

­­­­­

A Secure Place For Web3

**SMART CONTRACT AUDIT OF**

**ChainBook Presale**

Contract Address

**0x0332BFE76242dc3D3A82bFBad044bF274ff82D85**



www.expelee.com | Page 1 |



**­Audit Summary**

Expelee team has performed a line-by-line manual analysis and automated review of the smart contract. The smartcontract was analysed mainly for common smart contract vulnerabilities, exploits, and manipulation hacks. According to the smart contract audit:

Audit Result: **PASSED (WITH LOW SEVERITY)**

KYC Verification: Not done till date of audit

Audit Date: 28/05/2022

Audit Team: **EXPELEE**

Be aware that smart contracts deployed on the blockchain aren’t resistant to internal exploit, external vulnerability, or hack. For a detailed understanding of risk severity, source code vulnerability, functional hack, and audit disclaimer, kindly refer to the audit.



www.expelee.com | Page 2 |



**DISCLAMER**

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 guarantees against the sale of team tokens or the removal of liquidity by the project audited in this document. Always Do your own research and protect 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. Always 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.



www.expelee.com | Page 3 |



**Contract Review**

|  |  |
| --- | --- |
| **Contract Name** | **ChainBook** |
| **Compiler Version** | **v0.6.12+commit.27d51765** |
| **Optimization** | **No with 200 runs** |
| **License** | **MIT license** |
| **Explorer** | **https://bscscan.com/address/0x0332BFE76242dc3D3A82bFBad044bF274ff82D85#code** |
| **Symbol** | **CNB** |
| **Decimals** | **18** |
| **Total Supply** | **10,000,000,000** |
| **Domain** | **https://chainbook.in/** |



www.expelee.com | Page 4 |



**Project Review**

Token Name: ChainBook

Web Site: https://chainbook.in/

Twitter: https://twitter.com/ChainBookBSC

Telegram: https://t.me/ChainBookGlobal

Contract Address: 0x0332BFE76242dc3D3A82bFBad044bF274ff82D85

Platform: Binance Smart Chain

Token Type: BEP 20

Language: SOLIDITY



www.expelee.com | Page 5 |



Audit Methodology

The scope of this report is to audit the smart contract source code. We have scanned the contract and reviewed the project for common vulnerabilities, exploits, hacks, and back-doors. Below is the list of commonly known smart contract vulnerabilities, exploits, and hacks:

Category

- Unhandled Exceptions - Transaction Order Dependency Smart Contract - Integer Overflow Vulnerabilities - Unrestricted Action - Incorrect Inheritance Order - Typographical Errors - Requirement Violation

Source Code - Gas Limit and Loops Review - Deployment Consistency - Repository Consistency - Data Consistency - Token Supply Manipulation

Functional - Operations Trail & Event Generation Assessment - Assets Manipulation - Liquidity Access



www.expelee.com | Page 6 |



Vulnerability Checklist

|  |  |  |
| --- | --- | --- |
| № | Description. | Result |
| 1 | Compiler warnings. | Passed |
| 2 | Race conditions and Re-entrancy. Cross-function race conditions. | Passed |
| 3 | Possible delays in data delivery. | Passed |
| 4 | Oracle calls. | Passed |
| 5 | Front running. | Passed |
| 6 | Timestamp dependence. | Passed |
| 7 | Integer Overflow and Underflow. | Passed |
| 8 | DoS with Revert. | Passed |
| 9 | DoS with block gas limit. | Passed |
| 10 | Methods execution permissions. | Passed |
| 11 | Economy model. | Passed |
| 12 | The impact of the exchange rate on the logic. | Passed |
| 13 | Private user data leaks. | Passed |
| 14 | Malicious Event log. | Passed |
| 15 | Scoping and Declarations. | Passed |
| 16 | Uninitialized storage pointers. | Passed |
| 17 | Arithmetic accuracy. | Passed |
| 18 | Design Logic. | Passed |
| 19 | Cross-function race conditions. | Passed |
| 20 | Safe Zeppelin module. | Passed |
| 21 | Fallback function security. | Passed |



www.expelee.com | Page 7



Manual Audit

* **Low-Risk**
* 5 low-risk code issues found
* **Medium-Risk**
* 0 medium-risk code issues found
* **High-Risk**
* 0 high-risk code issues found



www.expelee.com | Page 8 |



* **Low-Risk**

1. **Contract contains Reentrancy vulnuerabilities**

function \_transfer( address from, address to,

uint256 amount

) internal override {

require(from != address(0), "ERC20: transfer from the zero address"); require(to != address(0), "ERC20: transfer to the zero address");

require(!\_isBlacklisted[from], 'Blacklisted address');

if(amount == 0) {

super.\_transfer(from, to, 0); return;

