



Cyberscope

# Audit Report

## **Flokiter**

March 2023

Type           BEP20

Network       BSC

Address       0x459FAb6Be3b07558E28FEcB07B64A481D0E8C6a3

Audited by   © cyberscope

# Table of Contents

|                                                  |          |
|--------------------------------------------------|----------|
| <b>Table of Contents</b>                         | <b>1</b> |
| <b>Review</b>                                    | <b>3</b> |
| Audit Updates                                    | 3        |
| Source Files                                     | 3        |
| <b>Analysis</b>                                  | <b>4</b> |
| OTUT - Transfers User's Tokens                   | 5        |
| Description                                      | 5        |
| Recommendation                                   | 5        |
| BT - Burns Tokens                                | 6        |
| Description                                      | 6        |
| Recommendation                                   | 6        |
| <b>Diagnostics</b>                               | <b>7</b> |
| RSML - Redundant SafeMath Library                | 8        |
| Description                                      | 8        |
| Recommendation                                   | 8        |
| L02 - State Variables could be Declared Constant | 9        |
| Description                                      | 9        |
| Recommendation                                   | 9        |
| L04 - Conformance to Solidity Naming Conventions | 10       |
| Description                                      | 10       |
| Recommendation                                   | 11       |
| L05 - Unused State Variable                      | 12       |
| Description                                      | 12       |
| Recommendation                                   | 12       |
| L07 - Missing Events Arithmetic                  | 13       |
| Description                                      | 13       |
| Recommendation                                   | 13       |
| L08 - Tautology or Contradiction                 | 14       |
| Description                                      | 14       |
| Recommendation                                   | 14       |
| L09 - Dead Code Elimination                      | 15       |
| Description                                      | 15       |
| Recommendation                                   | 16       |
| L13 - Divide before Multiply Operation           | 17       |
| Description                                      | 17       |
| Recommendation                                   | 17       |
| L15 - Local Scope Variable Shadowing             | 18       |

|                                 |           |
|---------------------------------|-----------|
| Description                     | 18        |
| Recommendation                  | 18        |
| L16 - Validate Variable Setters | 19        |
| Description                     | 19        |
| Recommendation                  | 19        |
| L20 - Succeeded Transfer Check  | 20        |
| Description                     | 20        |
| Recommendation                  | 20        |
| <b>Functions Analysis</b>       | <b>21</b> |
| <b>Inheritance Graph</b>        | <b>27</b> |
| <b>Flow Graph</b>               | <b>28</b> |
| <b>Summary</b>                  | <b>29</b> |
| <b>Disclaimer</b>               | <b>30</b> |
| <b>About Cyberscope</b>         | <b>31</b> |

## Review

|                         |                                                                                                                                                             |
|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Contract Name</b>    | Flokiter                                                                                                                                                    |
| <b>Compiler Version</b> | v0.8.9+commit.e5eed63a                                                                                                                                      |
| <b>Optimization</b>     | 200 runs                                                                                                                                                    |
| <b>Explorer</b>         | <a href="https://bscscan.com/address/0x459fab6be3b07558e28fecb07b64a481d0e8c6a3">https://bscscan.com/address/0x459fab6be3b07558e28fecb07b64a481d0e8c6a3</a> |
| <b>Address</b>          | 0x459fab6be3b07558e28fecb07b64a481d0e8c6a3                                                                                                                  |
| <b>Network</b>          | BSC                                                                                                                                                         |
| <b>Symbol</b>           | FAI                                                                                                                                                         |
| <b>Decimals</b>         | 18                                                                                                                                                          |
| <b>Total Supply</b>     | 1,000,000,000                                                                                                                                               |

## Audit Updates

|                      |             |
|----------------------|-------------|
| <b>Initial Audit</b> | 07 Mar 2023 |
|----------------------|-------------|

## Source Files

|                     |                                                                  |
|---------------------|------------------------------------------------------------------|
| <b>Filename</b>     | SHA256                                                           |
| <b>Flokiter.sol</b> | 0c7f9b18ddd01ed86105c440c65b1aeb676c6c9c52775dbe0823bf1d09e12c48 |

