BlockSAFU

ADVANCE MANUAL SMART CONTRACT AUDIT

Project: MyEducation

Website: https://www.myedtoken.com/

✔ PASSED

BlockSAFU Score:

82

Contract Address:

0x8edc28E352dC48B7d5C7140f8e5EEbe5F335beAB

DISCLAMER

BlockSAFU has completed this report to provide a summary of the Smart Contract functions, and any security, dependency, or cybersecurity vulnerabilities. This is often a constrained report on our discoveries based on our investigation and understanding of the current programming versions as of this report's date. To understand the full scope of our analysis, it is vital for you to at the date of this report. To understand the full scope of our analysis, you need to review the complete report. Although we have done our best in conducting our investigation and creating this report, it is vital to note that you should not depend on this report and cannot make any claim against BlockSAFU or its Subsidiaries and Team members on the premise of what has or has not been included in the report. Please remember to conduct your independent examinations before making any investment choices. We do not provide investment advice or in any way claim to determine if the project will be successful or not.

By perusing this report or any portion of it, you concur to the terms of this disclaimer. In the unlikely situation where you do not concur with the terms, you should immediately terminate reading this report, and erase and discard any duplicates of this report downloaded and/or printed by you. This report is given for data purposes as it were and on a non-reliance premise and does not constitute speculation counsel. No one should have any right to depend on the report or its substance, and BlockSAFU and its members (including holding companies, shareholders, backups, representatives, chiefs, officers, and other agents) BlockSAFU and its subsidiaries owe no obligation of care towards you or any other person, nor does BlockSAFU make any guarantee or representation to any individual on the precision or completeness of the report.

ABOUT THE AUDITOR:

BlockSAFU (BSAFU) is an Anti-Scam Token Utility that reviews Smart Contracts and Token information to Identify Rug Pull and Honey Pot scamming activity. BlockSAFUs Development Team consists of several Smart Contract creators, Auditors Developers, and Blockchain experts. BlockSAFU provides solutions, prevents, and hunts down scammers. BSAFU is a utility token with features Audit, KYC, Token Generators, and Bounty Scammers. It will enrich the crypto ecosystem.

OVERVIEW

Mint Function

- No mint functions.

Fees

- Fee on buy 2% (owner can't set fees over 5%).
- Fee on sell 2% (owner can't set fees over 20%).

Tx Amount

- Owner cannot set a max tx amount.

Transfer Pausable

- Owner can't pause.

Blacklist

- Owner can't blacklist.

Ownership

- Owner can't take back ownership.

Proxy

- This contract has no proxy.

Anti Whale

- Owner can't limit the number of wallet holdings.

Trading Cooldown

- Owner can't set the selling time interval.

SMART CONTRACT REVIEW

Token Name	**Compound Staking**
Contract Address	0x16d93ab88024607b52cC02b0e12D05aAdd0eec13
Deployer Address	0x10D1F91b59B89C3B68ee06fA13CB2957fF005676
Owner Address	0x10d1f91b59b89c3b68ee06fa13cb2957ff005676
on buy	2%
on sell	2%
Gas Used for Buy	Will be updated after listing on dex
Gas Used for Sell	Will be updated after listing on dex
Contract Created	Dec-18-2022 09:25:16 AM +UTC
Initial Liquidity	Will be updated after listing on dex
Liquidity Status	Locked
Unlocked Date	Will be updated after listing on dex
Verified CA	Yes
Compiler	v0.8.17+commit.8df45f5f
Optimization	No with 200 runs
Sol License	MIT License
Other	default evmVersion

TAX

Buy	2%		Sell	2%

Token Holder

Rank	Address	Quantity	Percentage	Analytics
1	Pinksale: PinkLock V2	6,075,000,000	60.7500%	
2	0x0974c34b362d975782a28a393a848c930b30b30c	3,925,000,000	39.2500%	

Team Review

The Coma team has a nice website, their website is professionally built and the Smart contract is well developed, their social media is growing with over 593 people in their telegram group (count in audit date).

Official Website And Social Media

Website: https://www.myedtoken.com/

Telegram Group: https://t.me/MyedOffical

Twitter: https://twitter.com/MyedOfficial

MANUAL CODE REVIEW

🟢 Minor-risk

1 minor-risk code issue found

1. Use safeERC20 from transer checking

```solidity
function transferFrom(
    address sender,
    address recipient,
    uint256 amount
) public virtual override returns (bool) {
    uint256 currentAllowance = _allowances[sender][_msgSender()];
    if (currentAllowance != type(uint256).max) {
        require(currentAllowance >= amount, "ERC20: transfer amount exceeds allowance");
        unchecked {
            _approve(sender, _msgSender(), currentAllowance - amount);
        }
    }

    _transfer(sender, recipient, amount);

    return true;
}
```

🟡 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 problem.