}

// No adding liquidity before launched if (!liquidityLaunched) {

if (to == uniswapV2Pair) {

liquidityLaunched = true;

// high tax ends in x blocks

lastSnipeTaxBlock = block.number + snipeBlocks;

}

**Recommendation**

Apply the check-effects-interaction pattern



www.expelee.com | Page 9 |



1. **No zero address validation for some functions**

Detect missing zero address validation.

function setDevWallet(address payable wallet) external onlyOwner{

\_AppWalletAddress = wallet;

}

**Recommendation**

Check that the new address is not zero.



www.expelee.com | Page 10 |

1. **Missing events arithmethics**

Detect missing events for critical arithmetic parameters .

function setFee(

uint256 \_MktFee, uint256 \_DevFee, uint256 \_LpFee,

uint256 \_BurnFee,

uint256 \_UsefulShare, uint256 \_OtherShare

) public onlyOwner { MktFee = \_MktFee; DevFee = \_DevFee; LpFee = \_LpFee;

BurnFee = \_BurnFee;

UsefulShare = \_UsefulShare; OtherShare = \_OtherShare;

AllFee = MktFee.add(DevFee).add(LpFee).add(BurnFee); AllShare = UsefulShare.add(OtherShare);

}

**Recommendation**

Check that the new address is not zero.



www.expelee.com | Page 11 |



1. **Functions that send Ether to arbitary destinations**

Unprotected call to a function sending Ether to arbitary address.

function addLiquidity(uint256 tokenAmount, uint256 ethAmount) private {

// approve token transfer to cover all possible scenarios

\_approve(address(this), address(uniswapV2Router), tokenAmount);

// add the liquidity

uniswapV2Router.addLiquidityETH{value: ethAmount}( address(this),

tokenAmount,

0, // slippage is unavoidable 0, // slippage is unavoidable address(0),

block.timestamp

);

}

**Recommendation**

Ensure that an arbitary user cannot withdraw unauthorized funds



www.expelee.com | Page 12 |



1. **Reductant Statements**

Detect the usage of reductant usage statements that have no effect.

function \_msgData() internal view virtual returns (bytes calldata) {

this; // silence state mutability warning without generating bytecode - see https://github.com/e return msg.data;

}

**Recommendation**

Remove redundant statements if they congest code but offer no value.



www.expelee.com | Page 13 |



Manual Audit (Contract Function)

contract ChainBook is ERC20, Ownable { using SafeMath for uint256;

IUniswapV2Router02 public uniswapV2Router; address public uniswapV2Pair;

bool private swapping;

address public deadWallet = 0x000000000000000000000000000000000000dEaD;

address public \_AppWalletAddress = 0xdf3bcfEdc368142c3477D4F0A44d80844D8D7450; address public \_FoudWalletAddress = 0x568db913f1e2EA2Fb788220552F49954Ec3B3e85;

uint256 public totalSupply\_ = 100 \* (10\*\*8) \* (10\*\*18); uint256 public swapTokensAtAmount;

uint256 public burnEndAmount;



www.expelee.com | Page 14 |



Important Points To Consider

✔ The owner cannot mint tokens after Initial

✔ The owner cannot stop Trading.

✔ Verified contract source

✔ Token is sellable (not a honeypot) at this time

✔ Ownership renounced or source does not contain an owner contract

✔Creator not authorized for special permission

✔ Source does not contain a fee modifier

✔ Buy fee is less than 10% (8%)

✔ Sell fee is less than 10% (8%)

✔ Owner/creator wallet contains less than 10% of circulating token supply (0.01%)

— Tokens burned: 60.02%, circulating supply: 3,997,288,736.852



www.expelee.com | Page 15 |

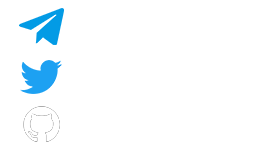
About Expelee

Expelee is a community driven organisation dedicated to fostering an anti-rug movement. We're here to keep investment safe from fraudsters. We've encountered several rug pulls and know how it feels to be duped, which is why we don't want anybody else to go through the same experience. We are here to raise awareness through ourservices so that the future of cryptocurrency can be rug-free.

The auditing process focuses to the following considerations with collaboration of an expert team:

* Functionality test of the Smart Contract to determine if proper logic has been followed throughout the whole process.
* Manually detailed examination of the code line by line by experts.
* Live test by multiple clients using Test net.
* Analysing failure preparations to check how the Smart
* Contract performs in case of any bugs and vulnerabilities.
* Checking whether all the libraries used in the code are on the latest version.
* Analysing the security of the on-chain data.

Social Media





www.expelee.com | Page 16 |