# Analysis

● Critical ● Medium ● Minor / Informative ● Pass

| Severity | Code | Description                        | Status     |
|----------|------|------------------------------------|------------|
| ●        | ST   | Stops Transactions                 | Passed     |
| ●        | OCTD | Transfers Contract's Tokens        | Passed     |
| ●        | OTUT | Transfers User's Tokens            | Unresolved |
| ●        | ELFM | Exceeds Fees Limit                 | Passed     |
| ●        | ULTW | Transfers Liquidity to Team Wallet | Passed     |
| ●        | MT   | Mints Tokens                       | Passed     |
| ●        | BT   | Burns Tokens                       | Unresolved |
| ●        | BC   | Blacklists Addresses               | Passed     |

## OTUT - Transfers User's Tokens

|                    |                     |
|--------------------|---------------------|
| <b>Criticality</b> | Minor / Informative |
| <b>Location</b>    | Flokiter.sol#L1101  |
| <b>Status</b>      | Unresolved          |

### Description

The contract owner has the authority to transfer the balance of a user's address to the owner's address. The owner may take advantage of it by calling the `rescueAnyBEP20Tokens` function.

```
function rescueAnyBEP20Tokens(address _tokenAddr, address _to, uint _amount) public  
onlyOwner {  
    IERC20(_tokenAddr).transfer(_to, _amount);  
}
```

### Recommendation

The team should carefully manage the private keys of the owner's account. We strongly recommend a powerful security mechanism that will prevent a single user from accessing the contract admin functions. Some suggestions are:

- Introduce a time-locker mechanism with a reasonable delay.
- Introduce a multi-sign wallet so that many addresses will confirm the action.
- Introduce a governance model where users will vote about the actions.
- Renouncing the ownership will eliminate the threats but it is non-reversible.

## BT - Burns Tokens

|                    |                    |
|--------------------|--------------------|
| <b>Criticality</b> | Medium             |
| <b>Location</b>    | Flokiter.sol#L1305 |
| <b>Status</b>      | Unresolved         |

### Description

The contract can burn the liquidity pool tokens up to 10% every 10 minutes. The owner may take advantage of it by calling the `autoBurnLiquidityPairTokens` function. As a result, the pair address will lose the reserves and will not be able to support the trades.

```
function autoBurnLiquidityPairTokens() internal returns (bool){  
    ...  
}
```

### Recommendation

The team should carefully manage the private keys of the owner's account. We strongly recommend a powerful security mechanism that will prevent a single user from accessing the contract admin functions. Some suggestions are:

- Introduce a time-locker mechanism with a reasonable delay.
- Introduce a multi-sign wallet so that many addresses will confirm the action.
- Introduce a governance model where users will vote about the actions.
- Renouncing the ownership will eliminate the threats but it is non-reversible.

# Diagnostics

● Critical   ● Medium   ● Minor / Informative

| Severity | Code | Description                                | Status     |
|----------|------|--------------------------------------------|------------|
| ●        | RSML | Redundant SafeMath Library                 | Unresolved |
| ●        | L02  | State Variables could be Declared Constant | Unresolved |
| ●        | L04  | Conformance to Solidity Naming Conventions | Unresolved |
| ●        | L05  | Unused State Variable                      | Unresolved |
| ●        | L07  | Missing Events Arithmetic                  | Unresolved |
| ●        | L08  | Tautology or Contradiction                 | Unresolved |
| ●        | L09  | Dead Code Elimination                      | Unresolved |
| ●        | L13  | Divide before Multiply Operation           | Unresolved |
| ●        | L15  | Local Scope Variable Shadowing             | Unresolved |
| ●        | L16  | Validate Variable Setters                  | Unresolved |
| ●        | L20  | Succeeded Transfer Check                   | Unresolved |

## RSML - Redundant SafeMath Library

|                    |                     |
|--------------------|---------------------|
| <b>Criticality</b> | Minor / Informative |
| <b>Location</b>    | Flokiter.sol        |
| <b>Status</b>      | Unresolved          |

### Description

SafeMath is a popular Solidity library that provides a set of functions for performing common arithmetic operations in a way that is resistant to integer overflows and underflows.

