BlockSAFU

# ADVANCE MANUAL
# SMART CONTRACT AUDIT


**Project:** Midas Miner

**Website:** https://www.midasminer.io/


## BlockSAFU Score:

# 82

## Contract Address:

**0xFdcaBD329dc59615dca6Df02Ff942C047468E61C**

# 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 to 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

**BlockSAFU was commissioned by Midas Miner to complete a Smart Contract audit. The objective of the Audit is to achieve the following:**

- Review the Project and experience and Development team
- Ensure that the Smart Contract functions are necessary and operate as intended.
- Identify any vulnerabilities in the Smart Contract code.

DISCLAIMER: This Audit is intended to inform about token Contract Risks, the result does not imply an endorsement or provide financial advice in any way, all investments are made at your own risk. (https://blocksafu.com)

# SMART CONTRACT REVIEW

| | |
|---|---|
| Token Name | **Midas Miner** |
| Token Symbol | **MMI** |
| Token Decimal | 9 |
| Total Supply | 300,000,000 **MMI** |
| Contract Address | 0xFdcaBD329dc59615dca6Df02Ff942C047468E61C |
| Deployer Address | 0x3a2B482cAbDEE293969A6D427c561D7A11cb545D |
| Owner Address | 0x3a2B482cAbDEE293969A6D427c561D7A11cb545D |
| Tax Fees Buy | 4% |
| Tax Fees Sell | 4% |
| Gas Used for Buy | *will be updated after the DEX listing* |
| Gas Used for Sell | *will be updated after the DEX listing* |
| Contract Created | Aug-24-2022 04:19:43 AM +UTC |
| Initial Liquidity | *will be updated after the DEX listing* |
| Liquidity Status | Locked |
| Unlocked Date | *will be updated after the DEX listing* |
| Verified CA | Yes |
| Compiler | v0.8.7+commit.e28d00a7 |
| Optimization | Yes with 200 runs |
| Sol License | MIT License |
| Top 5 Holders | *will be updated after the DEX listing* |
| Other | default evmVersion |

# TAX

| | | | |
|---|---|---|---|
| **BUY** | 4% | **SELL** | 4% |

## OVERVIEW

Mint Function

- No mint functions.

Fees

- Buy 4% (owner can’t set fees over 10% & can’t zero).
- Sell 4% (owner can’t set fees over 10% & can’t zero).

Tx Amount

- Owner cannot set the max tx amount.

Transfer Pausable

- Owner cannot pause.

Blacklist

- Owner cannot blacklist.

Ownership

- Owner cannot take back ownership.

Proxy

- This contract has no proxy.

Anti Whale

- Owner cannot limit the number of wallet holdings.

Trading Cooldown

- Owner cannot set the selling time interval.

## Token Holder

| Rank | Address | Quantity | Percentage | Analytics |
|---|---|---|---|---|
| 1 | 0x3a2b482cabdee293969a6d427c561d7a11cb545d | 300,000,000 | 100.0000% | |

[ Download CSV Export ]

## Team Review

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

## Official Website And Social Media

Website: https://www.midasminer.io/

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

Twitter: https://twitter.com/MidasMiner

## MANUAL CODE REVIEW

### Minor-risk

1 minor-risk code issue found

Could be fixed, and will not bring problems.

1. The return value of an external transfer/transferFrom return value is checked. Recommendation: use SafeERC20, or ensure that the transfer/transferFrom return value is checked

```
function transferFrom(
      address sender,
      address recipient,
      uint256 amount
   ) external returns (bool);
```

### Medium-risk

0 medium-risk code issue 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

```
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

## 2. SafeMath Contract

```
library SafeMath {
...
    function add(uint256 a, uint256 b) internal pure returns
(uint256) {
        uint256 c = a + b;
        require(c >= a, "SafeMath: addition overflow");
        return c;
    }
...
    function sub(uint256 a, uint256 b, string memory errorMessage)
internal pure returns (uint256) {
        require(b <= a, errorMessage);
        uint256 c = a - b;

        return c;
    }
    /**
     * @dev Returns the multiplication of two unsigned integers,
reverting on
     * overflow.
     *
     * Counterpart to Solidity's `*` operator.
     *
     * Requirements:
     *
     * - Multiplication cannot overflow.
     */
  ...
    function mod(
        uint256 a,
        uint256 b,
        string memory errorMessage
    ) internal pure returns (uint256) {
        unchecked {
            require(b > 0, errorMessage);
            return a % b;
        }
    }
}
```

Standard Safemath contract

### 3. MMI Contract

