



# Smart Contract Audit

FOR

## Floki Chairman

DATED : 13 MAR 23'



# AUDIT SUMMARY

---

**Project name** – Floki Chairman

**Date:** 13 March, 2023

**Scope of Audit-** Audit Ace was consulted to conduct the smart contract audit of the solidity source codes.

**Audit Status:** **Passed**

## Issues Found

Status	Critical	High	Medium	Low	Suggestion
Open	0	0	2	0	0
Acknowledged	0	0	0	0	0
Resolved	0	0	0	0	0

---

# USED TOOLS

---

## Tools:

### 1- Manual Review:

a line by line code review has been performed by audit ace team.

### 2- BSC Testnet network:

all tests were done on Bsc Testnet network, each test has its transaction has attached to it.

### 3- Slither : Static Analysis

**Testnet Link:** all tests were done using this contract, tests are done on BSC Testnet

<https://testnet.bscscan.com/token/0x40C14cBa93b2658aC67E9Ce812f04858abDFC72d#code>

---



# Token Information

---

**Token Name :** FLOKI CHAIRMAN

**Token Symbol:** CHAIRMAN

**Decimals:** 9

**Token Supply:** 100,000,000,000

**Token Address:**

0x82A3598dC39b426f47508E2A9a091F753b8ab710

**Checksum:**

f6cd3819dfe750f683aaa67ce06b2676af8b0447

**Owner:**

0x047400e53694F803e25A35945e0E97EDA051b0a6

---



# TOKEN OVERVIEW

---

## **Fees:**

Buy Fees: 8%

Sell Fees: 8%

Transfer Fees: 8%

---

**Fees Privilege:** None

---

**Ownership :** Owned

---

**Minting:** No mint function

---

**Max Tx Amount/ Max Wallet Amount:** No

---

**Blacklist:** No

---

**Other Privileges:** including and excluding from fees and rewards

---

---



# AUDIT METHODOLOGY

---

The auditing process will follow a routine as special considerations by Auditace:

- Review of the specifications, sources, and instructions provided to Auditace to make sure the contract logic meets the intentions of the client without exposing the user's funds to risk.
  - Manual review of the entire codebase by our experts, which is the process of reading source code line-by-line in an attempt to identify potential vulnerabilities.
  - Specification comparison is the process of checking whether the code does what the specifications, sources, and instructions provided to Auditace describe.
  - Test coverage analysis determines whether the test cases are covering the code and how much code is exercised when we run the test cases.
  - Symbolic execution is analysing a program to determine what inputs cause each part of a program to execute.
  - Reviewing the codebase to improve maintainability, security, and control based on the established industry and academic practices.
-

# VULNERABILITY CHECKLIST

---

- |  |   |
|--|---|
|  Return values of low-level calls  |  Gasless Send                  |
|  Private modifier                  |  Using block.timestamp         |
|  Multiple Sends                    |  Re-entrancy                   |
|  Using Suicide                    |  Tautology or contradiction   |
|  Gas Limitand Loops              |  Timestamp Dependence        |
|  Address hardcoded               |  Revert/require functions    |
|  Exception Disorder              |  Use of tx.origin            |
|  Using inline assembly           |  Integer overflow/underflow  |
|  Divide before multiply          |  Dangerous strict equalities |
|  Missing Zero Address Validation |  Using SHA3                  |
|  Compiler version not fixed      |  Using throw                 |
-

# CLASSIFICATION OF RISK

## Severity

## Description

### ◆ Critical

These vulnerabilities could be exploited easily and can lead to asset loss, data loss, asset, or data manipulation. They should be fixed right away.

### ◆ High-Risk

A vulnerability that affects the desired outcome when using a contract, or provides the opportunity to use a contract in an unintended way.

### ◆ Medium-Risk

A vulnerability that could affect the desired outcome of executing the contract in a specific scenario.

### ◆ Low-Risk

A vulnerability that does not have a significant impact on possible scenarios for the use of the contract and is probably subjective.

### ◆ Gas Optimization /Suggestion

A vulnerability that has an informational character but is not affecting any of the code.