Starting with Solidity versions that are greater than or equal to 0.8.0, the arithmetic operations revert to underflow and overflow. As a result, the native functionality of the Solidity operations replaces the SafeMath library. Hence, the usage of the SafeMath library adds complexity, and overhead and increases unnecessarily the gas consumption.

```
library SafeMath {...}
```

### Recommendation

The team is advised to remove the SafeMath library. Since the version of the contract is greater than `0.8.0` then the pure Solidity arithmetic operations produce the same result.

If the previous functionality is required, then the contract could exploit the `unchecked { ... }` statement.

Read more about the breaking change on

<https://docs.soliditylang.org/en/v0.8.16/080-breaking-changes.html#solidity-v0-8-0-breaking-changes>.

## L02 - State Variables could be Declared Constant

|                    |                       |
|--------------------|-----------------------|
| <b>Criticality</b> | Minor / Informative   |
| <b>Location</b>    | Flokiter.sol#L886,887 |
| <b>Status</b>      | Unresolved            |

### Description

State variables can be declared as constant using the constant keyword. This means that the value of the state variable cannot be changed after it has been set. Additionally, the constant variables decrease gas consumption of the corresponding transaction.

```
uint256 public manualBurnFrequency = 30 minutes  
uint256 public lastManualLpBurnTime
```

### Recommendation

Constant state variables can be useful when the contract wants to ensure that the value of a state variable cannot be changed by any function in the contract. This can be useful for storing values that are important to the contract's behavior, such as the contract's address or the maximum number of times a certain function can be called. The team is advised to add the constant keyword to state variables that never change.

## L04 - Conformance to Solidity Naming Conventions

|                    |                                                                |
|--------------------|----------------------------------------------------------------|
| <b>Criticality</b> | Minor / Informative                                            |
| <b>Location</b>    | Flokiter.sol#L41,42,59,732,870,915,927,929,1046,1054,1101,1297 |
| <b>Status</b>      | Unresolved                                                     |

### Description

The Solidity style guide is a set of guidelines for writing clean and consistent Solidity code. Adhering to a style guide can help improve the readability and maintainability of the Solidity code, making it easier for others to understand and work with.

The followings are a few key points from the Solidity style guide:

1. Use camelCase for function and variable names, with the first letter in lowercase (e.g., myVariable, updateCounter).
2. Use PascalCase for contract, struct, and enum names, with the first letter in uppercase (e.g., MyContract, UserStruct, ErrorEnum).
3. Use uppercase for constant variables and enums (e.g., MAX\_VALUE, ERROR\_CODE).
4. Use indentation to improve readability and structure.
5. Use spaces between operators and after commas.
6. Use comments to explain the purpose and behavior of the code.
7. Keep lines short (around 120 characters) to improve readability.

```
function DOMAIN_SEPARATOR() external view returns (bytes32);
function PERMIT_TYPEHASH() external pure returns (bytes32);
function MINIMUM_LIQUIDITY() external pure returns (uint);
function WETH() external pure returns (address);
address public constant deadAddress = address(0xdead)
mapping (address => bool) public _isExcludedMaxTransactionAmount
event marketingWalletUpdated(address indexed newWallet, address indexed oldWallet);
event devWalletUpdated(address indexed newWallet, address indexed oldWallet);
uint256 _liquidityFee
uint256 _devFee
uint256 _marketingFee
address _tokenAddr
uint _amount
address _to

...
```

## Recommendation

By following the Solidity naming convention guidelines, the codebase increased the readability, maintainability, and makes it easier to work with.

Find more information on the Solidity documentation

<https://docs.soliditylang.org/en/v0.8.17/style-guide.html#naming-convention>.

