Coinsult

Advanced Manual Smart Contract Audit

Project: BscMeta
Website: http://bscmeta.vip

● **Low-risk**	● **Medium-risk**	● **High-risk**
5 low-risk code issues found	0 medium-risk code issues found	0 high-risk code issues found

Contract address

0x55Fd08f5966aE82bF8c8E6eec2547bDd3252672b

Disclaimer: Coinsult is not responsible for any financial losses. Nothing in this contract audit is financial advice, please do your own research.

Disclaimer

Coinsult is not responsible if a project turns out to be a scam, rug-pull or honeypot. We only provide a detailed analysis for your own research.

Coinsult is not responsible for any financial losses. Nothing in this contract audit is financial advice, please do your own research.

The information provided in this audit is for informational purposes only and should not be considered investment advice. Coinsult does not endorse, recommend, support or suggest to invest in any project.

Coinsult can not be held responsible for when a project turns out to be a
rug-pull, honeypot or scam.

Tokenomics

Total Supply: 1,000,000,000
Total Holders: 1
Top 10 holders:

Rank	Address	Quantity (Token)	Percentage
1	0xa9be4daafea1932feb78a2df7e70c05682be5187	1,000,000,000	100.0000%

The top 100 holders collectively own 100.00% (1,000,000,000 Tokens) of BSCMETA

Note: This is a snapshot of when the audit was performed.

Source code

Coinsult was commissioned by BSCMETA to perform an audit based on the following smart contract:

