



SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT



Flokinomics
\$FLOKS

20/02/2023

TOKEN OVERVIEW

Fees

- Buy fees: 5%
- Sell fees: 5%

Fees privileges

- Can change buy fees up to 99% and sell fees up to 99%

Ownership

- Owned

Minting

- No mint function

Max Tx Amount / Max Wallet Amount

- Can change / set max tx amount or max wallet amount (with threshold)

Blacklist

- Blacklist function not detected

Other privileges

- Can exclude / include from fees
-

TABLE OF CONTENTS

- 1 DISCLAIMER
- 2 INTRODUCTION
- 3-4 AUDIT OVERVIEW
- 5-8 OWNER PRIVILEGES
- 9 CONCLUSION AND ANALYSIS
- 10 TOKEN DETAILS
- 11 FLOKS TOKEN ANALYTICS & TOP 10 TOKEN HOLDERS
- 12 TECHNICAL DISCLAIMER



DISCLAIMER

The information provided on this analysis document is only for general information and should not be used as a reason to invest.

FreshCoins Team will take no payment for manipulating the results of this audit.

The score and the result will stay on this project page information on our website <https://freshcoins.io>

FreshCoins Team does not guarantees that a project will not sell off team supply, or any other scam strategy (RUG or Honeypot etc)



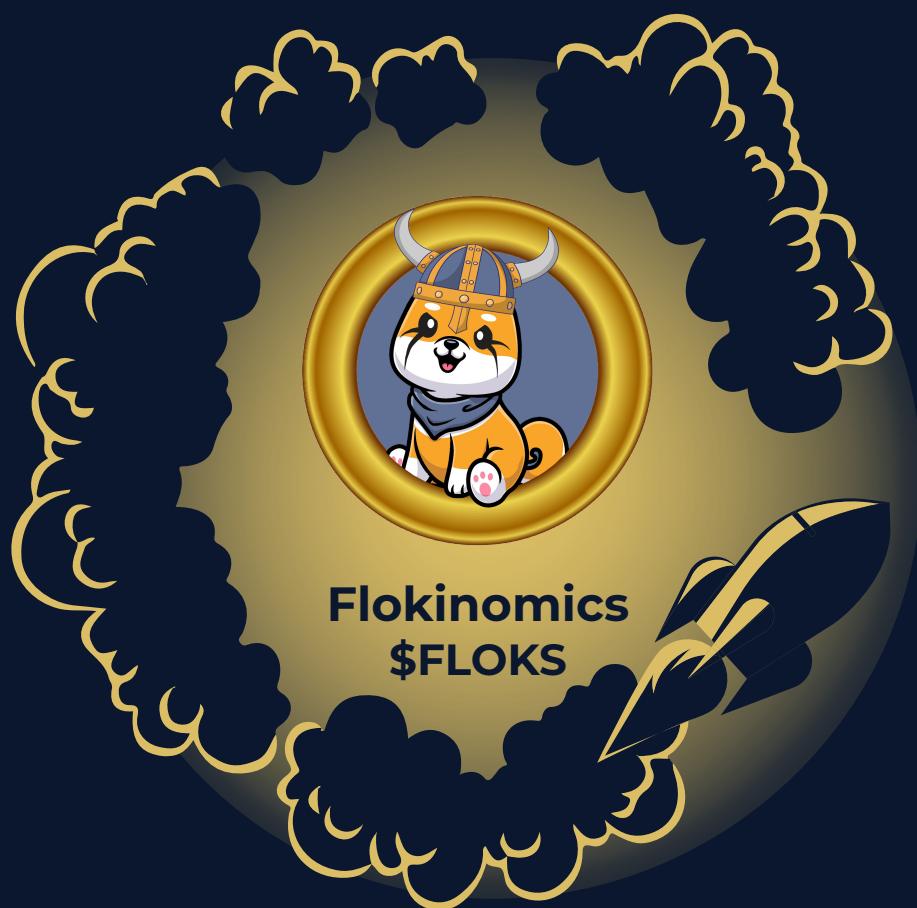
INTRODUCTION

FreshCoins (Consultant) was contracted by Flokinomics (Customer) to conduct a Smart Contract Code Review and Security Analysis.

0xA97D0ed34E6385D135bAdE415De4B0b3D542E629

Network: Binance Smart Chain (BSC)

This report presents the findings of the security assessment of Customer's smart contract and its code review conducted on 20/02/2023



AUDIT OVERVIEW



Security Score



Static Scan
Automatic scanning for common vulnerabilities



ERC Scan
Automatic checks for ERC's conformance

- 1 High
- 3 Medium
- 0 Low
- 0 Optimizations
- 0 Informational



No.	Issue description	Checking Status
1	Compiler Errors / Warnings	Passed
2	Reentrancy and Cross-function	Passed
3	Front running	Passed
4	Timestamp dependence	Passed
5	Integer Overflow and Underflow	Passed
6	Reverted DoS	Passed
7	DoS with block gas limit	Passed
8	Methods execution permissions	Passed
9	Exchange rate impact	Passed
10	Malicious Event	Passed
11	Scoping and Declarations	Passed
12	Uninitialized storage pointers	Passed
13	Design Logic	Passed
14	Safe Zeppelin module	Passed