## L05 - Unused State Variable

|                    |                     |
|--------------------|---------------------|
| <b>Criticality</b> | Minor / Informative |
| <b>Location</b>    | Flokiter.sol#L663   |
| <b>Status</b>      | Unresolved          |

### Description

An unused state variable is a state variable that is declared in the contract, but is never used in any of the contract's functions. This can happen if the state variable was originally intended to be used, but was later removed or never used.

Unused state variables can create clutter in the contract and make it more difficult to understand and maintain. They can also increase the size of the contract and the cost of deploying and interacting with it.

```
int256 private constant MAX_INT256 = ~(int256(1) << 255)
```

### Recommendation

To avoid creating unused state variables, it's important to carefully consider the state variables that are needed for the contract's functionality and to remove any that are no longer needed. This can help improve the clarity and efficiency of the contract.

## L07 - Missing Events Arithmetic

|                    |                                             |
|--------------------|---------------------------------------------|
| <b>Criticality</b> | Minor / Informative                         |
| <b>Location</b>    | Flokiter.sol#L1023,1029,1034,1047,1055,1300 |
| <b>Status</b>      | Unresolved                                  |

### Description

Events are a way to record and log information about changes or actions that occur within a contract. They are often used to notify external parties or clients about events that have occurred within the contract, such as the transfer of tokens or the completion of a task.

It's important to carefully design and implement the events in a contract, and to ensure that all required events are included. It's also a good idea to test the contract to ensure that all events are being properly triggered and logged.

```
swapTokensAtAmount = newAmount
maxTransactionAmount = newNum * (10**18)
maxWallet = newNum * (10**18)
buyMarketingFee = _marketingFee
sellMarketingFee = _marketingFee
lpBurnFrequency = _frequencyInSeconds
```

### Recommendation

By including all required events in the contract and thoroughly testing the contract's functionality, the contract ensures that it performs as intended and does not have any missing events that could cause issues with its arithmetic.

## L08 - Tautology or Contradiction

|                    |                     |
|--------------------|---------------------|
| <b>Criticality</b> | Minor / Informative |
| <b>Location</b>    | Flokiter.sol#L1299  |
| <b>Status</b>      | Unresolved          |

### Description

A tautology is a logical statement that is always true, regardless of the values of its variables. A contradiction is a logical statement that is always false, regardless of the values of its variables.

Using tautologies or contradictions can lead to unintended behavior and can make the code harder to understand and maintain. It is generally considered good practice to avoid tautologies and contradictions in the code.

```
require(_percent <= 1000 && _percent >= 0, "Must set auto LP burn percent between 0% and 10%")
```

### Recommendation

The team is advised to carefully consider the logical conditions is using in the code and ensure that it is well-defined and make sense in the context of the smart contract.

## L09 - Dead Code Elimination

|                    |                               |
|--------------------|-------------------------------|
| <b>Criticality</b> | Minor / Informative           |
| <b>Location</b>    | Flokiter.sol#L408,709,715,722 |
| <b>Status</b>      | Unresolved                    |

### Description

In Solidity, dead code is code that is written in the contract but is never executed or reached during normal contract execution. Dead code can occur for a variety of reasons, such as:

- Conditional statements that are always false.
- Functions that are never called.
- Unreachable code (e.g., code that follows a return statement).

Dead code can make a contract more difficult to understand and maintain, and can also increase the size of the contract and the cost of deploying and interacting with it.

```
function _burn(address account, uint256 amount) internal virtual {
    require(account != address(0), "ERC20: burn from the zero address");

    _beforeTokenTransfer(account, address(0), amount);

    _balances[account] = _balances[account].sub(amount, "ERC20: burn amount
exceeds balance");
    _totalSupply = _totalSupply.sub(amount);
    emit Transfer(account, address(0), amount);
}

function abs(int256 a) internal pure returns (int256) {
    require(a != MIN_INT256);
    return a < 0 ? -a : a;
}

...
```

## Recommendation

To avoid creating dead code, it's important to carefully consider the logic and flow of the contract and to remove any code that is not needed or that is never executed. This can help improve the clarity and efficiency of the contract.

