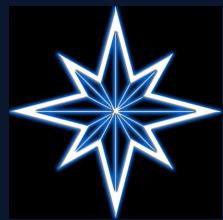




SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT



Meta Sirius
\$SIRI

22/01/2022

TABLE OF CONTENTS

- 1 DISCLAIMER
- 2 INTRODUCTION
- 3-4 WEBSITE DIAGNOSTIC
- 5-6 AUDIT OVERVIEW
- 7-8 OWNER PRIVILEGES
- 9 CONCLUSION AND ANALYSIS
- 10 TOKEN DETAILS
- 11 META SIRIUS TOKEN DISTRIBUTION & 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
Meta Sirius (Customer) to conduct a Smart Contract Code Review
and Security Analysis.**

0x04fCC89bD5a78e7C6d56037F9B4F778563763BaF

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 **22/01/2022**



WEBSITE DIAGNOSTIC

<https://metasirius.finance/>



0-49



50-89



90-100



Performance



Accessibility



Best Practices



SEO



Progressive
Web App

Metrics



First Contentful Paint

0.8 s



Time to interactive

0.8 s



Speed Index

1.5 s



Total Blocking Time

0 ms



Large Contentful Paint

2.4 s



Cumulative Layout Shift

0.008

WEBSITE IMPROVEMENTS

Reduce unused JavaScript

Ensure text remains visible during webfont load

Document does not have a meta description

AUDIT OVERVIEW



Security Score



Static Scan
Automatic scanning for common vulnerabilities



ERC Scan
Automatic checks for ERC's conformance



High



Medium



Low



Optimizations



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 exclude/include wallet from fees

```
function excludeFromFee(address account) public onlyOwner {
    _isExcludedFromFee[account] = true;
}

function includeInFee(address account) public onlyOwner {
    _isExcludedFromFee[account] = false;
}
```

Contract owner can exclude/include wallet from reward

```
function excludeFromReward(address account) public onlyOwner {
    // require(account != 0x7a250d5630B4cF539739dF2C5dAcb4c659F2488D, 'We can not exclude Uniswap router');
    require(!_isExcluded[account], "Account is already excluded");
    if (_rOwned[account] > 0) {
        _tOwned[account] = tokenFromReflection(_rOwned[account]);
    }
    _isExcluded[account] = true;
    _excluded.push(account);
}

function includeInReward(address account) external onlyOwner {
    require(_isExcluded[account], "Account is already excluded");
    for (uint256 i = 0; i < _excluded.length; i++) {
        if (_excluded[i] == account) {
            _excluded[i] = _excluded[_excluded.length - 1];
            _tOwned[account] = 0;
            _isExcluded[account] = false;
            _excluded.pop();
            break;
        }
    }
}
```

Contract owner can change swap settings

```
function setSwapAndLiquifyEnabled(bool _enabled) public onlyOwner() {
    swapAndLiquifyEnabled = _enabled;
    emit SwapAndLiquifyEnabledUpdated(_enabled);
}
```

Contract owner can renounce ownership

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

Contract owner can transfer ownership

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

Contract owner can change fees

```
function setTaxFeePercent(uint256 taxFeeBps) external onlyOwner {  
    _taxFee = taxFeeBps;  
    require(  
        _taxFee + _liquidityFee + _charityFee <= 10**4 / 4,  
        "Total fee is over 25%"  
    );  
}  
  
function setLiquidityFeePercent(uint256 liquidityFeeBps)  
    external  
    onlyOwner  
{  
    _liquidityFee = liquidityFeeBps;  
    require(  
        _taxFee + _liquidityFee + _charityFee <= 10**4 / 4,  
        "Total fee is over 25%"  
    );  
}
```



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 no issue during the first review.

TOKEN DETAILS

Details

Buy fees:	9%
Sell fees:	10%
Max TX:	N/A
Max Sell:	N/A

Honeypot Risk

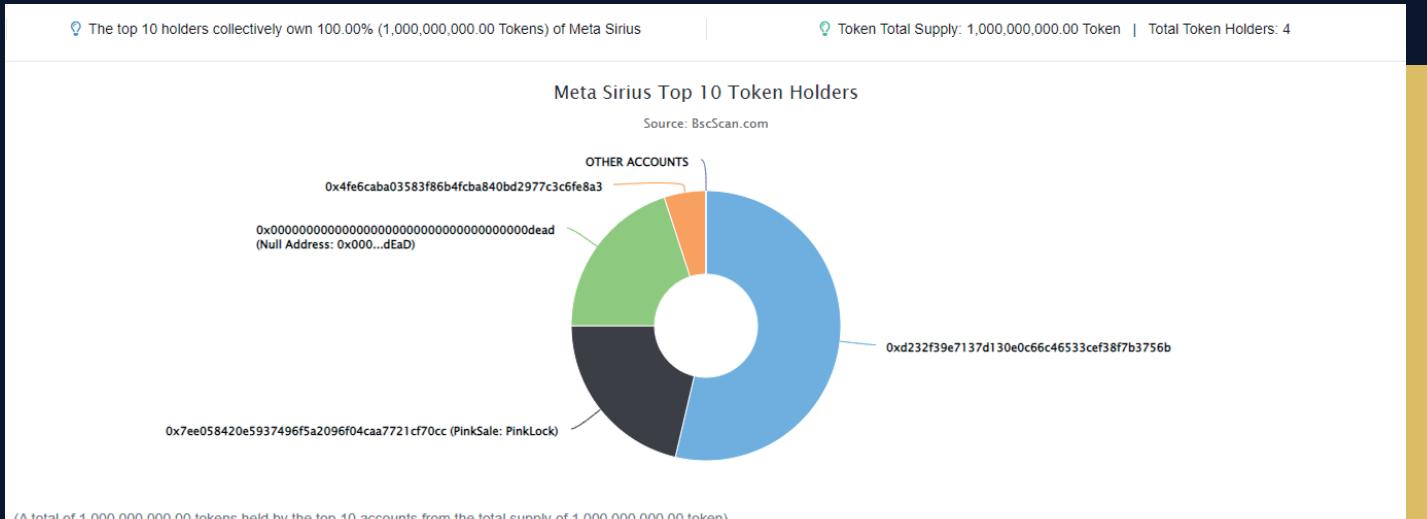
Ownership:	Owned
Blacklist:	Not detected
Modify Max TX:	Not detected
Modify Max Sell:	Not detected
Disable Trading:	Not detected

Rug Pull Risk

Liquidity:	N/A
Holders:	Clean



META SIRIUS TOKEN DISTRIBUTION & TOP 10 TOKEN HOLDERS



Rank	Address	Quantity (Token)	Percentage
1	0xd232f39e7137d130e0c66c46533cef38f7b3756b	536,700,000	53.6700%
2	PinkSale: PinkLock	213,300,000	21.3300%
3	Null Address: 0x000...dEaD	200,000,000	20.0000%
4	0x4fe6caba03583f86b4fcba840bd2977c3c6fe8a3	50,000,000	5.0000%

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.