## Findings

### Severity

### Found

#### ◆ Critical

0

#### ◆ High-Risk

0

#### ◆ Medium-Risk

2

#### ◆ Low-Risk

0

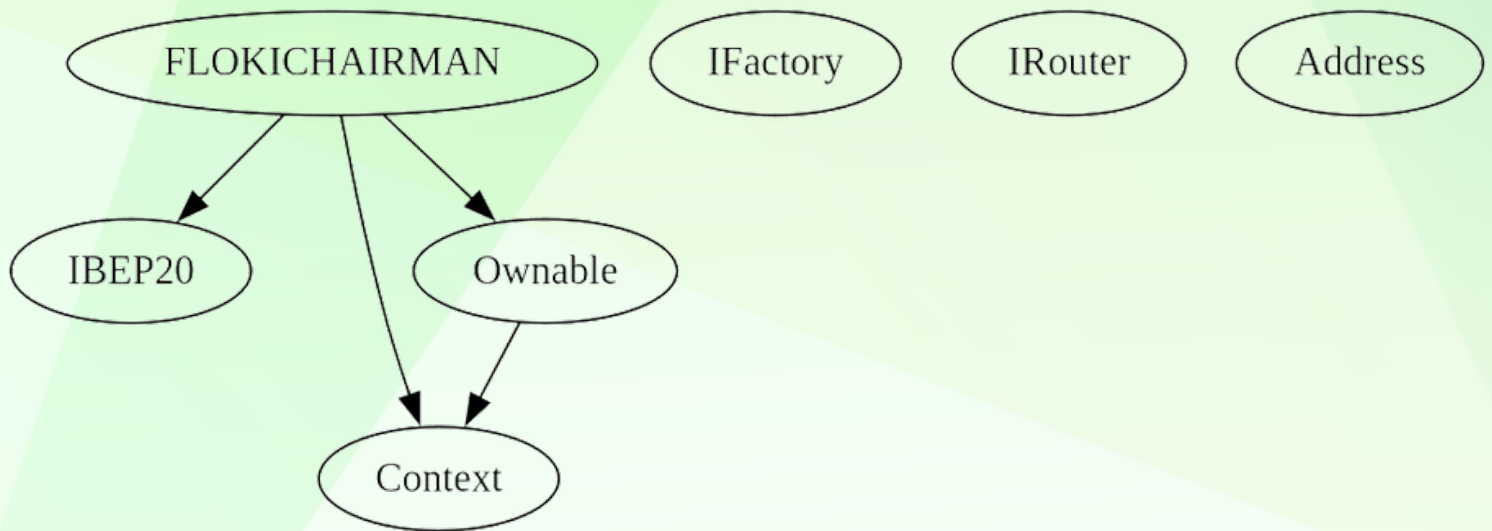
#### ◆ Gas Optimization / Suggestions

0



# INHERITANCE TREE

---





# POINTS TO NOTE

---

- Owner is not able to change fees (8% fee buy sell and transfers)
  - Owner is not able to set max buy/sell/transfer/hold amount
  - Owner is not able to blacklist an arbitrary wallet
  - Owner is not able to disable trades
  - Owner is not able to mint new tokens
  - **Owner must enable trading, otherwise holders will not be able to trade**
-



# TOKEN DISTRIBUTION

---

**It should be noted that the owner currently holds 100% of the total supply. However, information about the distribution of these tokens is not available, and it is recommended that investors exercise caution when considering this aspect.**