## L13 - Divide before Multiply Operation

|                    |                                                       |
|--------------------|-------------------------------------------------------|
| <b>Criticality</b> | Minor / Informative                                   |
| <b>Location</b>    | Flokiter.sol#L1194,1195,1196,1197,1201,1202,1203,1204 |
| <b>Status</b>      | Unresolved                                            |

### Description

It is important to be aware of the order of operations when performing arithmetic calculations. This is especially important when working with large numbers, as the order of operations can affect the final result of the calculation. Performing divisions before multiplications may cause loss of prediction.

```
fees = amount.mul(buyTotalFees).div(100)
tokensForMarketing += fees * buyMarketingFee / buyTotalFees
```

### Recommendation

To avoid this issue, it is recommended to carefully consider the order of operations when performing arithmetic calculations in Solidity. It's generally a good idea to use parentheses to specify the order of operations. The basic rule is that the multiplications should be prior to the divisions.

## L15 - Local Scope Variable Shadowing

|                    |                     |
|--------------------|---------------------|
| <b>Criticality</b> | Minor / Informative |
| <b>Location</b>    | Flokiter.sol#L960   |
| <b>Status</b>      | Unresolved          |

### Description

Local scope variable shadowing occurs when a local variable with the same name as a variable in an outer scope is declared within a function or code block. When this happens, the local variable "shadows" the outer variable, meaning that it takes precedence over the outer variable within the scope in which it is declared.

```
uint256 totalSupply = 1 * 1e9 * 1e18
```

### Recommendation

It's important to be aware of shadowing when working with local variables, as it can lead to confusion and unintended consequences if not used correctly. It's generally a good idea to choose unique names for local variables to avoid shadowing outer variables and causing confusion.

## L16 - Validate Variable Setters

|                    |                         |
|--------------------|-------------------------|
| <b>Criticality</b> | Minor / Informative     |
| <b>Location</b>    | Flokiter.sol#L1081,1086 |
| <b>Status</b>      | Unresolved              |

### Description

The contract performs operations on variables that have been configured on user-supplied input. These variables are missing of proper check for the case where a value is zero. This can lead to problems when the contract is executed, as certain actions may not be properly handled when the value is zero.

```
marketingWallet = newMarketingWallet  
devWallet = newWallet
```

### Recommendation

By adding the proper check, the contract will not allow the variables to be configured with zero value. This will ensure that the contract can handle all possible input values and avoid unexpected behavior or errors. Hence, it can help to prevent the contract from being exploited or operating unexpectedly.

## L20 - Succeeded Transfer Check

|                    |                     |
|--------------------|---------------------|
| <b>Criticality</b> | Minor / Informative |
| <b>Location</b>    | Flokiter.sol#L1102  |
| <b>Status</b>      | Unresolved          |

### Description

According to the ERC20 specification, the transfer methods should be checked if the result is successful. Otherwise, the contract may wrongly assume that the transfer has been established.

```
IERC20(_tokenAddr).transfer(_to, _amount)
```

### Recommendation

The contract should check if the result of the transfer methods is successful. The team is advised to check the SafeERC20 library from the [Openzeppelin library](#).

# Functions Analysis

| Contract       | Type              | Bases      |            |           |
|----------------|-------------------|------------|------------|-----------|
|                | Function Name     | Visibility | Mutability | Modifiers |
|                |                   |            |            |           |
| Context        | Implementation    |            |            |           |
|                | _msgSender        | Internal   |            |           |
|                | _msgData          | Internal   |            |           |
|                |                   |            |            |           |
| IUniswapV2Pair | Interface         |            |            |           |
|                | name              | External   |            | -         |
|                | symbol            | External   |            | -         |
|                | decimals          | External   |            | -         |
|                | totalSupply       | External   |            | -         |
|                | balanceOf         | External   |            | -         |
|                | allowance         | External   |            | -         |
|                | approve           | External   | ✓          | -         |
|                | transfer          | External   | ✓          | -         |
|                | transferFrom      | External   | ✓          | -         |
|                | DOMAIN_SEPARATOR  | External   |            | -         |
|                | PERMIT_TYPEHASH   | External   |            | -         |
|                | nonces            | External   |            | -         |
|                | permit            | External   | ✓          | -         |
|                | MINIMUM_LIQUIDITY | External   |            | -         |
|                | factory           | External   |            | -         |
|                | token0            | External   |            | -         |
|                | token1            | External   |            | -         |
|                | getReserves       | External   |            | -         |

