internal 🔒	🛑	
└	swapAndSendDividends	Private 🔐	🛑	
└	setSwapTokensAtAmount	External ❗	🛑	onlyOwner
└	updateClaimWait	External ❗	🛑	onlyOwner
└	getClaimWait	External ❗		NO ❗
└	getTotalDividendsDistributed	External ❗		NO ❗
└	withdrawableDividendOf	Public ❗		NO ❗
└	dividendTokenBalanceOf	Public ❗		NO ❗
└	totalRewardsEarned	Public ❗		NO ❗
└	getAccountDividendsInfo	External ❗		NO ❗
└	getAccountDividendsInfoAtIndex	External ❗		NO ❗
└	processDividendTracker	External ❗	🛑	NO ❗
└	getLastProcessedIndex	External ❗		NO ❗
└	getNumberOfDividendTokenHolders	External ❗		NO ❗
└	claim	External ❗	🛑	NO ❗
└	setLastProcessedIndex	External ❗	🛑	onlyOwner
└	claimAddress	External ❗	🛑	onlyOwner
└	excludeFromDividends	External ❗	🛑	onlyOwner

Legend

Symbol	Meaning
🛑	Function can modify state
💵	Function is payable

TESTNET VERSION

Adding Liquidity ✅
Tx:
https://testnet.bscscan.com/tx/0xb86b5225140b8f3f5745f4e3dacf77ebffed3096ff9d144efc56eb3a15 b960da

==========

Buying when excluded from fees ✅
Tx (0% tax):
https://testnet.bscscan.com/tx/0xa20bf35e7b475d5fde16a33cfcb3af02f5fa76b845b5e189938f05cf8b 648494

==========

Selling when excluded from fees ✅
Tx (0% tax):
https://testnet.bscscan.com/tx/0xd60e05666d73396eb6b4cb10e6d9684cec0abdfcd93708fcff168aa8 487e176a

==========

Transferring when excluded from fees ✅
Tx (0% tax):
https://testnet.bscscan.com/tx/0x7113328596cf3cadbbbc340901a89a91da9a5af6fe955e22e1307732 0b42dcbf

==========

Buying ✅
Tx (5% tax):
https://testnet.bscscan.com/tx/0x078f580e92fc34d529358f768523eab47538769025bdaaa8dd7ab96ca965b1e6

TESTNET VERSION

Selling ✅
Tx (5% tax):
https://testnet.bscscan.com/tx/0x53943494060d7a70552350c711a5286fd517e1f9f35defc0a1458564e12ce311

==========

Transferring ✅
Tx (0% tax):
https://testnet.bscscan.com/tx/0xbccf2e4ce63b6ee169d0bb76f0761765b32640553ece677738e6438824afed73

==========

Internal swap (BNB to marketing wallet | reward token to dividend tracker | reward distrubution) ✅
Tx:
https://testnet.bscscan.com/tx/0x53943494060d7a70552350c711a5286fd517e1f9f35defc0a1458564e12ce311

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

Claiming rewards

Category: Centralization
Subject: Trades are disabled by default
Status: Resolved (contract owned by safu developer)
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(!isTradingEnabled, "Trading already enabled");
  isTradingEnabled = true;
  emit _enableTrading();
}
```

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.