OWNER PRIVILEGES

- Contract owner can't mint tokens after initial contract deploy
- Contract owner can't exclude an address from transactions
- Contract owner can exclude/include wallet from tax

```
function excludeFromFees(address account, bool excluded) public onlyOwner {  
    _isExcludedFromFees[account] = excluded;  
    emit ExcludeFromFees(account, excluded);  
}
```

- Contract owner can exclude/include wallet from tx limitations

```
function excludeFromMaxTransaction(address updAds, bool isEx) public onlyOwner {  
    _isExcludedMaxTransactionAmount[updAds] = isEx;  
}
```

- Contract owner can change swap settings

```
function updateSwapEnabled(bool enabled) external onlyOwner {  
    swapEnabled = enabled;  
}  
  
function updateSwapTokensAtAmount(uint256 newAmount)  
external  
onlyOwner  
returns (bool)  
{  
    require(  
        newAmount >= (totalSupply() * 1) / 100000,  
        "Swap amount cannot be lower than 0.001% total supply."  
    );  
    require(  
        newAmount <= (totalSupply() * 5) / 1000,  
        "Swap amount cannot be higher than 0.5% total supply."  
    );  
    swapTokensAtAmount = newAmount;  
    return true;  
}
```

- Contract owner has to call `enableTrading` function to enable trade

```
function enableTrading() external onlyOwner {  
    tradingActive = true;  
    swapEnabled = true;  
    lastLpBurnTime = block.timestamp;  
}
```

● Contract owner can change buy fees up to 99% and sell fees up to 99%

```
function updateBuyFees(
    uint256 _marketingFee,
    uint256 _liquidityFee,
    uint256 _devFee
) external onlyOwner {
    buyMarketingFee = _marketingFee;
    buyLiquidityFee = _liquidityFee;
    buyDevFee = _devFee;
    buyTotalFees = buyMarketingFee + buyLiquidityFee + buyDevFee;
    require(buyTotalFees <= 99, "Must keep fees at 99% or less");
}

function updateSellFees(
    uint256 _marketingFee,
    uint256 _liquidityFee,
    uint256 _devFee
) external onlyOwner {
    sellMarketingFee = _marketingFee;
    sellLiquidityFee = _liquidityFee;
    sellDevFee = _devFee;
    sellTotalFees = sellMarketingFee + sellLiquidityFee + sellDevFee;
    require(sellTotalFees <= 99, "Must keep fees at 99% or less");
}
```

● Contract owner can remove all limits (transfers delay, tx limitations, wallet limitations)

```
function removeLimits() external onlyOwner returns (bool) {
    limitsInEffect = false;
    return true;
}
```

● Contract owner can remove delay between trades limitation

Only one purchase per block allowed

```
function disableTransferDelay() external onlyOwner returns (bool) {
    transferDelayEnabled = false;
    return true;
}
```

● Contract owner can change max wallet amount (with threshold)

```
function updateMaxWalletAmount(uint256 newNum) external onlyOwner {
    require(
        newNum >= ((totalSupply() * 5) / 1000) / 1e18,
        "Cannot set maxWallet lower than 0.5%"
    );
    maxWallet = newNum * (10**18);
}
```

● Contract owner can change max tx amount (with threshold)

```
function updateMaxTxnAmount(uint256 newNum) external onlyOwner {
    require(
        newNum >= ((totalSupply() * 1) / 1000) / 1e18,
        "Cannot set maxTransactionAmount lower than 0.1%"
    );
    maxTransactionAmount = newNum * (10**18);
}
```

● Contract owner can change marketingWallet and devWallet addresses

Current values:

marketingWallet : 0xf3d0a4453f8d5169d5ba61b5f26e35bd14112720

devWallet : 0x89ef34b7bdd91d63907cffa08430f42d9a01dcbe

```
function updateMarketingWallet(address newMarketingWallet)
    external
    onlyOwner
{
    emit marketingWalletUpdated(newMarketingWallet, marketingWallet);
    marketingWallet = newMarketingWallet;
}

function updateDevWallet(address newWallet) external onlyOwner {
    emit devWalletUpdated(newWallet, devWallet);
    devWallet = newWallet;
}
```

● Contract owner can transfer ownership

```
function transferOwnership(address newOwner) public virtual onlyOwner {
    require(newOwner != address(0), "Ownable: new owner is the zero address");
    _transferOwnership(newOwner);
}

function _transferOwnership(address newOwner) internal virtual {
    address oldOwner = _owner;
    _owner = newOwner;
    emit OwnershipTransferred(oldOwner, newOwner);
}
```

● Contract owner can renounce ownership

```
function renounceOwnership() public virtual onlyOwner {
    _transferOwnership(address(0));
}
```

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. The risk can be prevented by temporarily locking the contract or renouncing ownership.



