expelee

A Secure Place For Web3

SMART CONTRACT AUDIT OF

SAFUU CLASSIC

Contract Address

0xD724Bbe5B419394E370200e7D2370F731678cB48

Audit Summary

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**

Ownership: **NOT RENOUNCED**

KYC Verification: Done

Audit Date: 13/07/2022

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.

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.

Contract Review

Contract Name	**SafuuClassic**
Compiler Version	**v0.7.4+commit.3f05b770**
Optimization	**Yes with 200 runs**
License	**None license**
Explorer	**https://bscscan.com/address/0xD724Bbe5B419394E370200e7D2370F731678cB48#code**
Symbol	**SAFUUC**
Decimals	**5**
Total Supply	**376,972.67565**
Domain	**https://safuuclassic.com/**

Project Review

Token Name: SAFUU CLASSIC

Web Site: https://safuuclassic.com/

Twitter: https://twitter.com/SafuuClassic

Telegram: https://t.me/safuuclassic

Contract Address:
0xD724Bbe5B419394E370200e7D2370F731678cB48

Platform: Binance Smart Chain

Token Type: BEP 20

Language: SOLIDITY

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

Smart Contract Vulnerabilities	- Unhandled Exceptions - Transaction Order Dependency - Integer Overflow - Unrestricted Action - Incorrect Inheritance Order - Typographical Errors - Requirement Violation
Source Code Review	- Gas Limit and Loops - Deployment Consistency - Repository Consistency - Data Consistency - Token Supply Manipulation
Functional Assessment	- Operations Trail & Event Generation - Assets Manipulation - Liquidity Access

Vulnerability Checklist

№	Description.	Result
1	Compiler warnings.	Passed
2	Race conditions and Re-entrancy. Cross-function raceconditions.	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

Manual Audit

Low-Risk

3 low-risk code issues found

Medium-Risk

0 medium-risk code issues found

High-Risk

0 high-risk code issues found

Low-Risk

1) Avoid reyling on block.timestamp

Block.timestamp may be manipulated by miners.

```
function rebase() internal {


    if ( inSwap ) return;uint256
    rebaseRate;

    uint256 deltaTimeFromInit = block.timestamp - _initRebaseStartTime;uint256
    deltaTime = block.timestamp - _lastRebasedTime;

    uint256 times = deltaTime.div(15 minutes);uint256
    epoch = times.mul(15);


    if (deltaTimeFromInit = (7 * 365 days)) {rebaseRate =
        2;

    } else if (deltaTimeFromInit &gt;= ((15 * 365 days) / 10)) {rebaseRate =
        14;

    }else if (deltaTimeFromInit &gt;= (365 days)) {rebaseRate =
        211;

    }


    for (uint256 i = 0; i &lt; times; i++) {

        _totalSupply = _totalSupply

            .mul((10**RATE_DECIMALS).add(rebaseRate))

            .div(10**RATE_DECIMALS);

    }
```

Recommendation

Do not use block.timestamp as a source of randomness.

2) Functions that send Ether to arbitary destinations

Unprotected call to a function sending Ether to arbitary address.

```
function swapBack() internal swapping {

    uint256 amountToSwap = _gonBalances[address(this)].div(_gonsPerFragment);if( amountToSwap == 0) {
        return;
    }

    uint256 balanceBefore = address(this).balance;address[] memory path = new
    address[](2);
    path[0] = address(this);
    path[1] = router.WETH();

router.swapExactTokensForETHSupportingFeeOnTransferTokens(
    amountToSwap,0,
    path,
    address(this), block.timestamp
);
```

Recommendation

Ensure that an arbitary user cannot withdraw unauthorized funds

3) Contract contains Reentrancy vulnuerabilities

```
function _transferFrom(address
    sender,
    address recipient,uint256
    amount
) internal returns (bool) {

    require(!blacklist[sender] &amp;amp;&amp;amp; !blacklist[recipient], "in_blacklist");if (inSwap)
    {
        return _basicTransfer(sender, recipient, amount);
    }
    if (shouldRebase()) {rebase();
    }

    if (shouldAddLiquidity()) {
        addLiquidity();
    }

    if (shouldSwapBack()) {swapBack();
    }
```

Recommendation

Apply the check-effects-interaction pattern

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 NOT renounced or source does contain an ownership functionality

✗ Buy fee is less than 10% (14%)

✗ Sell fee is less than 10% (16%)

✗ Source doesn’t contain fee modifier

✓ Owner wallet contains less than 10% of circulating token supply (6.93%)

✓Creator wallet contains less than 10% of circulating token supply (0%)

✓ All other holders possess less than 10% of circulating token supply

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 our services 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

/Expelee

/ExpeleeOfficial

/expelee-co