Audit Report for **DMSCoin**
Date: 27 February 2024
Audit result: **Passed with High Risk**

Token Address: 0x16878aCb7190281feff007A7B38B41DBB5780558

Name: DMScoin

Symbol: DMS

Decimals: 18

Network: BscScan

Token Type: BEP-20

Owner: 0xB7Cd7175B8D423B10640756Ee547AcF666e82A26

Deployer: 0xB7Cd7175B8D423B10640756Ee547AcF666e82A26

Token Supply: 10000000000

Checksum: ke1c3a4fbb6e83e8393a57617b5a5B21

Testnet:
https://testnet.bscscan.com/address/0x49f58721b19A066b8BD6669338D904E9a8Cda81f#code

Token Overview:

Buy Fee: 0-0%

Sell Fee: 0-0%

Transfer Fee: 0-0%

Fee Privilege: Owner

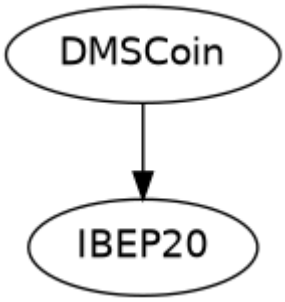
Ownership: Owned

Minting: Yes

Max Tx: No

Blacklist: No

Inheritance Tree



Static Analysis

A static analysis of the code was performed using Slither. No issues were found.

```
INFO:Detectors:
Pragma version^0.8.0 (DMSCoin.sol#6) allows old versions
solc-0.8.0 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Parameter DMSCoin.allowance(address,address)._owner (DMSCoin.sol#56) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
DMSCoin.decimals (DMSCoin.sol#23) should be constant
DMSCoin.name (DMSCoin.sol#21) should be constant
DMSCoin.symbol (DMSCoin.sol#22) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant
INFO:Detectors:
DMSCoin.owner (DMSCoin.sol#28) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
INFO:Slither:DMSCoin.sol analyzed (2 contracts with 93 detectors), 7 result(s) found
```

Functional Tests
Router (PCS V2):

1- Approve (passed):
https://testnet.bscscan.com/tx/0x3e93584343416616b63a38241961ce9d2757759e29a9d71db35803e586abb111

2- Burn (passed):
https://testnet.bscscan.com/tx/0x78c223019323b08e5a71d98319123cea19d4d149b9e79549e80688e477d0be79

3- Mint (passed):
https://testnet.bscscan.com/tx/0xd2af51b0215b764e7e9ff647c53016fabc4338c5ae1d5c6532f468b646540dde

4- Transfer (passed):
https://testnet.bscscan.com/tx/0x0e3ad3c19437bd3c01ae63c976edb103a399e1c917204c1bacc0fe36198558e2

Findings:
Critical: 0
High: 1
Medium: 0
Low: 1
Informational & Optimizations: 1

Centralization – Owner Can Mint Tokens.

Severity: High
Status: Open
Function: mint
Overview:
The owner is able to mint unlimited tokens which is not recommended as this functionality can cause the token to lose it's value and the owner can also use it to manipulate the price of the token.

```
function mint(address account, uint256 amount) public onlyOwner returns (bool) {
        require(account != address(0), "BEP20: mint to the zero address");

        totalSupply += amount;
        balances[account] += amount;
        emit Transfer(address(0), account, amount);
        return true;
}
```

Suggestion:
It is recommended that the total supply of the tokens should not be changed after initial deployment.

Centralization – Missing Visibility

Severity: Low
Subject: Visibility
Status: Open
Overview:
It's simply saying that no visibility was specified, so it's going with the default. This has been related to security issues in contracts.

```
mapping(address => uint256) balances;
mapping(address => mapping(address => uint256)) allowed;
```

Suggestion:
You can easily silence the warning by adding the public/private.

Optimization

Severity: **Informational**
Subject: Floating Pragma.
Status: Open
Overview:
It is considered best practice to pick one compiler version and stick with it. With a floating pragma, contracts may accidentally be deployed using an outdated.

```
pragma solidity ^0.8.0;
```

Suggestion:
Adding the latest constant version of solidity is recommended, as this prevents the unintentional deployment of a contract with an outdated compiler that contains unresolved bugs.