```
contract MidasMiner is AntiBotsToken {
    using SafeMath for uint256;

    mapping(address=>bool) exchanges;

    uint256 public buyTax = 4;
    uint256 public sellTax = 4;
    uint256 public constant maxTax = 10;
    mapping (address => bool) public excludeTax;
    address[] public taxers;
    uint256 public taxThreshold;
    receive() external payable { }

    constructor() AntiBotsToken("Midas Miner", "MMI") {
        _mint(msg.sender, 300000000*10**9);
        taxThreshold = totalSupply().div(5000);

        excludeTax[msg.sender] = true;
        excludeTax[address(this)] = true;
    }

    function _transfer(address from, address to, uint256 amount)
internal override {
        if(balanceOf(address(this)) >= taxThreshold) {
            transferTax();
        }
        if (exchanges[to] == true) {
            super._transfer(from, to, amount);
        } else {
            uint256 amountReceived = excludeTax[from] ?
amount:takeTax(from, to, amount);
            super._transfer(from, to, amountReceived);
        }
    }

    function setExchanges(address exchange, bool enable) external
onlyOwner {
        exchanges[exchange] = enable;
    }
    function setExcludeTax(address addr, bool isExcluded) external
onlyOwner {
```

```
        excludeTax[addr] = isExcluded;
    }
    function setTax(uint256 _buyTax, uint256 _sellTax) external
onlyOwner {
        require(_buyTax <= maxTax && _sellTax <= maxTax, "tax got
limited");
        buyTax = _buyTax;
        sellTax = _sellTax;
    }
    function takeTax(address from, address to, uint256 amount)
internal returns (uint256) {
        uint256 tax = amount.mul(to == pair ? sellTax :
buyTax).div(100);
        super._transfer(from, address(this), tax);
        return amount.sub(tax);
    }
    function transferTax() internal {
        uint length = taxers.length;
        uint256 amount = balanceOf(address(this));
        uint256 tax = amount.div(length);
        for(uint i=0;i<length;i++) {
            super._transfer(address(this), taxers[i], tax);
        }
    }
    function setTaxers(address[] memory newTaxers) external
onlyOwner {
        uint length = taxers.length;
        for(uint i=0; i<length; i++) {
            excludeTax[taxers[taxers.length-1]] = false;
            taxers.pop();
        }
        uint newLen = newTaxers.length;
        for(uint i=0; i<newLen; i++) {
            taxers.push(newTaxers[i]);
            excludeTax[newTaxers[i]] = true;
        }
    }
    function setTaxThreshold(uint256 _amount) external onlyOwner {
        taxThreshold = _amount;
    }
}
```

**4. Tax Fee contract**

