expelee

Building the Futuristic **Blockchain Ecosystem**

Audit Report

FOR

OVERVIEW

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 KYC
Audit Date	8 Sep 2022

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.

- Team Expelee

PROJECT DESCRIPTION

ShibChain

Shibchain supercharges $Dogecoin to bring crypto applications like NFTs, games, and DeFi to the $Dogecoin community. Unfortunately, $Dogecoin cannot support any of these applications (or any other dApps). Shibchain fixes this.
WIth Shibchain, Dogecoin holders can do more than simply hodl and wait for Dogecoin to moon!

shibchain.echostudio.xyz

shibchainarmy

shibchainarmy

It's always good to check the social profiles of the project, before making your investment.

- Team Expelee

CONTRACT DETAILS

Contract Name
SHIBCHAIN

Symbol
$SC

Contract Address
0x2dFf9c00Fac681131cbbe36Fb3F5454cc366bB7c

Network
BSC

Language
Solidity

Total Supply
1,000,000,000

Decimals
18

Compiler
v0.8.9+commit.e5eed63a

License
Default license

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
- Complier
- Hardhat

FUNCTION OVERVIEW

Can Take Back Ownership	Not Detected
Owner Change Balance	Not Detected
Blacklist	Not Detected
Modify Fees	Detected
Proxy	Not Detected
Whitelisted	Not Detected
Anti Whale	Detected
Trading Cooldown	Not Detected
Transfer Pausable	Detected
Cannot Sell All	Not Detected
Hidden Owner	Not Detected
Creator Address	0xF6e4E693f3bc2a67906b86981aa37fb9b719B2D9
Creator Balance	243,500,000 SC
Owner Address	0xF6e4E693f3bc2a67906b86981aa37fb9b719B2D9
Mint	Not Detected

VULNERABILITY CHECKLIST

Design Logic	Passed
Compiler warnings.	Passed
Private user data leaks	Passed
Timestamp 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 Zeppelin module	Passed
Fallback function security	Passed

RISK CLASSIFICATION

When performing smart contract audits, our specialists look for known vulnerabilities as well as logical and access 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

Information level is to offer suggestions for improvement of efficacy or security for features with a risk free factor.

MANUAL AUDIT

- Launch tested on local blockchain, there was no issues with buying, selling and swapping contract collected fees.
- Owner is not able to set taxes more than 10% on buys / sells
- Owner is not able to disable trading after enabling it

Centralization Issues

- **High-** owner must call **enableTrading** function to enable trading for everyone, otherwise, no one is able to buy or sell except excluded wallets.
- **Medium-** owner is able to set a max transaction amount, non-whitelisted wallets can't buy or sell more than this amount.this limit can't be lower than 0.001% of total supply.

```
function updateMaxTxnAmount(uint256 newNum) external onlyOwner {
        require(newNum >= (totalSupply() * 1 / 1000)/1e18, "Cannot set
maxTransactionAmount lower than 0.1%");
        maxTransactionAmount = newNum * (10**18);
    }
```

- **Medium-** owner is able to set a max wallet amount, non-whitelisted wallets can't hold more than this amount. this limit can't be lower than 0.005% of total supply.

```
function updateMaxWalletAmount(uint256 newNum) external onlyOwner {
        require(newNum >= (totalSupply() * 5 / 1000)/1e18, "Cannot set
maxWallet lower than 0.5%");
        maxWallet = newNum * (10**18);
    }
```

- **Medium-** a malicious owner is able to set dev wallet or marketing wallet to a contract that doesn't acceprt ether, doing this reverts all sells after contract balance reaching a threshold.

Logical Issues

- **Medium-** require statement doesn't match error message at this conditions, match error message with required conditions:

```
require(newAmount >= totalSupply() * 1 / 100000, "Swap amount cannot be
lower than 0.001% total supply.");
require(newAmount <= totalSupply() * 5 / 1000, "Swap amount cannot be higher
than 0.5% total supply.");
```

```
require(newNum >= (totalSupply() * 1 / 1000)/1e18, "Cannot set
maxTransactionAmount lower than 0.1%");
```

- **low-** `require(pair != pancakePair, "FRTNA::The PancakeSwap pair cannot be removed from automatedMarketMakerPairs");`
  at function **setAutomatedMarketMakerPair**, this require statement is not matching error message. the problem is that if you make a new pair and then decide to add it here as a automatedMarketMaker, you are not able to do so because this function will revert even if **value** is true

Gas Optimizations

- **GO-1: launchedAt** variable never used inside contract, its also a private variable and can't not be readed, delete this variable from contract or make this variable public
- **GO-2:** line 1132, checking if owner() is **to** is redundant

Suggestions

- **S-1:** once **removeLimits()** function is used limitsInEffect will be false forever, there is no way to change it to true again, same thing happens with **disableTransferDelay()**
- **S-2:** emit an event for this functions: **updateSwapEnabled**, **excludeFromMaxTransaction**, **updateMaxWalletAmount**, **updateMaxTxnAmount**, **updateSwapTokensAtAmount**, **disableTransferDelay**, **removeLimits**, **enableTrading**
- **S-3:** avoid potential sandwich attacks by providing an output amount at this functions: **addLiquidity**, **swapTokensForEth**

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 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.