



Building the Futuristic **Blockchain Ecosystem**

SECURITY AUDIT REPORT

Oggy Floki

TOKEN OVERVIEW

Risk Findings

Severity	Found
● High	0
● Medium	0
● Low	0
● Informational	0

Centralization Risks

Owner Privileges	Description
● Can Owner Set Taxes >25% ?	Not Detected
● Owner needs to enable trading ?	Not Detected
● Can Owner Disable Trades ?	Not Detected
● Can Owner Mint ?	Not Detected
● Can Owner Blacklist ?	Not Detected
● Can Owner set Max Wallet amount ?	Not Detected
● Can Owner Set Max TX amount ?	Not Detected

TABLE OF CONTENTS

02	Token Overview	
03	Table of Contents	
04	Overview	
05	Contract Details	
06	Audit Methodology	
07	Vulnerabilities Checklist	
08	Risk Classification	
09	Static Analysis	
11	Testnet Version	
12	Manual Review	
13	About Expelee	
14	Disclaimer	

OVERVIEW

The 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	-
Audit Date	14 November 2023

CONTRACT DETAILS

Token Address: 0xD724627BA Ae352623Efa352Ac42fA6365AE d94a5

Name: Oggy Floki

Symbol: \$OGF

Decimals: 18

Network: Binance smart chain

Token Type: BEP20

Owner: 0xD4545DdBC582e5d80bE8d11e9e6B46871Ea91566

Deployer: 0xea687d0144fd445ba695f96ffe76429163171a3f

Token Supply: 4696900000000000

Checksum: b391b4a737a5cdfd0edf7de92a1b2764

Testnet version:

The tests were performed using the contract deployed on the Binance smart chain Testnet, which can be found at the following address:

<https://testnet.bscscan.com/address/0x1c4E97890b81d9d11d74EaCe15F3F122f0f77b15#code>

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
- Compiler
- Hardhat

VULNERABILITY CHECKS

Design Logic	Passed
Compiler warnings	Passed
Private user data leaks	Passed
Timestamps 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

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 warnings that can remain unfixed.

Informational

Issues on this level are minor details and warnings that can remain unfixed.

STATIC ANALYSIS

```

INFO:Detectors:
OggyFlokiToken.constructor(address,address,address,address,address,address[]) (OggyFloki.sol#833-874) performs a multiplication on the result of a division:
- _mint(liquidityWallet,((_supply * liquidityPercent) / 100) * 10 ** 9) (OggyFloki.sol#848)
OggyFlokiToken.constructor(address,address,address,address,address,address[]) (OggyFloki.sol#833-874) performs a multiplication on the result of a division:
- _mint(presaleWallet,((_supply * presalePercent) / 100) * 10 ** 9) (OggyFloki.sol#849)
OggyFlokiToken.constructor(address,address,address,address,address,address[]) (OggyFloki.sol#833-874) performs a multiplication on the result of a division:
- _mint(burnWallet,((_supply * burnPercent) / 100) * 10 ** 9) (OggyFloki.sol#850)
OggyFlokiToken.constructor(address,address,address,address,address,address[]) (OggyFloki.sol#833-874) performs a multiplication on the result of a division:
- _mint(cexListingWallet,((_supply * cexListingPercent) / 100) * 10 ** 9) (OggyFloki.sol#851)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#divide-before-multiply
INFO:Detectors:
Contract locking ether found:
  Contract OggyFlokiToken (OggyFloki.sol#789-1029) has payable functions:
    - OggyFlokiToken.receive() (OggyFloki.sol#1026)
  But does not have a function to withdraw the ether
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#contracts-that-lock-ether
INFO:Detectors:
OggyFlokiToken.swapTokens(uint256,address,address) (OggyFloki.sol#989-1013) ignores return value by router.swapExactTokensForTokens(tokenAmount,0,path,addres
ss(this),block.timestamp) (OggyFloki.sol#1001-1007)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return
INFO:Detectors:
OggyFlokiToken.transfer(address,uint256).owner (OggyFloki.sol#937) shadows:
- Ownable.owner() (OggyFloki.sol#184-186) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing
INFO:Detectors:
OggyFlokiToken.constructor(address,address,address,address,address,address,address[])__liquidityWallet (OggyFloki.sol#834) lacks a zero-check on :
- liquidityWallet = _liquidityWallet (OggyFloki.sol#840)
OggyFlokiToken.constructor(address,address,address,address,address,address,address[])__presaleWallet (OggyFloki.sol#835) lacks a zero-check on :
- presaleWallet = _presaleWallet (OggyFloki.sol#841)
OggyFlokiToken.constructor(address,address,address,address,address,address,address[])__burnWallet (OggyFloki.sol#836) lacks a zero-check on :
- burnWallet = _burnWallet (OggyFloki.sol#842)
OggyFlokiToken.constructor(address,address,address,address,address,address,address[])__cexListingWallet (OggyFloki.sol#837) lacks a zero-check on :
- cexListingWallet = _cexListingWallet (OggyFloki.sol#843)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation
INFO:Detectors:

```

```

INFO:Detectors:
OggyFlokiToken.constructor(address,address,address,address,address,address,address[])__liquidityWallet (OggyFloki.sol#834) lacks a zero-check on :
- liquidityWallet = _liquidityWallet (OggyFloki.sol#840)
OggyFlokiToken.constructor(address,address,address,address,address,address,address[])__presaleWallet (OggyFloki.sol#835) lacks a zero-check on :
- presaleWallet = _presaleWallet (OggyFloki.sol#841)
OggyFlokiToken.constructor(address,address,address,address,address,address,address[])__burnWallet (OggyFloki.sol#836) lacks a zero-check on :
- burnWallet = _burnWallet (OggyFloki.sol#842)
OggyFlokiToken.constructor(address,address,address,address,address,address,address[])__cexListingWallet (OggyFloki.sol#837) lacks a zero-check on :
- cexListingWallet = _cexListingWallet (OggyFloki.sol#843)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation
INFO:Detectors:
Reentrancy in OggyFlokiToken.swapTokens(uint256,address,address) (OggyFloki.sol#989-1013):
  External calls:
  - router.swapExactTokensForTokens(tokenAmount,0,path,address(this),block.timestamp) (OggyFloki.sol#1001-1007)
  State variables written after the call(s):
  - super._transfer(msg.sender,_to,tokenAmount) (OggyFloki.sol#1010)
  - _balances[from] = fromBalance - amount (OggyFloki.sol#568)
  - _balances[to] += amount (OggyFloki.sol#571)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2
INFO:Detectors:
Reentrancy in OggyFlokiToken.swapTokens(uint256,address,address) (OggyFloki.sol#989-1013):
  External calls:
  - router.swapExactTokensForTokens(tokenAmount,0,path,address(this),block.timestamp) (OggyFloki.sol#1001-1007)
  Event emitted after the call(s):
  - Swapped(_token,tokenAmount,_to) (OggyFloki.sol#1012)
  - Transfer(from,to,amount) (OggyFloki.sol#574)
  - super._transfer(msg.sender,_to,tokenAmount) (OggyFloki.sol#1010)
Reentrancy in OggyFlokiToken.withdrawBEP20(address,uint256) (OggyFloki.sol#1015-1023):
  External calls:
  - IERC20(_token).transfer(owner(),amount) (OggyFloki.sol#1020)
  Event emitted after the call(s):
  - WithdrawnBEP20(_token,amount) (OggyFloki.sol#1022)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3
INFO:Detectors:
Different versions of Solidity are used:
- Version used: ['0.8.19', '^0.8.0', '^0.8.19']
- 0.8.19 (OggyFloki.sol#747)
- ^0.8.0 (OggyFloki.sol#12)
- ^0.8.0 (OggyFloki.sol#39)