● Critical-Risk

0 critical-risk code issues found

Must be fixed, and will bring problem.

EXTRA NOTES SMART CONTRACT

1. IERC20

```solidity
interface IERC20 {
  /**
   * @dev Returns the number of tokens in existence.
   */
  function totalSupply() external view returns (uint256);
…
  function balanceOf(address account) external view returns (uint256);
…
  function transfer(address recipient, uint256 amount) external returns (bool);
…
  function allowance(address owner, address spender) external view returns (uint256);
…
  function approve(address spender, uint256 amount) external returns (bool);
…
  function transferFrom(
    address sender,
    address recipient,
    uint256 amount
  ) external returns (bool);

  /**
   * @dev Emitted when `value` tokens are moved from one account (`from`) to
   * another (`to`).
   *
   * Note that `value` may be zero.
   */
  event Transfer(address indexed from, address indexed to, uint256 value);
…
}
```

IERC20 Normal Base Template

```solidity
contract MYED is ERC20, Ownable {
    using Address for address payable;

    IUniswapV2Router02 public uniswapV2Router;
    address public  uniswapV2Pair;

    mapping (address => bool) private _isExcludedFromFees;

    string  public creator;

    uint256 public  feeOnBuy;
    uint256 public  feeOnSell;

    address public  marketingWallet;

    uint256 public  swapTokensAtAmount;
    bool     private swapping;

    bool     public swapEnabled;

    event ExcludeFromFees(address indexed account, bool
isExcluded);
    event MarketingWalletChanged(address marketingWallet);
    event DevelopmentWalletChanged(address developmentWallet);
    event UpdateBuyFees(uint256 feeOnBuy);
    event UpdateSellFees(uint256 feeOnSell);
    event UpdateWalletToWalletTransferFee(uint256
walletToWalletTransferFee);
    event SwapAndSendFee(uint256 tokensSwapped, uint256 bnbSend);
    event SwapTokensAtAmountUpdated(uint256 swapTokensAtAmount);

    constructor () ERC20("MyEducation", "MYED")
    {
        address router;
        if (block.chainid == 56) {
            router = 0x10ED43C718714eb63d5aA57B78B54704E256024E;
// BSC Pancake Mainnet Router
        } else if (block.chainid == 97) {
```

```solidity
            router = 0xD99D1c33F9fC3444f8101754aBC46c52416550D1;
// BSC Pancake Testnet Router
        } else if (block.chainid == 1 || block.chainid == 5) {
            router = 0x7a250d5630B4cF539739dF2C5dAcb4c659F2488D;
// ETH Uniswap Mainnet % Testnet
        } else {
            revert();
        }

        IUniswapV2Router02 _uniswapV2Router =
IUniswapV2Router02(router);
        address _uniswapV2Pair =
IUniswapV2Factory(_uniswapV2Router.factory())
            .createPair(address(this), _uniswapV2Router.WETH());

        uniswapV2Router = _uniswapV2Router;
        uniswapV2Pair   = _uniswapV2Pair;

        _approve(address(this), address(uniswapV2Router),
type(uint256).max);

        creator = "coinsult.net";

        feeOnBuy  = 2;
        feeOnSell = 2;

        marketingWallet =
0xAd1aA14523Cbc1481A34b1057281cCEa24b74535;

        _isExcludedFromFees[owner()] = true;
        _isExcludedFromFees[address(0xdead)] = true;
        _isExcludedFromFees[address(this)] = true;
        _isExcludedFromFees[marketingWallet] = true;

        _mint(owner(), 1e10 * (10 ** decimals()));
        swapTokensAtAmount = totalSupply() / 5_000;

        swapEnabled = true;
    }

    receive() external payable {
```

```solidity
    }

    function claimStuckTokens(address token) external onlyOwner {
        require(token != address(this), "Owner cannot claim
contract's balance of its own tokens");
        if (token == address(0x0)) {
            payable(msg.sender).sendValue(address(this).balance);
            return;
        }
        IERC20 ERC20token = IERC20(token);
        uint256 balance = ERC20token.balanceOf(address(this));
        ERC20token.transfer(msg.sender, balance);
    }

    /////////////////////////////////////// FEE SYSTEM

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

        emit ExcludeFromFees(account, excluded);
    }

    function isExcludedFromFees(address account) public view
returns(bool) {
        return _isExcludedFromFees[account];
    }

    function updateBuyFees(uint256 _feeOnBuy) external onlyOwner {
