

# Blue Floki Smart Contract Audit

November 3, 2021

# Table of Contents

1

## Smart Contract - Audit Overview

A general overview of our findings. This includes the project summary, the audit summary, and the vulnerabilities summary.

2

## Smart Contract - Contract Overview

A general overview of our findings. This includes contract name, ticker, addresses, token holders/transfers etc.

3

## Smart Contract - Vulnerabilities

We cover the majority of the vulnerabilities found in the SWC registry and lay them out in an easy-to-read table. Aside from the SWC Registry, we also conduct a line-by-line analysis to watch for common errors and exploits.

4

## Smart Contract – Code Analysis

The code analysis is the complete overview of the vulnerabilities assessment showing all the issues we found whether they are low severity or high severity.

5

## Contract Ownership & Mint Function

Investors need to know what role the project owners play in ability to change features and settings within the contract. In this section we take a deep dive into ownership privileges and ability to mint new tokens.

6

## Smart Contract - Tokenomics

Tokenomics vary by each project, that come in the form of a tax per each transfer. Most common tax transfers are seen as rewards, liquidity, marketing, burns/buybacks.

7

## Token Holdings & Analytics

An easy way to spot risk is to take a look at the top token holders. We list them out for you to review for yourself. All data is provided by block explorer sites.

8

## Team Overview

The captain is the most important part of the ship. This section takes a look at the team – whether they are anonymous or public and provides all the information we can get our hands on.

# DISCLAIMER

Audits.finance Inc. is in no way responsible or liable for any legal actions resulting from the use of this presentation. By reading this audit or any part of it, you agree to the terms of this disclaimer. If you do not agree to these terms, please stop reading now, and delete any duplicates of this report. Audits.finance Inc. is an official auditor utilizing the Solidity auditing industry standard. Audits.finance hereby excludes any liability and responsibility. Neither you nor any other person shall have any claim against Audits.finance for any economic loss or damages. Audits.finance Inc. does not guarantee the authenticity of a project, nor does it guarantee the project will not participate in one or any scamming including but not limited to, removing liquidity, selling off team supply, or exit scams. Audits.finance Inc. does not give investment advice in any way. Audits.finance Inc. supplies this presentation for information purposes only, and strongly suggests that none of this information be used as investment advice. Audits.finance in no way endorses or recommends any projects that it audits. Audits.finance is solely responsible for smart contract and project analysis of the projects that it is contracted to audit. Audits.finance may be contracted by teams, investors, or any other 3rd party in regard to a contract address or project. Audits.finance provides a full report for informational purposes only.

# Smart Contract – Audit Overview






## Project Summary

Project Name	Blue Floki
Platform	Binance Smart Chain
Language	Solidity
Commits	0x7aD8383Ac98F20B23f133160D9634C22931dD24E

## Audit Summary

Delivery Date	November 3, 2021
Method of Audit	Human and AI
Consultants Engaged	One
Timeline	November 2, 2021 – November 4, 2021

## Vulnerability Summary

Vulnerability Level	Total	Resolved
 Critical	0	✓
 Major	0	✓
 Medium	0	✓
 Minor	1	X
 Informational	1	X

# Smart Contract - Contract Overview

All information is recorded as of 11/03/2021.

Contract Name	BLUEFLOKI
Contract Ticker	BLUEFLOKI
Contract Address	0x7aD8383Ac98F20B23f133160D9634C22931dD24E
Contract Creator	0xb1E856636284Fbbfc690605dBcff5D57662e6529
Decimals	9
Total Supply	1,000,000,000,000,000
Token Holders	1
Token Transfers	1
Compiler Version	v0.6.12+commit.27d51765
Source Code	Solidity
Optimization Enabled	No with 200 runs
Other Settings	default evmVersion, None license