https://bscscan.com/address/0x55fd08f5966ae82bf8c8e6eec2547bdd3252672b#code

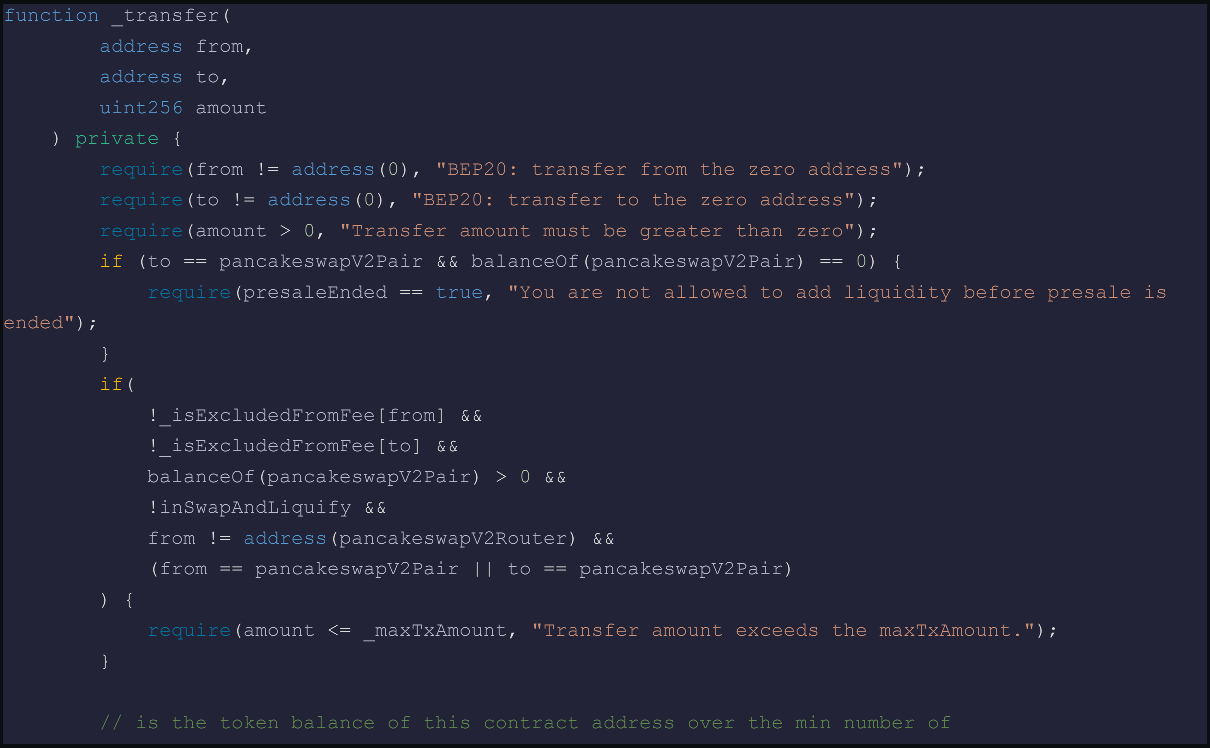
Manual Code Review

● Low-risk

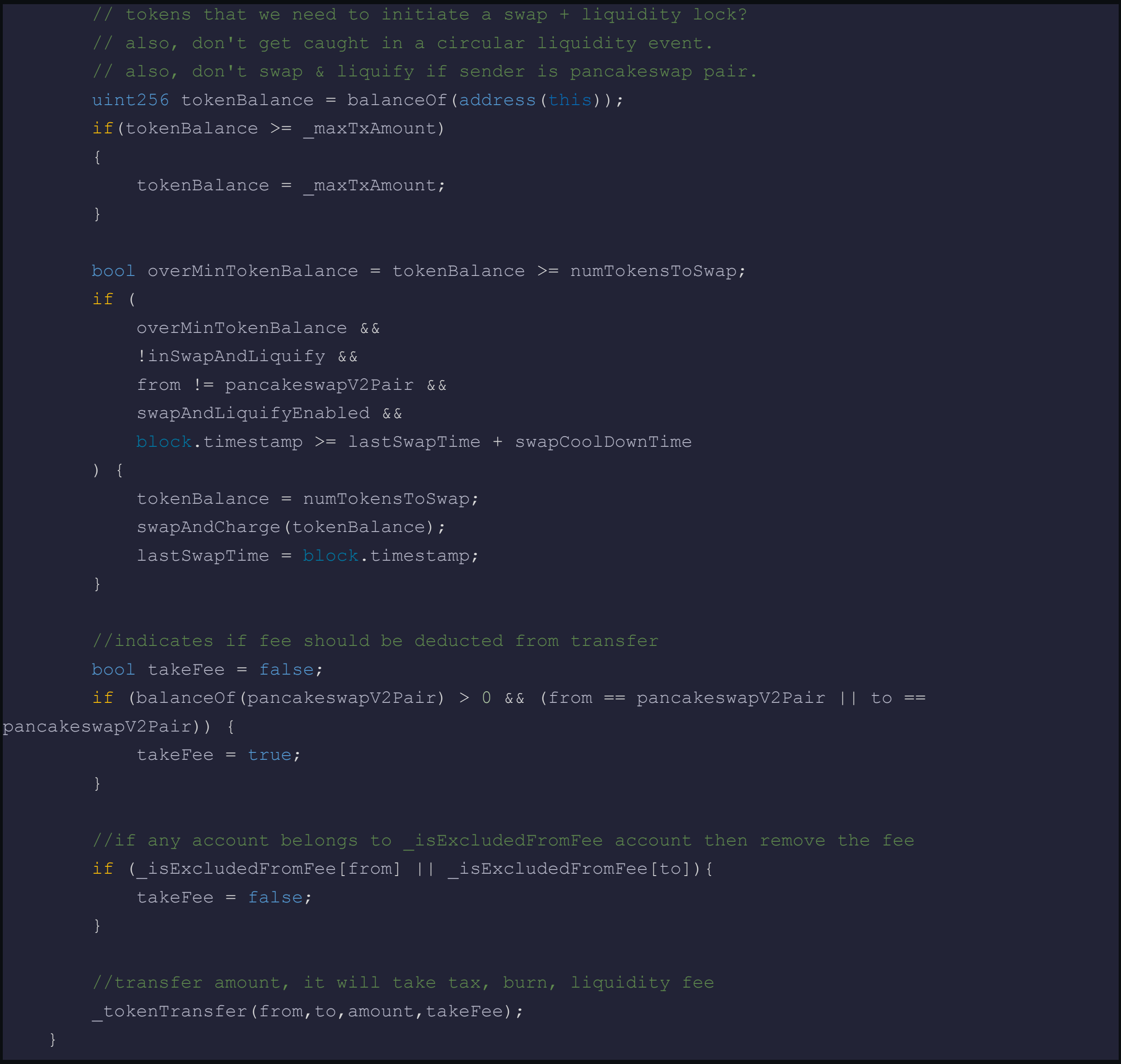
5 low-risk code issues found.
Could be fixed, will not bring problems.

