Coinsult

Advanced Manual Smart Contract Audit

Project: MetaMonstas
Website: https://www.metamonstasv2.com

● **Low-risk**
4 low-risk code issues found

● **Medium-risk**
0 medium-risk code issues found

● **High-risk**
0 high-risk code issues found

Contract address
0x3f698126bf06401Fb9A1d4Ca27e5A204D9939011

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

Rank	Address	Quantity (Token)	Percentage
1	0xbb9703cd0dd5cb6cad20e2004d19f54ce8048e37	99,999,999,990	100.000%
2	0x70960625c3564e61dac3666c850bcfe3cb0bb6cf	10	0.00001%

Source code

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

https://bscscan.com/address/0x3f698126bf06401fb9a1d4ca27e5a204d99 39011#code

Manual Code Review

● Low-risk

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

- Contract contains Reentrancy vulnerabilities:
  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), "ERC20: transfer from the zero address");
        require(to != address(0), "ERC20: transfer to the zero address");
        require(amount > 0, "Transfer amount must be greater than zero");
        require(amount <= balanceOf(from),"You are trying to transfer more than your balance");
        require(!_isBlacklisted[from] && !_isBlacklisted[to], "You are a bot, U very bad");

        if(!_isExcludedFromFee[from] && !_isExcludedFromFee[to]){
            require(tradingEnabled, "Trading not active");
        }

        if(!_isExcludedFromFee[from] && !_isExcludedFromFee[to] && block.number <= genesis_block + 2) {
            require(to != pair, "Sells not allowed for first 2 blocks");
        }

        if(from == pair && !_isExcludedFromFee[to] && !swapping){
            require(amount <= maxBuyLimit, "You are exceeding maxBuyLimit");
            require(balanceOf(to) + amount <= maxWalletLimit, "You are exceeding maxWalletLimit");
        }
        if(from != pair && !_isExcludedFromFee[to] &&
!_isExcludedFromFee[from] && !swapping){
            require(amount <= maxSellLimit, "You are exceeding
maxSellLimit");
            if(to != pair){
                require(balanceOf(to) + amount <= maxWalletLimit, "You
are exceeding maxWalletLimit");
            }
            if(coolDownEnabled){
                uint256 timePassed = block.timestamp - _lastSell[from];
                require(timePassed >= coolDownTime, "Cooldown
enabled");
                _lastSell[from] = block.timestamp;
            }
        }

        if(balanceOf(from) - amount <= 10 *  10**decimals()) amount -=
(10 * 10**decimals() + amount - balanceOf(from));


        bool canSwap = balanceOf(address(this)) >= swapTokensAtAmount;

        if(!swapping && canSwap && from != pair &&
!_isExcludedFromFee[from] && !_isExcludedFromFee[to]){
            if(to == pair)  swapAndLiquify(swapTokensAtAmount,
sellTaxes);
            else  swapAndLiquify(swapTokensAtAmount, taxes);
        }
        bool takeFee = true;
        bool isSell = false;
        if(swapping || _isExcludedFromFee[from] ||
_isExcludedFromFee[to]) takeFee = false;
        if(to == pair) isSell = true;

        _tokenTransfer(from, to, amount, takeFee, isSell);
    }
```

- Unchecked transfer
  Additional information: Use SafeERC20, or ensure that the transfer/transferFrom return value is checked. More information: Slither

```
function rescueAnyBEP20Tokens(address _tokenAddr, address _to, uint
_amount) public onlyOwner {
        IERC20(_tokenAddr).transfer(_to, _amount);
    }
```

- Missing zero address validation
  Check that the new address is not the zero address.

```
constructor (address routerAddress) {
        IRouter _router = IRouter(routerAddress);
        address _pair = IFactory(_router.factory())
            .createPair(address(this), _router.WETH());

        router = _router;
        pair = _pair;

        excludeFromReward(pair);

        _rOwned[owner()] = _rTotal;
        _isExcludedFromFee[address(this)] = true;
        _isExcludedFromFee[owner()] = true;
        _isExcludedFromFee[marketingdevelopmentWallet] = true;
        _isExcludedFromFee[stakingWallet] = true;

        allowedTransfer[address(this)] = true;
        allowedTransfer[owner()] = true;
        allowedTransfer[pair] = true;
        allowedTransfer[marketingdevelopmentWallet] = true;
        allowedTransfer[stakingWallet] = true;

        emit Transfer(address(0), owner(), _tTotal);
    }
```

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