# Smart Contract - Vulnerabilities

Vulnerability Tested	Human Review	Ai Review	Lines Affected	Results
Function Default Visibility				
Integer Overflow and Underflow				
Outdated Compiler Version				
FloatingPragma			L9	
Unchecked Call Return Value				
Unprotected Ether Withdrawal				
Unprotected SELFDESTRUCT Instruction				
Unencrypted Private Data On-Chain				

# Smart Contract - Vulnerabilities

Vulnerability Tested	Human Review	Ai Review	Line(s) Affected	Results
Reentrancy				
State Variable Default Visibility			L728	
Uninitialized Storage Pointer				
Assert Violation				
Use of Deprecated Solidity Functions				
Delegatecall to Untrusted Callee				
DoS with Failed Call				
Code With No Effects				

# Smart Contract - Vulnerabilities

Vulnerability Tested	Human Review	Ai Review	Line(s) Affected	Results
Transaction Order Dependence				
Authorization through tx.origin				
Block values as a proxy for time				
Signature Malleability				
Incorrect Constructor Name				
Shadowing State Variables				
Weak Sources of Randomness from Chain Attributes				



# Smart Contract - Vulnerabilities

Vulnerability Tested	Human Review	Ai Review	Line(s) Affected	Results
Missing Protection against Signature Replay Attacks				
Lack of Proper Signature Verification				
Requirement Violation				
Write to Arbitrary Storage Location				
Incorrect Inheritance Order				
Insufficient Gas Griefing				
Arbitrary Jump with Function Type Variable				

# Smart Contract - Vulnerabilities

Vulnerability Tested	Human Review	Ai Review	Line(s) Affected	Results
DoS With Block Gas Limit				
Typographical Error				
Right-To-Left-Override control character				
Presence of unused variables				
Unexpected Ether balance				
Hash Collisions With Multiple Variable Length Arguments				
Message call with hardcoded gas amount				

# Smart Contract - Code Analysis

Floating Pragma

Severity: **Informational**

Blueflok.sol

Line: 9

The current pragma Solidity directive is ""^0.6.12"". It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

```
7  */
8
9  pragma solidity ^0.6.12;
10 // SPDX-License-Identifier: Unlicensed
11 interface IERC20 {
```

State variable visibility is not set.

Severity: **Low**

Blueflok.sol

Line: 728

It is best practice to set the visibility of state variables explicitly. The default visibility for "inSwapAndLiquify" is internal. Other possible visibility settings are public and private.

```
726 address public immutable uniswapV2Pair;
727
728 bool inSwapAndLiquify;
729 bool public swapAndLiquifyEnabled = true;
```

# Smart Contract – Mint function

This contract cannot mint new BLUEFLOKI.  
We were unable to locate a mint function that  
is used to mint new Blue Floki tokens.



## Smart Contract – Contract Ownership

Contract ownership  
has not been  
renounced at the time  
of the audit. The  
owner's address is  
shown as:

**0xb1e856636284fbbf  
c690605dbcff5d5766  
2e6529**

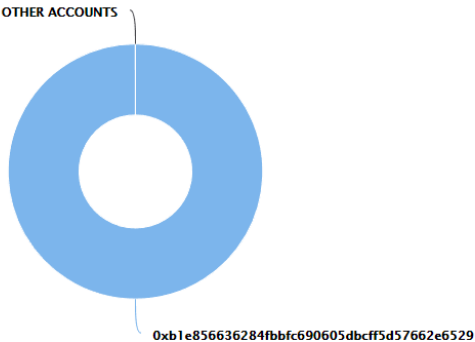
# Smart Contract - Tokenomics

At the time of Audit the transaction fees ("tax") listed below are the fees associated with trading. These fees are taken from every buy and sell transaction unless otherwise stated. Token taxes vary by each project.