- Contract contains Reentrancy vulnerabilities:
  _transfer(address,address,uint256)

  Additional information: This combination increases risk of malicious intent. While it may be justified by some complex mechanics (e.g. rebase, reflections, buyback). More information: Slither

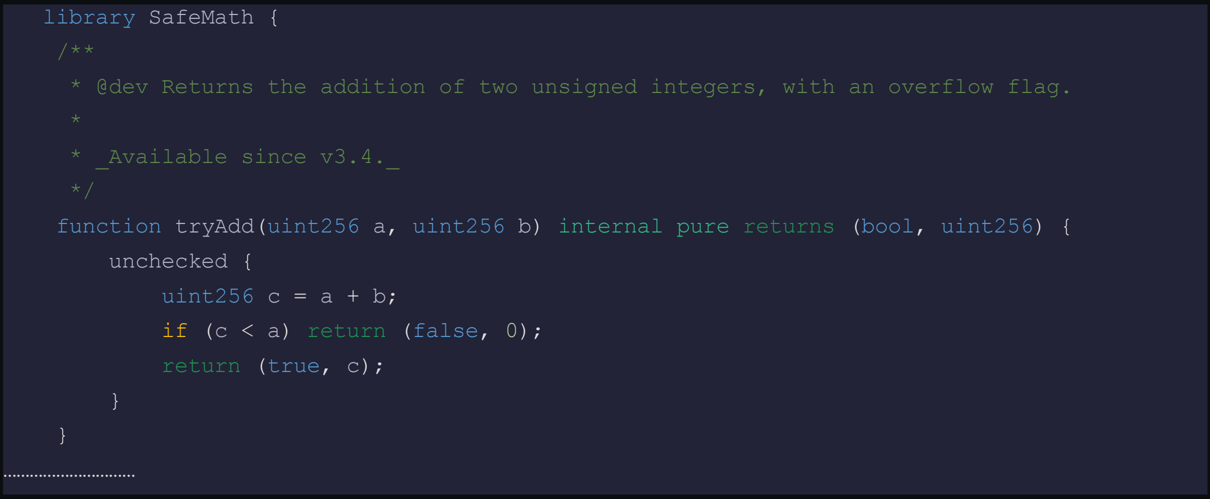


```
function _transfer(
        address from,
        address to,
        uint256 amount
    ) private {
        require(from != address(0), "BEP20: transfer from the zero address");
        require(to != address(0), "BEP20: transfer to the zero address");
        require(amount > 0, "Transfer amount must be greater than zero");
        if (to == pancakeswapV2Pair && balanceOf(pancakeswapV2Pair) == 0) {
            require(presaleEnded == true, "You are not allowed to add liquidity before presale is
ended");
        }
        if(
            !_isExcludedFromFee[from] &&
            !_isExcludedFromFee[to] &&
            balanceOf(pancakeswapV2Pair) > 0 &&
            !inSwapAndLiquify &&
            from != address(pancakeswapV2Router) &&
            (from == pancakeswapV2Pair || to == pancakeswapV2Pair)
        ) {
            require(amount <= _maxTxAmount, "Transfer amount exceeds the maxTxAmount.");
        }

        // is the token balance of this contract address over the min number of
```



```
        // tokens that we need to initiate a swap + liquidity lock?
        // also, don't get caught in a circular liquidity event.
        // also, don't swap & liquify if sender is pancakeswap pair.
        uint256 tokenBalance = balanceOf(address(this));
        if(tokenBalance >= _maxTxAmount)
        {
            tokenBalance = _maxTxAmount;
        }

        bool overMinTokenBalance = tokenBalance >= numTokensToSwap;
        if (
            overMinTokenBalance &&
            !inSwapAndLiquify &&
            from != pancakeswapV2Pair &&
            swapAndLiquifyEnabled &&
            block.timestamp >= lastSwapTime + swapCoolDownTime
        ) {
            tokenBalance = numTokensToSwap;
            swapAndCharge(tokenBalance);
            lastSwapTime = block.timestamp;
        }

        //indicates if fee should be deducted from transfer
        bool takeFee = false;
        if (balanceOf(pancakeswapV2Pair) > 0 && (from == pancakeswapV2Pair || to ==
pancakeswapV2Pair)) {
            takeFee = true;
        }

        //if any account belongs to _isExcludedFromFee account then remove the fee
        if (_isExcludedFromFee[from] || _isExcludedFromFee[to]){
            takeFee = false;
        }

        //transfer amount, it will take tax, burn, liquidity fee
        _tokenTransfer(from,to,amount,takeFee);
    }
```