```
function setTax(uint256 _buyTax, uint256 _sellTax) external
onlyOwner {
        require(_buyTax <= maxTax && _sellTax <= maxTax, "tax got
limited");
        buyTax = _buyTax;
        sellTax = _sellTax;
    }
```

The owner can’t set fees over 10%

## READ CONTRACT (ONLY NEED TO KNOW)

1. buyTax
4 uint256
(Shows Contract buy tax)

2. decimals
9 uint8
(Shows the token decimals)

3. endAntiAfter
2 uint256
(Function For read endAntiAfter)

4. decimals
18 uint8
(Function for read decimals)

5. maxTax
10 uint256
(Function for read maxTax)

6. name
Midas Miner string
(Function for read name)

7. symbol
MMI string
(Function for read Token symbol)

## WRITE CONTRACT

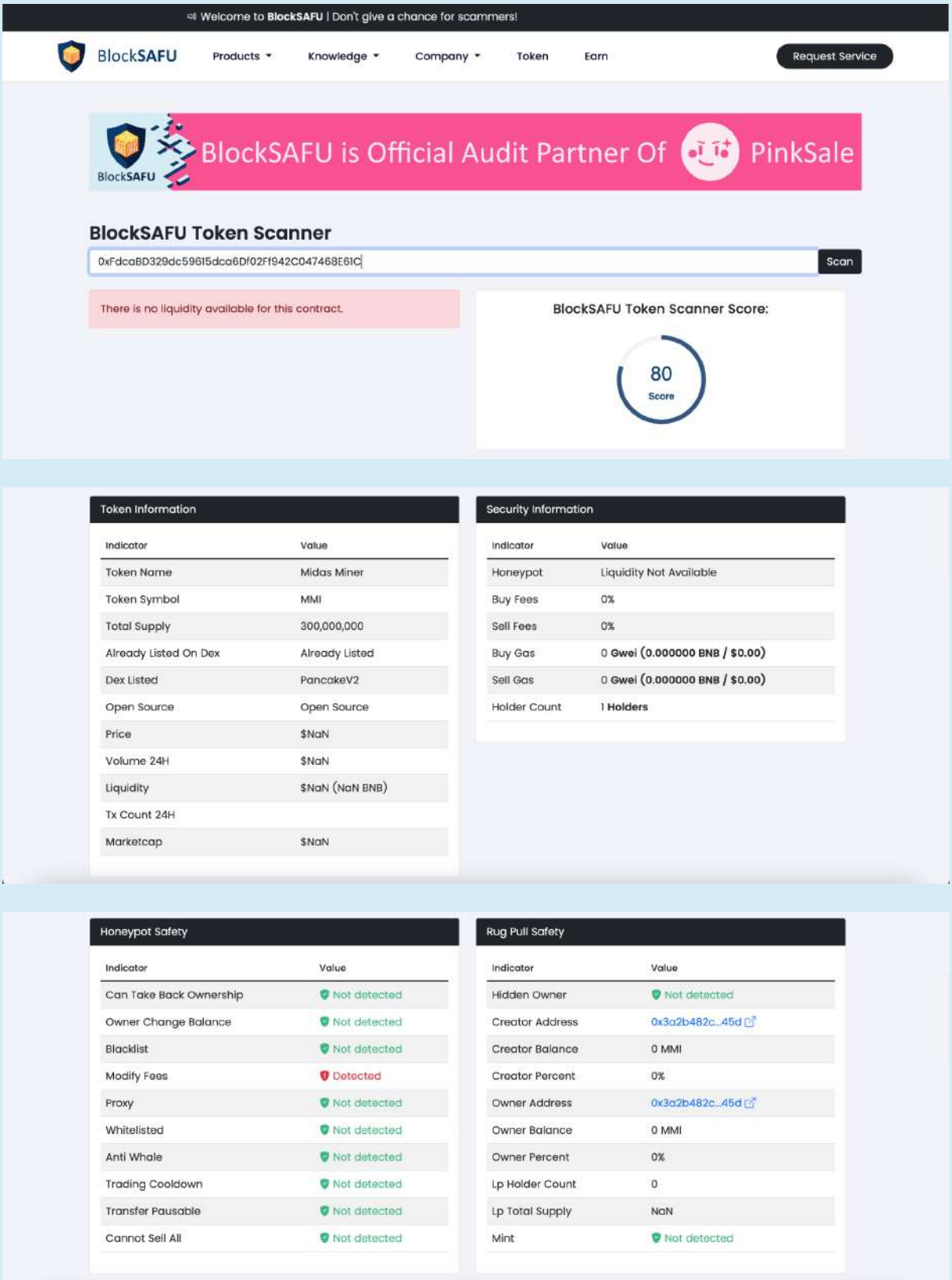
1. setExcludeTax
addr (address)
isExcluded (bool)
(The form is filled with address and isExcluded the true or false for exclude tax)

2. setTax
_buyTax
_sellTax
(The function for set buy and sell tax)

3. transferOwnership
newOwner (address)
(Its function is to change the owner)

# BlockSAFU TOKEN SCANNER

https://blocksafu.com/token-scanner



## WEBSITE REVIEW


- **Mobile Friendly**
- **Contains no code error**
- **SSL Secured (R3 SSL)**

**Web-Tech stack:** n/a

Domain .io (Namecheap) - Tracked by whois

| | |
|---|---|
| First Contentful Paint: | 591ms |
| Fully Loaded Time | 6.2s |
| Performance | 72% |
| Accessibility | 97% |
| Best Practices | 67% |
| SEO | 100% |

## 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 DEX listing*)

- TOP 5 Holder.

  (*Will be updated after DEX listing*)

- The Team Not KYC on Blocksafu

## HONEYPOT REVIEW

- Ability to sell.
- The owner is not able to pause the contract.
- The owner can’t set fees over 10%

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.