        feeOnBuy = _feeOnBuy;

        require(_feeOnBuy <= 2, "Total Fees cannot exceed the
maximum");

        emit UpdateBuyFees(feeOnBuy);
    }

    function updateSellFees(uint256 _feeOnSell) external onlyOwner
{
        feeOnSell = _feeOnSell;
```

```solidity
        require(_feeOnSell <= 2, "Total Fees cannot exceed the
maximum");

        emit UpdateSellFees(feeOnSell);
    }

    function changeMarketingWallet(address _marketingWallet)
external onlyOwner{
        require(_marketingWallet != marketingWallet,"Marketing
wallet is already that address");
        require(_marketingWallet != address(0),"Marketing wallet
cannot be the zero address");
        marketingWallet = _marketingWallet;

        emit MarketingWalletChanged(marketingWallet);
    }

    //
    //
    //
    //
    //
    //

    function _transfer(address from,address to,uint256 amount)
internal  override {
        require(from != address(0), "ERC20: transfer from the zero
address");
        require(to != address(0), "ERC20: transfer to the zero
```

```solidity
address");

        if (amount == 0) {
            super._transfer(from, to, 0);
            return;
        }

        uint256 contractTokenBalance =
balanceOf(address(this));

        bool canSwap = contractTokenBalance >= swapTokensAtAmount;

        if (canSwap &&
            !swapping &&
            to == uniswapV2Pair &&
            feeOnBuy + feeOnSell > 0 &&
            swapEnabled
        ) {
            swapping = true;

            swapAndSendFee(swapTokensAtAmount);

            swapping = false;
        }

        uint256 _totalFees;
        if (_isExcludedFromFees[from] || _isExcludedFromFees[to]
|| swapping) {
            _totalFees = 0;
        } else if (from == uniswapV2Pair) {
            _totalFees = feeOnBuy;
        } else if (to == uniswapV2Pair) {
            _totalFees = feeOnSell;
        } else {
            _totalFees = 0;
        }

        if (_totalFees > 0) {
            uint256 fees = (amount * _totalFees) / 100;
            amount = amount - fees;
            super._transfer(from, address(this), fees);
        }
```

```solidity
        super._transfer(from, to, amount);
    }

    function setSwapEnabled(bool _enabled) external onlyOwner{
        require(swapEnabled != _enabled, "swapEnabled already at
this state.");
        swapEnabled = _enabled;
    }

    function setSwapTokensAtAmount(uint256 newAmount) external
onlyOwner{
        require(newAmount > totalSupply() / 1_000_000,
"SwapTokensAtAmount must be greater than 0.0001% of total
supply");
        swapTokensAtAmount = newAmount;

        emit SwapTokensAtAmountUpdated(swapTokensAtAmount);
    }

    function swapAndSendFee(uint256 tokenAmount) private {
        uint256 initialBalance = address(this).balance;

        address[] memory path = new address[](2);
        path[0] = address(this);
        path[1] = uniswapV2Router.WETH();


uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens
(
            tokenAmount,
            0,
            path,
            address(this),
            block.timestamp);

        uint256 newBalance = address(this).balance -
initialBalance;

        payable(marketingWallet).sendValue(newBalance);

        emit SwapAndSendFee(tokenAmount, newBalance);
```

```
    }
}
```

READ CONTRACT (ONLY NEED TO KNOW)

1. Fee on Buy
2 uint256
(Function for read stakeFee)

2 Fee on sell
2 uint256
(Function for read withdrawFee)

WEBSITE REVIEW

🟢 **Mobile Friendly**

🟢 **Contains no code error**

🟢 **SSL Secured (By Let's Encrypt SSL)**

Domain .app - Tracked by whois

First Contentful Paint:	1.6s
Fully Loaded Time	9.3s
Performance	?
Accessibility	?
Best Practices	?
SEO	?

RUG-PULL REVIEW

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

- Locked Liquidity (Locked by pinksale)

    will be updated after listing dex

- TOP 5 Holder.

    will be updated after listing dex

- The Team is KYC By Pinksale

- The Contract is SAFU By Coinsult

HONEYPOT REVIEW

- Ability to sell.

- The owner is not able to pause the contract.

- The owner can't set fees over 5% for stake and 20% for withdraw

Note: Please check the disclaimer above and note that the audit makes no statements or warranties on the 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.