- `SafeMath` is no longer needed starting with Solidity 0.8. The compiler now has built-in overflow checking. It can be removed.



```
   library SafeMath {
    /**
     * @dev Returns the addition of two unsigned integers, with an overflow flag.
     *
     * _Available since v3.4._
     */
    function tryAdd(uint256 a, uint256 b) internal pure returns (bool, uint256) {
        unchecked {
            uint256 c = a + b;
            if (c < a) return (false, 0);
            return (true, c);
        }
    }
```

- Contract has high starting tax fees

```
    uint256 public _BNBFee = 11;
    uint256 private _previousBNBFee = _BNBFee;

    uint256 public _liquidityFee = 2;
    uint256 private _previousLiquidityFee = _liquidityFee;
```

- Follow the Solidity naming convention consistently. Not everything is in mixedCase.

```
function WETH() external pure returns (address);
```

- Avoid relying on block.timestamp. block.timestamp can be manipulated by miners.

```
  if (
            overMinTokenBalance &&
            !inSwapAndLiquify &&
            from != pancakeswapV2Pair &&
            swapAndLiquifyEnabled &&
            block.timestamp >= lastSwapTime + swapCoolDownTime
        ) {
            tokenBalance = numTokensToSwap;
            swapAndCharge(tokenBalance);
            lastSwapTime = block.timestamp;
        }
```

Potential exploit:
Bob's contract relies on block.timestamp for its randomness. Eve is a miner and manipulates block.timestamp to exploit Bob's contract.

🟡 Medium-risk

0 medium-risk code issues found.

Should be fixed, could bring problems.

🔴 High-risk

0 high-risk code issues found

Must be fixed, and will bring problems.

Extra notes by the team

🟢 There is a lot of commented text within the contract, this will cause an unclear code to be read.

Notes:

🟡 Owner can set max transaction amount

🟡 Owner can whitelist addresses from fee

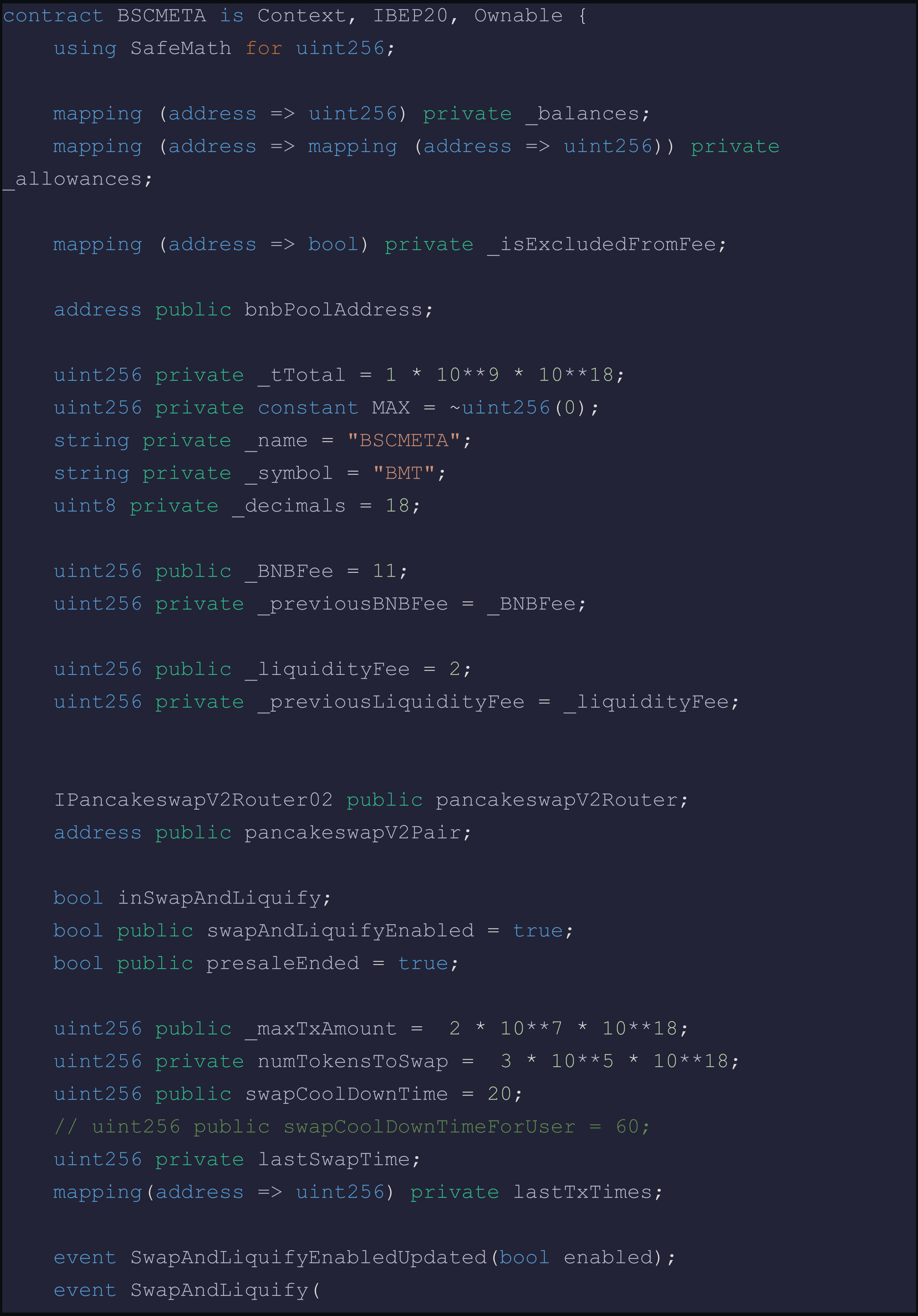
🟡 The ownership of the contract isn’t renounced