```

STATIC ANALYSIS

```
INFO:Detectors:
OggyFlokiToken.constructor(address,address,address,address,address,address[]) (OggyFloki.sol#833-874) uses literals with too many digits:
- _supply = 4696900000000000 (OggyFloki.sol#845)
OggyFlokiToken.constructor(address,address,address,address,address,address[]) (OggyFloki.sol#833-874) uses literals with too many digits:
- maxSwapAmount = 100000 * 10 ** 9 (OggyFloki.sol#855)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits
INFO:Detectors:
Loop condition i < investors.length (OggyFloki.sol#967) should use cached array length instead of referencing 'length' member of the storage array.
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#cache-array-length
INFO:Detectors:
OggyFlokiToken._busd (OggyFloki.sol#826) should be constant
OggyFlokiToken._usdt (OggyFloki.sol#827) should be constant
OggyFlokiToken.burnPercent (OggyFloki.sol#814) should be constant
OggyFlokiToken.cexListingPercent (OggyFloki.sol#815) should be constant
OggyFlokiToken.liquidityPercent (OggyFloki.sol#812) should be constant
OggyFlokiToken.presalePercent (OggyFloki.sol#813) should be constant
OggyFlokiToken.uniswapV2Pair (OggyFloki.sol#825) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant
INFO:Detectors:
OggyFlokiToken._busdpair (OggyFloki.sol#828) should be immutable
OggyFlokiToken._usdtpair (OggyFloki.sol#829) should be immutable
OggyFlokiToken.burnWallet (OggyFloki.sol#801) should be immutable
OggyFlokiToken.cexListingWallet (OggyFloki.sol#802) should be immutable
OggyFlokiToken.liquidityWallet (OggyFloki.sol#799) should be immutable
OggyFlokiToken.presaleWallet (OggyFloki.sol#800) should be immutable
OggyFlokiToken.router (OggyFloki.sol#824) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
INFO:Slither:OggyFloki.sol analyzed (10 contracts with 93 detectors), 56 result(s) found
```

TESTNET VERSION

1- Set Max Swap Amount (**passed**):

<https://testnet.bscscan.com/tx/0xa4103487b63ce1f0f614ff21e570fca5e1bdabdd79534a8031c2386529fc2be3>

2- Set Max Wallet Amount (**passed**):

<https://testnet.bscscan.com/tx/0x490582e2a11a88d7865bf7efe756befe3522ad059198ddfb951e391cc813e051>

3- Burn (**passed**):

<https://testnet.bscscan.com/tx/0xc5f4cc3931ec10ecb35f28b3ef63cbcebd7418ed1c9506b6b97058abb8f6d162>

4- Approve (**passed**):

<https://testnet.bscscan.com/tx/0xc256e5d327066da579e294110ed5c5859106609be5edc04047282b53b2e1c17e>

5- Transfer Ownership (**passed**):

<https://testnet.bscscan.com/tx/0xd3bb053f8ad3ca54eb396ce257e8b7ce079b04988ace86bf0a1bc7aaba8a81c3>

MANUAL REVIEW

Severity Criteria

Expelee assesses the severity of disclosed vulnerabilities according to methodology based on OWASP standarts.

Vulnerabilities are dividend into three primary risk categroies:

High

Medium

Low

High-level considerations for vulnerabilities span the following key areas when conducting assessments:

- Malicious input handling
- Escalation of privileges
- Arithmetic
- Gas use

Overall Risk Severity				
Impact	HIGH	Medium	High	Critical
	MEDIUM	Low	Medium	High
	LOW	Note	Low	Medium
		LOW	MEDIUM	HIGH
	Likelihood			

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](https://twitter.com/expeleeofficial)

 [expelee](https://medium.com/expelee)

 [Expelee](https://t.me/Expelee)

 [expelee](https://in.linkedin.com/company/expelee)

 [expelee_official](https://www.instagram.com/expelee_official)

 [expelee-co](https://github.com/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 guarantess against the sale of team tokens or the removal of liquidity by the project audited in this document.

Always do your own research and project 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. Alway 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.

The logo for 'expelee' is displayed in a stylized font. The 'ex' is in white, and 'pelee' is in orange. The 'e' after 'ex' is white, and the 'ee' is orange. The logo is centered on a dark blue background with faint, abstract geometric shapes.

Building the Futuristic **Blockchain Ecosystem**