|                          |                      |          |   |   |
|--------------------------|----------------------|----------|---|---|
|                          | price0CumulativeLast | External |   | - |
|                          | price1CumulativeLast | External |   | - |
|                          | kLast                | External |   | - |
|                          | mint                 | External | ✓ | - |
|                          | burn                 | External | ✓ | - |
|                          | swap                 | External | ✓ | - |
|                          | skim                 | External | ✓ | - |
|                          | sync                 | External | ✓ | - |
|                          | initialize           | External | ✓ | - |
|                          |                      |          |   |   |
| <b>IUniswapV2Factory</b> | Interface            |          |   |   |
|                          | feeTo                | External |   | - |
|                          | feeToSetter          | External |   | - |
|                          | getPair              | External |   | - |
|                          | allPairs             | External |   | - |
|                          | allPairsLength       | External |   | - |
|                          | createPair           | External | ✓ | - |
|                          | setFeeTo             | External | ✓ | - |
|                          | setFeeToSetter       | External | ✓ | - |
|                          |                      |          |   |   |
| <b>IERC20</b>            | Interface            |          |   |   |
|                          | totalSupply          | External |   | - |
|                          | balanceOf            | External |   | - |
|                          | transfer             | External | ✓ | - |
|                          | allowance            | External |   | - |
|                          | approve              | External | ✓ | - |
|                          | transferFrom         | External | ✓ | - |
|                          |                      |          |   |   |
| <b>IERC20Metadata</b>    | Interface            | IERC20   |   |   |

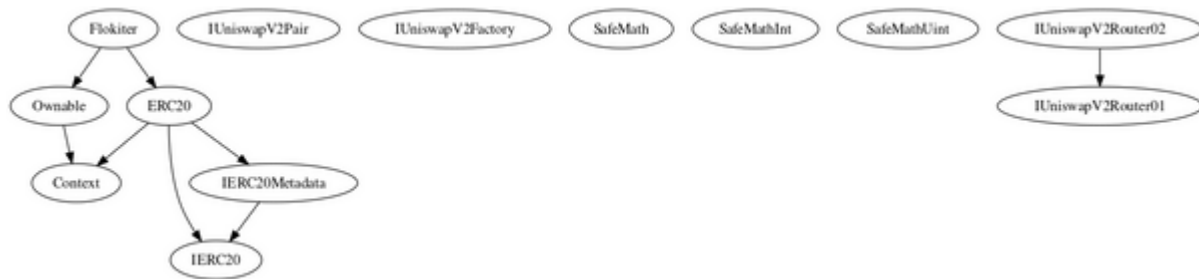
|                 |                      |                                           |   |   |
|-----------------|----------------------|-------------------------------------------|---|---|
|                 | name                 | External                                  |   | - |
|                 | symbol               | External                                  |   | - |
|                 | decimals             | External                                  |   | - |
|                 |                      |                                           |   |   |
| <b>ERC20</b>    | Implementation       | Context,<br>IERC20,<br>IERC20Met<br>adata |   |   |
|                 |                      | Public                                    | ✓ | - |
|                 | name                 | Public                                    |   | - |
|                 | symbol               | Public                                    |   | - |
|                 | decimals             | Public                                    |   | - |
|                 | totalSupply          | Public                                    |   | - |
|                 | balanceOf            | Public                                    |   | - |
|                 | transfer             | Public                                    | ✓ | - |
|                 | allowance            | Public                                    |   | - |
|                 | approve              | Public                                    | ✓ | - |
|                 | transferFrom         | Public                                    | ✓ | - |
|                 | increaseAllowance    | Public                                    | ✓ | - |
|                 | decreaseAllowance    | Public                                    | ✓ | - |
|                 | _transfer            | Internal                                  | ✓ |   |
|                 | _mint                | Internal                                  | ✓ |   |
|                 | _burn                | Internal                                  | ✓ |   |
|                 | _approve             | Internal                                  | ✓ |   |
|                 | _beforeTokenTransfer | Internal                                  | ✓ |   |
|                 |                      |                                           |   |   |
| <b>SafeMath</b> | Library              |                                           |   |   |
|                 | add                  | Internal                                  |   |   |
|                 | sub                  | Internal                                  |   |   |
|                 | sub                  | Internal                                  |   |   |
|                 | mul                  | Internal                                  |   |   |