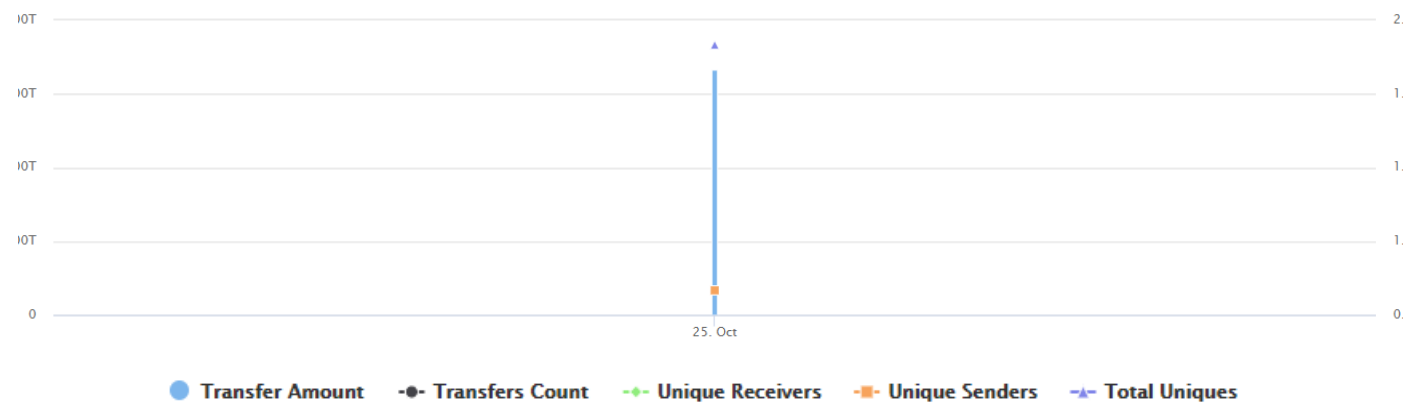


# Token Holders & Contract Analytics

## Top 100 Token Holders



## Token Contract Analytics

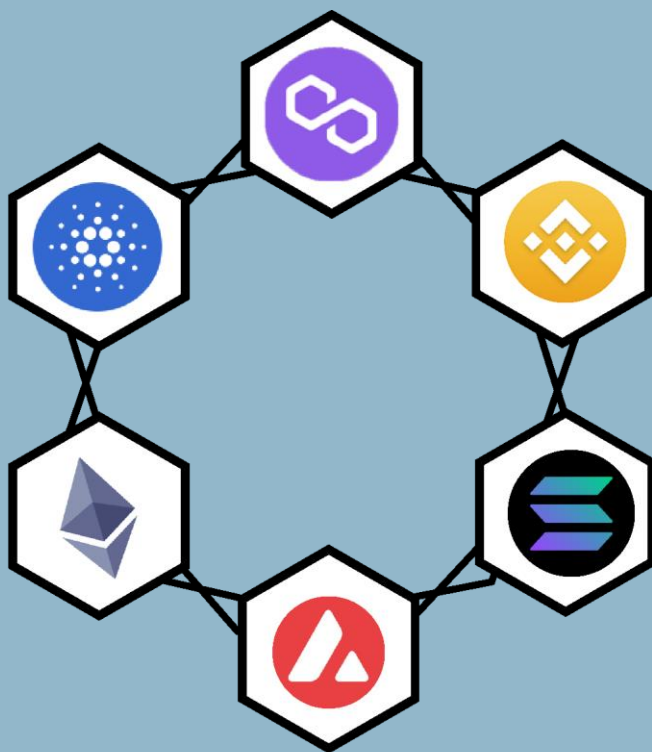


# Team Overview



## KYC NOT CERTIFIED

Audits.finance has not completed a KYC for the project. Audits.finance has not verified the identity of any team member(s) with government issued ID and photo evidence to match. This project is anonymous.



# audits.finance

**Contact information:**

Website: [audits.finance](https://audits.finance)

Telegram: [auditsfinancegroup](https://t.me/auditsfinancegroup)

Twitter: [auditsfinance](https://twitter.com/auditsfinance)

Email: [hello@audits.finance](mailto:hello@audits.finance)

