



SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT



SpaceMeta
\$SPMETA

13/03/2022

TABLE OF CONTENTS

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

0x517626AbeEEd54435BEFFCae5b27312F66E33Df1

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 13/03/2022



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't exclude an address from transactions

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

```
function excludeFromFees(address account, bool excluded) public onlyOwner {
    require(
        _isExcludedFromFees[account] != excluded,
        "BABYTOKEN: 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 {
    require(!excludedFromDividends[account]);
    excludedFromDividends[account] = true;
    _setBalance(account, 0);
    tokenHoldersMap.remove(account);
    emit ExcludeFromDividends(account);
}
```

Contract owner can exclude/include wallet from tx limitations

```
function excludeFromMaxTransferAmount(address account, bool excluded)
    external
    onlyOwner
{
    require(
        _isExcludedFromMaxTransferAmount[account] != excluded,
        "BABYTOKEN: Account is already the value of 'excluded'"
    );
    _isExcludedFromMaxTransferAmount[account] = excluded;
    emit ExcludeFromMaxTransferAmount(account, excluded);
}
```

Contract owner can exclude/include wallet from wallet limitations

```
function excludeFromMaxWalletBalance(address account, bool excluded)
    external
    onlyOwner
{
    require(
        _isExcludedFromMaxWalletBalance[account] != excluded,
        "BABYTOKEN: Account is already the value of 'excluded'"
    );
    _isExcludedFromMaxWalletBalance[account] = excluded;
    emit ExcludeFromMaxWalletBalance(account, excluded);
}
```

Contract owner can change `_marketingWalletAddress` and `_teamWalletAddress` addresses

Current values:

`_marketingWalletAddress` : `0xb38e1e4f74b28018159f0fbea969ea84e3cd70ab`

`_teamWalletAddress` : `0xc480bf019ab71a6dc6984f5eb02c622ae62e2100`

```
function setMarketingWallet(address payable wallet) external onlyOwner {
    _marketingWalletAddress = wallet;
}

function setTeamWallet(address payable wallet) external onlyOwner {
    _teamWalletAddress = wallet;
}
```

Contract owner can change fees up to 25%

```
function setTokenRewardsFee(uint256 value) external onlyOwner {
    tokenRewardsFee = value;
    totalFees = tokenRewardsFee.add(liquidityFee).add(marketingFee).add(teamFee);
    require(totalFees <= 25, "Total fee is over 25%");
}

function setLiquidityFee(uint256 value) external onlyOwner {
    liquidityFee = value;
    totalFees = tokenRewardsFee.add(liquidityFee).add(marketingFee).add(teamFee);
    require(totalFees <= 25, "Total fee is over 25%");
}

function setMarketingFee(uint256 value) external onlyOwner {
    marketingFee = value;
    totalFees = tokenRewardsFee.add(liquidityFee).add(marketingFee).add(teamFee);
    require(totalFees <= 25, "Total fee is over 25%");
}

function setTeamFee(uint256 value) external onlyOwner {
    teamFee = value;
    totalFees = tokenRewardsFee.add(liquidityFee).add(marketingFee).add(teamFee);
    require(totalFees <= 25, "Total fee is over 25%");
}
```

Contract owner can change max tx amount

```
function setMaxTransferAmount(uint256 amount) external onlyOwner {  
    maxTransferAmount = amount;  
}
```

Contract owner can change max wallet amount

```
function setMaxWalletBalance(uint256 amount) external onlyOwner {  
    maxWalletBalance = amount;  
}
```

Contract owner can change swap settings

```
function setSwapTokensAtAmount(uint256 amount) external onlyOwner {  
    swapTokensAtAmount = amount;  
}
```

Contract owner can renounce ownership

```
function renounceOwnership() public virtual onlyOwner {  
    _transferOwnership(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");  
    _transferOwnership(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 no issue during the first review.

TOKEN DETAILS

Details

Buy fees:	10%
Sell fees:	10%
Max TX:	1,000,000,000
Max Sell:	N/A

Honeypot Risk

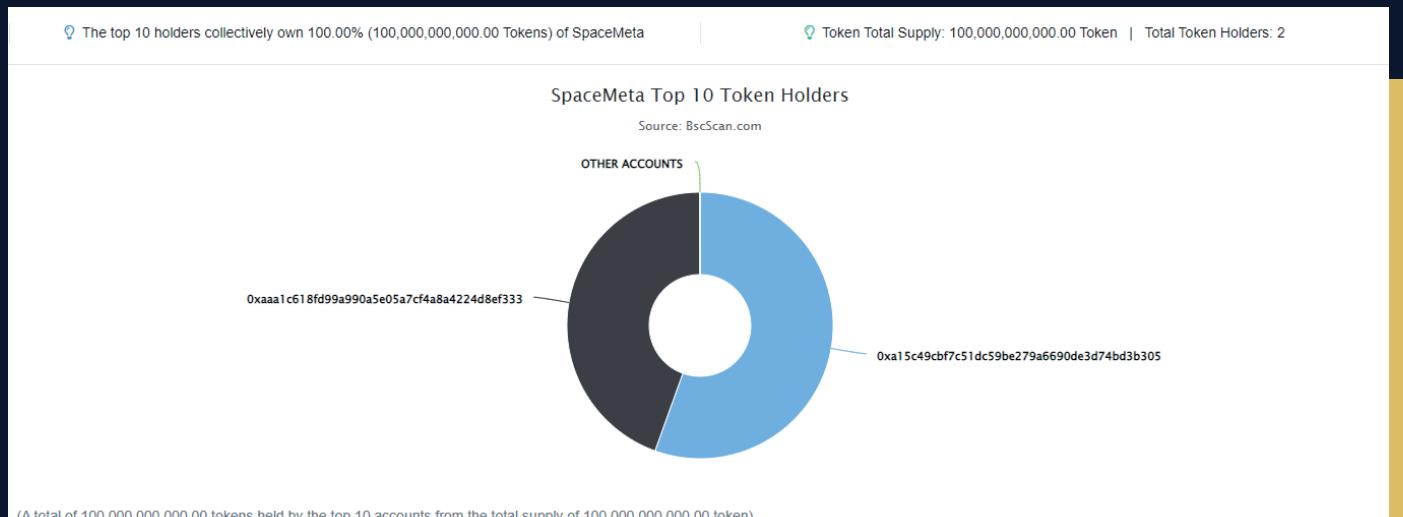
Ownership:	Owned
Blacklist:	Not detected
Modify Max TX:	Detected
Modify Max Sell:	Not detected
Disable Trading:	Not detected

Rug Pull Risk

Liquidity:	N/A
Holders:	Clean



SPACEMETA TOKEN ANALYTICS & TOP 10 TOKEN HOLDERS



Rank	Address	Quantity (Token)	Percentage
1	0xa15c49cbf7c51dc59be279a6690de3d74bd3b305	55,500,000,000	55.5000%
2	0xaa1c618fd99a990a5e05a7cf4a8a4224d8ef333	44,500,000,000	44.5000%

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.