🟡 Testnet router address also in code for mainnet useless as it will not be used

🔴 Owner entitled to change the transaction fees up to 100%

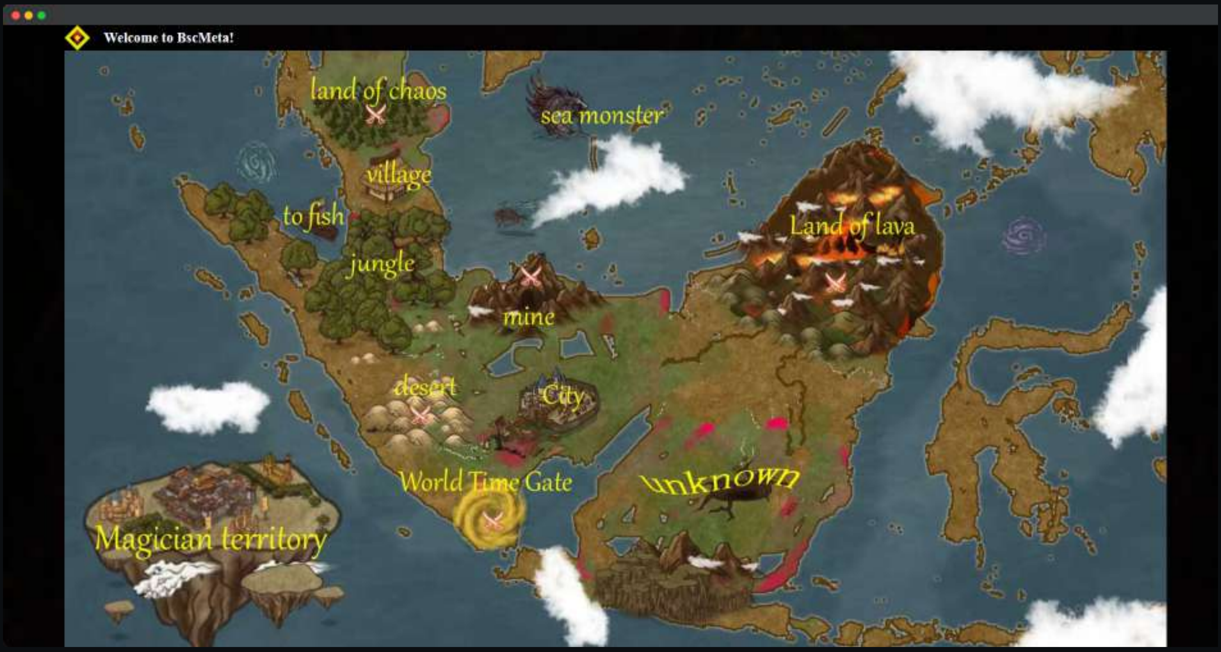
🔴 Contract uses a CoolDownTime function. If the input is very high no swapping can take place for a long amount of time.

Contract Snapshot



```
contract BSCMETA is Context, IBEP20, Ownable {
    using SafeMath for uint256;

    mapping (address => uint256) private _balances;
    mapping (address => mapping (address => uint256)) private
_allowances;

    mapping (address => bool) private _isExcludedFromFee;

    address public bnbPoolAddress;

    uint256 private _tTotal = 1 * 10**9 * 10**18;
    uint256 private constant MAX = ~uint256(0);
    string private _name = "BSCMETA";
    string private _symbol = "BMT";
    uint8 private _decimals = 18;

    uint256 public _BNBFee = 11;
    uint256 private _previousBNBFee = _BNBFee;

    uint256 public _liquidityFee = 2;
    uint256 private _previousLiquidityFee = _liquidityFee;


    IPancakeswapV2Router02 public pancakeswapV2Router;
    address public pancakeswapV2Pair;

    bool inSwapAndLiquify;
    bool public swapAndLiquifyEnabled = true;
    bool public presaleEnded = true;

    uint256 public _maxTxAmount =  2 * 10**7 * 10**18;
    uint256 private numTokensToSwap =  3 * 10**5 * 10**18;
    uint256 public swapCoolDownTime = 20;
    // uint256 public swapCoolDownTimeForUser = 60;
    uint256 private lastSwapTime;
    mapping(address => uint256) private lastTxTimes;

    event SwapAndLiquifyEnabledUpdated(bool enabled);
    event SwapAndLiquify(
```

Website Review



Coinsult checks the website completely manually and looks for visual, technical and textual errors. We also look at the security, speed and accessibility of the website. In short, a complete check to see if the website meets the current standard of the web development industry.

- 🟡 Semi-mobile Friendly
- 🔴 Contains jQuery errors
- 🟢 SSL Secured
- 🟢 Appropriate spelling

Note: The website does not contain a lot of information at the moment of the audit.

Loading speed: 81%

Rug-pull Review

Based on the available information analyzed by us, we come to the following conclusions:

🟢 Locked Liquidity

🔴 Large unlocked wallets (100%)
(This will probably change once liquidity pair is build)

🔴 No Doxxed Team

Honeypot Review

Based on the available information analyzed by us, we come to the following conclusions:

🟡 Ability to sell
(Fees can be set higher than 25% by owner)

🟡 Owner unable to prevent selling
(But, fees can be set higher than 25% by owner)

🟢 Accurate liquidity pair

Note: Please check the disclaimer above and note, the audit makes no statements or warranties on business model, investment attractiveness or code sustainability. The report is provided for the only contract mentioned in the report and does not include any other potential contracts deployed by the project owner.