



freshcoins

SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT



Meta Space Debris
\$DEBRIS



12/03/2022



TABLE OF CONTENTS

- 1 DISCLAIMER
- 2 INTRODUCTION
- 3-4 AUDIT OVERVIEW
- 5-6 OWNER PRIVILEGES
- 7 CONCLUSION AND ANALYSIS
- 8 TOKEN DETAILS
- 9 META SPACE DEBRIS TOKEN ANALYTICS & TOP 10 TOKEN HOLDERS
- 10 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 Space Debris (Customer) to conduct a Smart Contract Code Review and Security Analysis.

0x426f15297B351f51234B9d7A73f9E2BD87700Ddd

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 12/03/2022



AUDIT OVERVIEW



Security Score



Static Scan
Automatic scanning for common vulnerabilities



ERC Scan
Automatic checks for ERC's conformance

0 **High**

0 **Medium**

1 **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	Low
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 exclude an address from transactions.

Contract owner can't mint tokens after initial contract deploy

Contract owner can exclude/include wallet(s) from tax

```
function excludeFromFees(address account, bool excluded) public onlyOwner {
    require(_isExcludedFromFees[account] != excluded, "DEBRIS: Account is already the value of 'excluded'");
    _isExcludedFromFees[account] = excluded;

    emit ExcludeFromFees(account, excluded);
}

function excludeMultipleAccountsFromFees(address[] calldata accounts, bool excluded) public onlyOwner {
    for(uint256 i = 0; i < accounts.length; i++) {
        _isExcludedFromFees[accounts[i]] = excluded;
    }

    emit ExcludeMultipleAccountsFromFees(accounts, excluded);
}
```

Contract owner can exclude wallet from dividends

```
function excludeFromDividends(address account) external onlyOwner {
    dividendTracker.excludeFromDividends(account);
}
```

Contract owner can exclude/include wallet from tx limitations

```
function setExcludeFromMaxTx(address _address, bool value) public onlyOwner {
    _isExcludedFromMaxTx[_address] = value;
}
```

Contract owner can exclude wallet from tx limitations, fees and dividends

```
function setExcludeFromAll(address _address) public onlyOwner {
    _isExcludedFromMaxTx[_address] = true;
    _isExcludedFromFees[_address] = true;
    dividendTracker.excludeFromDividends(_address);
}
```

Contract owner can burn tokens

```
function burn(uint256 amount) public onlyOwner() {
    _burn(msg.sender, amount);
}
```

Contract owner can change max tx amount

```
function setMaxtx(uint256 _maxSellTxAmount) public onlyOwner {  
    maxSellTransactionAmount = _maxSellTxAmount;  
}
```

Contract owner can change the fees up to 100%

```
function setFee(uint256 _bnbRewardFee, uint256 _liquidityFee, uint256 _marketingFee) public onlyOwner {  
    BNBRewardsFee = _bnbRewardFee;  
    liquidityFee = _liquidityFee;  
    marketingFee = _marketingFee;  
  
    totalFees = BNBRewardsFee.add(liquidityFee).add(marketingFee); // total fee transfer and buy  
}
```

Contract owner can add extra fee on sell

```
function setExtraFeeOnSell(uint256 _extraFeeOnSell) public onlyOwner {  
    extraFeeOnSell = _extraFeeOnSell; // extra fee on sell  
}
```

Contract owner can include wallet in whitelist

```
function includeToWhiteList(address[] memory _users) external onlyOwner {  
    for(uint8 i = 0; i < _users.length; i++) {  
        _whiteList[_users[i]] = true;  
    }  
}
```

Contract owner can change swap settings

```
function setSwapAmont(uint256 _swapTokensAtAmount) public onlyOwner {  
    swapTokensAtAmount = _swapTokensAtAmount;  
}  
  
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);  
}
```

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 LOW issue during the first review.

TOKEN DETAILS

Details

Buy fees:	15%
Sell fees:	15%
Max TX:	2,000,000
Max Sell:	N/A

Honeypot Risk

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

Rug Pull Risk

Liquidity:	N/A
Holders:	Clean



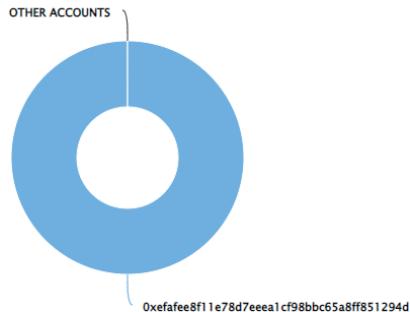
META SPACE DEBRIS TOKEN ANALYTICS & TOP 10 TOKEN HOLDERS

The top 10 holders collectively own 100.00% (1,000,000,000,000.00 Tokens) of Meta Space Debris

Token Total Supply: 1,000,000,000,000.00 Token | Total Token Holders: 1

Meta Space Debris Top 10 Token Holders

Source: BscScan.com



(A total of 1,000,000,000,000.00 tokens held by the top 10 accounts from the total supply of 1,000,000,000,000.00 token)

Rank	Address	Quantity (Token)	Percentage
1	Oxefafee8f11e78d7eeee1cf98bbc65a8ff851294d	1,000,000,000,000	100.0000%

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.