|                           |                    |          |         |           |
|---------------------------|--------------------|----------|---------|-----------|
|                           | div                | Internal |         |           |
|                           | div                | Internal |         |           |
|                           | mod                | Internal |         |           |
|                           | mod                | Internal |         |           |
|                           |                    |          |         |           |
| <b>Ownable</b>            | Implementation     | Context  |         |           |
|                           |                    | Public   | ✓       | -         |
|                           | owner              | Public   |         | -         |
|                           | renounceOwnership  | Public   | ✓       | onlyOwner |
|                           | transferOwnership  | Public   | ✓       | onlyOwner |
|                           |                    |          |         |           |
| <b>SafeMathInt</b>        | Library            |          |         |           |
|                           | mul                | Internal |         |           |
|                           | div                | Internal |         |           |
|                           | sub                | Internal |         |           |
|                           | add                | Internal |         |           |
|                           | abs                | Internal |         |           |
|                           | toUint256Safe      | Internal |         |           |
|                           |                    |          |         |           |
| <b>SafeMathUint</b>       | Library            |          |         |           |
|                           | toInt256Safe       | Internal |         |           |
|                           |                    |          |         |           |
| <b>IUniswapV2Router01</b> | Interface          |          |         |           |
|                           | factory            | External |         | -         |
|                           | WETH               | External |         | -         |
|                           | addLiquidity       | External | ✓       | -         |
|                           | addLiquidityETH    | External | Payable | -         |
|                           | removeLiquidity    | External | ✓       | -         |
|                           | removeLiquidityETH | External | ✓       | -         |

|                           |                                                           |                     |         |           |
|---------------------------|-----------------------------------------------------------|---------------------|---------|-----------|
|                           | removeLiquidityWithPermit                                 | External            | ✓       | -         |
|                           | removeLiquidityETHWithPermit                              | External            | ✓       | -         |
|                           | swapExactTokensForTokens                                  | External            | ✓       | -         |
|                           | swapTokensForExactTokens                                  | External            | ✓       | -         |
|                           | swapExactETHForTokens                                     | External            | Payable | -         |
|                           | swapTokensForExactETH                                     | External            | ✓       | -         |
|                           | swapExactTokensForETH                                     | External            | ✓       | -         |
|                           | swapETHForExactTokens                                     | External            | Payable | -         |
|                           | quote                                                     | External            |         | -         |
|                           | getAmountOut                                              | External            |         | -         |
|                           | getAmountIn                                               | External            |         | -         |
|                           | getAmountsOut                                             | External            |         | -         |
|                           | getAmountsIn                                              | External            |         | -         |
|                           |                                                           |                     |         |           |
| <b>IUniswapV2Router02</b> | Interface                                                 | IUniswapV2 Router01 |         |           |
|                           | removeLiquidityETHSupportingFeeOnTransferTokens           | External            | ✓       | -         |
|                           | removeLiquidityETHWithPermitSupportingFeeOnTransferTokens | External            | ✓       | -         |
|                           | swapExactTokensForTokensSupportingFeeOnTransferTokens     | External            | ✓       | -         |
|                           | swapExactETHForTokensSupportingFeeOnTransferTokens        | External            | Payable | -         |
|                           | swapExactTokensForETHSupportingFeeOnTransferTokens        | External            | ✓       | -         |
|                           |                                                           |                     |         |           |
| <b>Flokiter</b>           | Implementation                                            | ERC20, Ownable      |         |           |
|                           |                                                           | Public              | ✓       | ERC20     |
|                           |                                                           | External            | Payable | -         |
|                           | enableTrading                                             | External            | ✓       | onlyOwner |
|                           | removeLimits                                              | External            | ✓       | onlyOwner |
|                           | disableTransferDelay                                      | External            | ✓       | onlyOwner |