---

# CONTRACT ASSESMENT

Contract	Type	Bases			
└──	└──	└──	└──	└──	└──
Function Name	Visibility	Mutability	Modifiers		
IBEP20	Interface				
└── totalSupply	External	!	NO	!	
└── balanceOf	External	!	NO	!	
└── transfer	External	!	⊗	NO	!
└── allowance	External	!		NO	!
└── approve	External	!	⊗	NO	!
└── transferFrom	External	!	⊗	NO	!
Context	Implementation				
└── _msgSender	Internal	🔒			
└── _msgData	Internal	🔒			
Ownable	Implementation	Context			
└── <Constructor>	Public	!	⊗	NO	!
└── owner	Public	!		NO	!
└── renounceOwnership	Public	!	⊗	onlyOwner	
└── transferOwnership	Public	!	⊗	onlyOwner	
└── _setOwner	Private	🔒	⊗		
IFactory	Interface				
└── createPair	External	!	⊗	NO	!
IRouter	Interface				
└── factory	External	!		NO	!
└── WETH	External	!		NO	!
└── addLiquidityETH	External	!	🟢	NO	!
└── swapExactTokensForETHSupportingFeeOnTransferTokens	External	!	⊗	NO	!
Address	Library				
└── sendValue	Internal	🔒	⊗		
FLOKICHAIRMAN	Implementation	Context, IBEP20, Ownable			
└── <Constructor>	Public	!	⊗	NO	!
└── name	Public	!		NO	!
└── symbol	Public	!		NO	!
└── decimals	Public	!		NO	!
└── totalSupply	Public	!		NO	!
└── balanceOf	Public	!		NO	!





# STATIC ANALYSIS

```
Reentrancy in FLOKICHAIRMAN.transferFrom(address,address,uint256) (contracts/Token.sol#264-279):
  External calls:
    - _transfer(sender,recipient,amount) (contracts/Token.sol#269)
      - (success) = recipient.call{value: amount}{} (contracts/Token.sol#137)
      - router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,address(this),block.timestamp) (contracts/Token.sol#549-555)
      - address(marketingWallet).sendValue(deltaBalance) (contracts/Token.sol#536)
  External calls sending eth:
    - _transfer(sender,recipient,amount) (contracts/Token.sol#269)
      - (success) = recipient.call{value: amount}{} (contracts/Token.sol#137)
  Event emitted after the call(s):
    - Approval(owner,spender,amount) (contracts/Token.sol#474)
      - approve(sender,msgSender(),currentAllowance - amount) (contracts/Token.sol#276)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

FLOKICHAIRMAN.includeInReward(address) (contracts/Token.sol#354-365) has costly operations inside a loop:
  - excluded.pop() (contracts/Token.sol#361)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#costly-operations-inside-a-loop

Context._msgData() (contracts/Token.sol#51-54) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

FLOKICHAIRMAN._rTotal (contracts/Token.sol#165) is set pre-construction with a non-constant function or state variable:
  - (MAX - (MAX % _tTotal))
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#function-initializing-state

Pragma version^0.8.17 (contracts/Token.sol#12) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.16
solc-0.8.18 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (contracts/Token.sol#131-142):
  - (success) = recipient.call{value: amount}{} (contracts/Token.sol#137)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function IRouter.WETH() (contracts/Token.sol#107) is not in mixedCase
Struct FLOKICHAIRMAN.valuesFromGetValues (contracts/Token.sol#189-197) is not in CapWords
Parameter FLOKICHAIRMAN.rescueAnyBEP20Tokens(address,address,uint256)._tokenAddr (contracts/Token.sol#588) is not in mixedCase
Parameter FLOKICHAIRMAN.rescueAnyBEP20Tokens(address,address,uint256)._to (contracts/Token.sol#589) is not in mixedCase
Parameter FLOKICHAIRMAN.rescueAnyBEP20Tokens(address,address,uint256)._amount (contracts/Token.sol#590) is not in mixedCase
Constant FLOKICHAIRMAN._decimals (contracts/Token.sol#161) is not in UPPER CASE WITH UNDERSCORES
Constant FLOKICHAIRMAN._name (contracts/Token.sol#172) is not in UPPER CASE WITH UNDERSCORES
Constant FLOKICHAIRMAN._symbol (contracts/Token.sol#173) is not in UPPER CASE WITH UNDERSCORES
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (contracts/Token.sol#52)" inContext (contracts/Token.sol#46-55)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

FLOKICHAIRMAN._tTotal (contracts/Token.sol#164) should be constant
FLOKICHAIRMAN.deadWallet (contracts/Token.sol#169) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant

FLOKICHAIRMAN.pair (contracts/Token.sol#159) should be immutable
FLOKICHAIRMAN.router (contracts/Token.sol#158) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
```

