expelee

Building the Futuristic Blockchain Ecosystem

SECURITY AUDIT REPORT

Beanz Finance

TOKEN OVERVIEW

Risk Findings

Severity	Found
High	0
Medium	0
Low	1
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	**No**
Audit Date	**30 November 2023**

CONTRACT DETAILS

Token Name: Beanz Finance

Symbol: BEANZ

Network: Binance

Language: Solidity

Contract Address:

0x37b5f55d5Ab4b8B4a6b87A0763eCe429C1e7b477

Total Supply:

Owner's Wallet:

0xEBDd36C3521B459cd870DE0C0Fb0C67266ef2166

Deployer's Wallet:

0xEBDd36C3521B459cd870DE0C0Fb0C67266ef2166

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

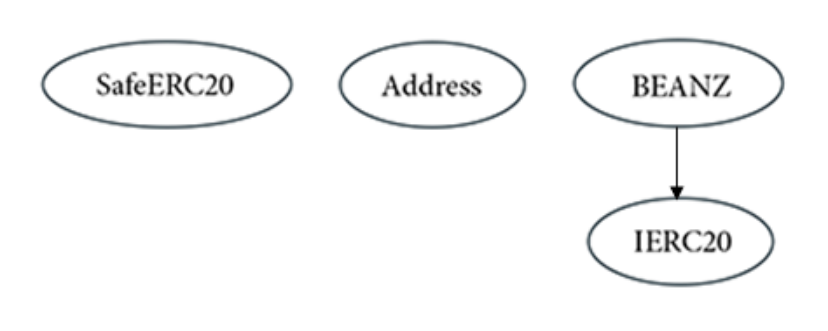
Issues on this level are minor details and warning that can remain unfixed.

Informational

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

expelee

INHERITANCE TREES



expelee

FUNCTION DETAILS

Symbol	Definition
	Function can modify state
	Function is payable

IERC20	Interface			
	totalSupply	External		
	balanceOf	External		
	transfer	External		
	allowance	External		
	approve	External		
	transferFrom	External		
Address	Implementation			
	isContract	Internal		
	sendValue	Internal		
	functionCall	Internal		
	functionCallWithValue	Internal		
	functionStaticCall	Internal		
	functionDelegateCall	Internal		
	verifyCallResult	Internal		
SafeERC20	Implementation			
	safeTransfer	Internal		
	safeTransferFrom	Internal		
	safeApprove	Internal		
	safeIncreaseAllowance	Internal		
	safeDecreaseAllowance	Internal		
	_callOptionalReturn	Private		
BEANZ	Implementation			
	mint	Private		
	burn	Public		
	name	Public		
	symbol	Public		
	decimals	Public		
	totalSupply	Public		
	balanceOf	Public		

FUNCTION DETAILS

	transfer	Public	🛑	
	_transfer	Public	🛑	
	transferFrom	Public	🛑	
	approve	Public	🛑	
	allowance	Public	🛑	
	buy	Public	💸	onlyStarted
	register	Private	🛑	
	checkRef	Private		
	sendRef	Private	🛑	
	sell	Public	🛑	onlyStarted
	sellAll	Public	🛑	onlyStarted
	invest	Public	💸	onlyStarted
	claim	Public	🛑	onlyStarted
	compound	Public	🛑	onlyStarted
	checkpoint	Private	🛑	
	_receive	Private	🛑	
	_receiveBeanz	Private	🛑	
	takeSellTax	Private	🛑	
	addMilestone	Public	🛑	onlyOwner
	updateStartTime	Public	🛑	onlyOwner
	usdtToBeanz	Public		
	beanzToUsdt	Public		
	getPrice	Public		
	getRate	Public		
	getInvestments	Public		
	getContractBalance	Public		
	getUsdtSinceLastClaim	Public		
	getBeanzSinceLastClaim	Public		
	getUserClaimable	Public		

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			

LOW RISK FINDING

Owner can update startTime

Severity: Low

Overview

Owner can update the startTime of the contract if it has not yet started. The input value is not checked and may set a timestamp lower than the current timestamp.

```
function updateStartTime(uint t) public {
    require(block.timestamp < startTime);
    require(msg.sender == dev);
    startTime = t;
}
```

Recommendation

Owner can create a function to control the start of the contract instead of relying on timestamps.

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_official
expelee
expelee
expelee-com

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.