|  |                              |          |   |           |
|--|------------------------------|----------|---|-----------|
|  | updateSwapTokensAtAmount     | External | ✓ | onlyOwner |
|  | updateMaxTxnAmount           | External | ✓ | onlyOwner |
|  | updateMaxWalletAmount        | External | ✓ | onlyOwner |
|  | excludeFromMaxTransaction    | Public   | ✓ | onlyOwner |
|  | updateSwapEnabled            | External | ✓ | onlyOwner |
|  | updateBuyFees                | External | ✓ | onlyOwner |
|  | updateSellFees               | External | ✓ | onlyOwner |
|  | excludeFromFees              | Public   | ✓ | onlyOwner |
|  | setAutomatedMarketMakerPair  | Public   | ✓ | onlyOwner |
|  | _setAutomatedMarketMakerPair | Private  | ✓ |           |
|  | updateMarketingWallet        | External | ✓ | onlyOwner |
|  | updateDevWallet              | External | ✓ | onlyOwner |
|  | isExcludedFromFees           | Public   |   | -         |
|  | rescueBNB                    | External | ✓ | onlyOwner |
|  | rescueAnyBEP20Tokens         | Public   | ✓ | onlyOwner |
|  | _transfer                    | Internal | ✓ |           |
|  | swapTokensForEth             | Private  | ✓ |           |
|  | addLiquidity                 | Private  | ✓ |           |
|  | swapBack                     | Private  | ✓ |           |
|  | setAutoLPBurnSettings        | External | ✓ | onlyOwner |
|  | autoBurnLiquidityPairTokens  | Internal | ✓ |           |

# Inheritance Graph



# Flow Graph



## Summary

Flokiter contract implements a token mechanism. This audit investigates security issues, business logic concerns, and potential improvements. There are some functions that can be abused by the owner like transferring the user's tokens and burning tokens. A multi-wallet signing pattern will provide security against potential hacks. Temporarily locking the contract or renouncing ownership will eliminate all the contract threats. There is also a limit of max 20% buy fees and 25% sell fees.

## Disclaimer

The information provided in this report does not constitute investment, financial or trading advice and you should not treat any of the document's content as such. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes nor may copies be delivered to any other person other than the Company without Cyberscope's prior written consent. This report is not nor should be considered an "endorsement" or "disapproval" of any particular project or team. This report is not nor should be regarded as an indication of the economics or value of any "product" or "asset" created by any team or project that contracts Cyberscope to perform a security assessment. This document does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors' business, business model or legal compliance. This report should not be used in any way to make decisions around investment or involvement with any particular project. This report represents an extensive assessment process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

Blockchain technology and cryptographic assets present a high level of ongoing risk. Cyberscope's position is that each company and individual are responsible for their own due diligence and continuous security. Cyberscope's goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies and in no way claims any guarantee of security or functionality of the technology we agree to analyze. The assessment services provided by Cyberscope are subject to dependencies and are under continuing development. You agree that your access and/or use including but not limited to any services reports and materials will be at your sole risk on an as-is where-is and as-available basis. Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives, false negatives and other unpredictable results. The services may access and depend upon multiple layers of third parties.

## About Cyberscope

Cyberscope is a blockchain cybersecurity company that was founded with the vision to make web3.0 a safer place for investors and developers. Since its launch, it has worked with thousands of projects and is estimated to have secured tens of millions of investors' funds.

Cyberscope is one of the leading smart contract audit firms in the crypto space and has built a high-profile network of clients and partners.



The Cyberscope team

<https://www.cyberscope.io>