CONCLUSION AND ANALYSIS



Smart Contracts within the scope were manually reviewed and analyzed with static tools.



Audit report overview contains all found security vulnerabilities and other issues in the reviewed code.



Found 1 HIGH issues during the first review.

TOKEN DETAILS

Details

Buy fees:	5%
Sell fees:	5%
Max TX:	20,000,000,000,000
Max Sell:	N/A

Honeypot Risk

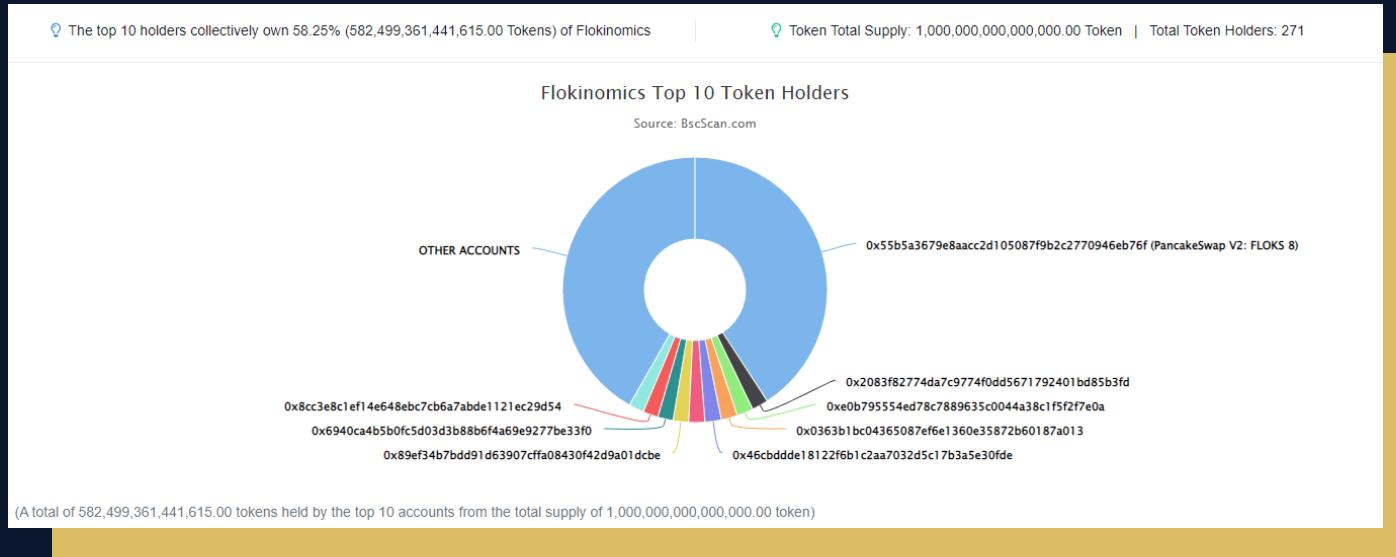
Ownership:	Owned
Blacklist:	Not detected
Modify Max TX:	Detected
Modify Max Sell:	Not detected
Disable Trading:	Not detected

Rug Pull Risk

Liquidity:	N/A
Holders:	Clean



FLOKS TOKEN ANALYTICS & TOP 10 TOKEN HOLDERS



Rank	Address	Quantity (Token)	Percentage
1	PancakeSwap V2: FLOKS 8	408,174,966,074,798,166,455,713,686,634,397	40.8175%
2	0x2083f82774da7c9774f0dd5671792401bd85b3fd	19,999,959,020,788,844,428,647,230,016,024	2.0000%
3	0xe0b795554ed78c7889635c0044a38c1f5f2f7e0a	19,999,925,522,004,790,623,611,413,899,59	2.0000%
4	0x0363b1bc04365087ef6e1360e35872b60187a013	19,941,730,250,003,800,164,293,130,217,064	1.9942%
5	0x46cbddde18122f6b1c2aa7032d5c17b3a5e30fde	19,941,395,283,112,272,196,021,709,966,209	1.9941%
6	0x939fa5dbf2b69248d7087cea2af5958a19a94bc7	19,659,684,408,819,773,513,764,045,970,793	1.9660%
7	0x89ef34b7bdd91d63907cffa08430f42d9a01dcbe	18,915,026,675,433,019,034,812,359,961,327	1.8915%
8	0x6940ca4b5b0fc5d03d3b88b6f4a69e9277be33f0	18,725,557,620,488,313,081,167,628,423,891	1.8726%
9	0x8cc3e8c1ef14e648ebc7cb6a7abde1121ec29d54	18,698,207,993,003,035,668,463,625,521,495	1.8698%
10	0x96e68cbf4ab9afd123b8c04508bf1a77f76c8d7a	18,442,908,593,162,890,844,214,676,10797	1.8443%

TECHNICAL DISCLAIMER

Smart contracts are deployed and executed on the blockchain platform. The platform, its programming language, and other software related to the smart contract can have its vulnerabilities that can lead to hacks. The audit can't guarantee the explicit security of the audited project / smart contract.