**Result => A static analysis of contract's source code has been performed using slither,**

**No issues found**



# FUNCTIONAL TESTING

---

## Router (PCS V2):

0xD99D1c33F9fC3444f8101754aBC46c52416550D1

### 1- Adding Liquidity (Passed):

liquidity added on Pancakeswap V2:

<https://testnet.bscscan.com/tx/0x3442b73a1335a75d410a030c91771140fcd51ad7d78c9e43dd7b1e268f78f5bb>

### 2- Buying when excluded (0% )(Passed):

<https://testnet.bscscan.com/tx/0x9ae3910957189b907722e27aae2fc7abebb81bd56f8052fb4091f3ecbb3eacdd>

### 3- Selling when excluded (0% )(Passed):

<https://testnet.bscscan.com/tx/0x7f3a0f583ccd2a21ff018b3ee4a43343fe332e72b11f7d9b89b185f6dbff73a9>

### 4- Transferring when excluded (0% tax) (passed):

<https://testnet.bscscan.com/tx/0x4b9e2ce325dd17149ed3e32c13a9bb9280975f304f8e9edeb39e23e7bd0b42ca>

### 5- Buying when not excluded (8% tax) (passed):

<https://testnet.bscscan.com/tx/0xb4f2cd1f165c23d1aa525dcee44cb6500a19876c586762ed9a510bff603a2d3a>

---



# FUNCTIONAL TESTING

---

## 6- Selling when not excluded (8% tax) (**passed**):

<https://testnet.bscscan.com/tx/0xfa986944f301fd29d3f34996009739c201b3a7fa8c141d483c88ade28b66c33b>

## 7- Transferring when not excluded(8% tax) (**passed**):

<https://testnet.bscscan.com/tx/0xb3169db4388f40dbb23682307b9d65a34c7d0e6aa6b6ef23fbce01c869a63aa6>

## 8- Internal swap (**passed**):

**marketing wallet received ETH**

<https://testnet.bscscan.com/address/0x37c55fdc707cbbd0dfca25a14d06f9840e6ef085#internaltx>

## 9- Reflections (**passed**):

we monitors wallet balances for testing this features, wallets received reflection after trades, they stop getting reflections after getting excluded from rewards

---



# MANUAL TESTING

---

Issue: swap threshold can revert some sells

Type: logical

Function: updateSwapTokensAtAmount

Line: 535-538

Severity: **Medium**

## Overview:

If swap threshold and contract balance are both zero, **swapAndLiquify** function will fail the transaction.

```
function swapAndLiquify() private lockTheSwap {
    uint256 contractBalance = balanceOf(address(this));
    swapTokensForBNB(contractBalance);
    uint256 deltaBalance = address(this).balance;

    if (deltaBalance > 0) {
        payable(marketingWallet).sendValue(deltaBalance);
    }
}
```

## Recommendation:

- make sure that swap threshold is always higher than 0

# MANUAL TESTING

---

Issue: Trades must be enabled by owner

Type: Centralization

Function: **EnableTrading**

Line: 475-478

Severity: **Medium - Informational**

## Overview:

Owner must enable trading otherwise holders of the token will not be able to trade (sell/transfer) their tokens. However once trading is enabled, owner is not able to disable it again

```
function EnableTrading() external onlyOwner {  
    require(!tradingEnabled, "Cannot use this function again");  
    tradingEnabled = true;  
}
```

# Social Media Overview

---

**Here are the Social Media Accounts of  
Floki Chairman**



**<https://t.me/flokichairmangroup>**



**<https://twitter.com/FlokiChairman>**



**<https://www.flokichairman.com>**

---



# 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 Auditace 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 Auditace 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 Auditace team disclaims any liability for the resulting losses.

---



# ABOUT AUDITACE

---

We specialize in providing thorough and reliable audits for Web3 projects. With a team of experienced professionals, we use cutting-edge technology and rigorous methodologies to evaluate the security and integrity of blockchain systems. We are committed to helping our clients ensure the safety and transparency of their digital assets and transactions.



**<https://auditace.tech/>**



**[https://t.me/Audit\\_Ace](https://t.me/Audit_Ace)**



**[https://twitter.com/auditace\\_](https://twitter.com/auditace_)**



**<https://github.com/Audit-Ace>**

---