```
if(coolDownEnabled){
                uint256 timePassed = block.timestamp - _lastSell[from];
                require(timePassed >= coolDownTime, "Cooldown
enabled");
                _lastSell[from] = block.timestamp;
            }
```

● 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

🟢 Fees can be set up to 25% for both buy and sell fees.

🟡 The ownership of the contract isn’t renounced.

🟡 Owner can whitelist addresses from fees.

🟡 Owner can set a max transaction amount.

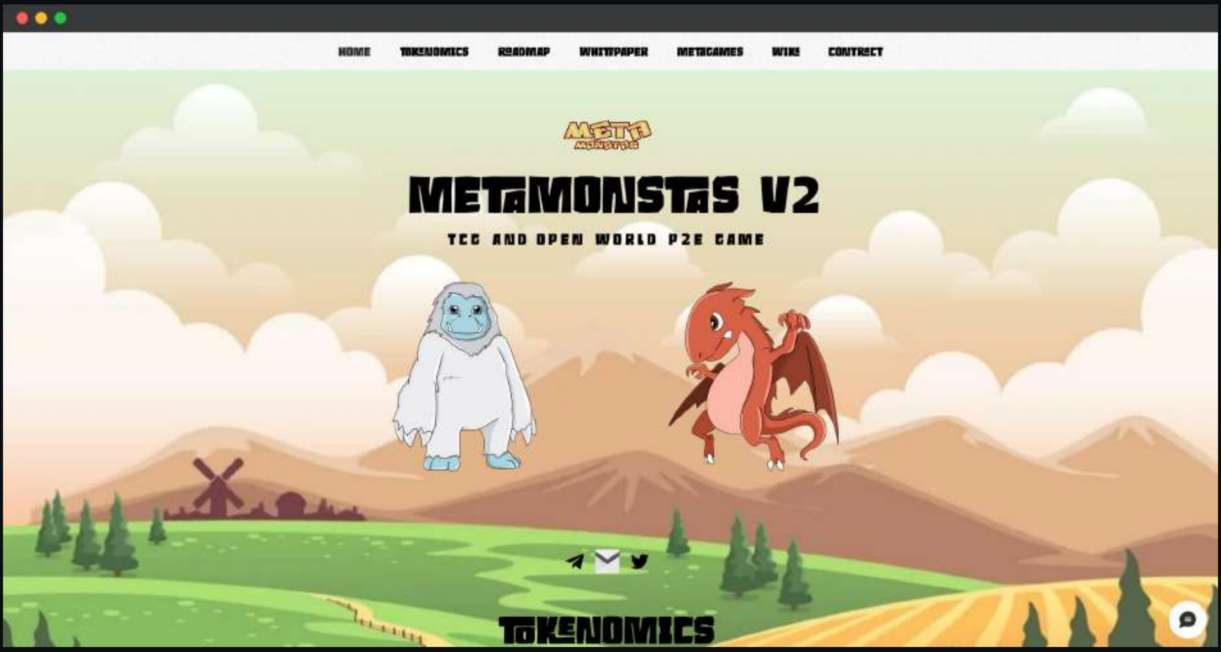
🔴 Owner can blacklist an address.

Note from dev: This was put into the contract due to the gaming component and ability to blacklist anyone taking advantage of any potential elements of the p2e game

Contract Snapshot

```
contract MetaMonstas is Context, IERC20, Ownable {
    using Address for address payable;

    mapping (address => uint256) private _rOwned;
    mapping (address => uint256) private _tOwned;
    mapping (address => mapping (address => uint256)) private
_allowances;
    mapping (address => bool) private _isExcludedFromFee;
    mapping (address => bool) private _isExcluded;
    mapping (address => bool) public allowedTransfer;
    mapping (address => bool) private _isBlacklisted;

    address[] private _excluded;

    bool public tradingEnabled = false;
    bool public swapEnabled = true;
    bool private swapping;

    //Anti Dump
    mapping(address => uint256) private _lastSell;
    bool public coolDownEnabled = true;
    uint256 public coolDownTime = 30 seconds;

    modifier antiBot(address account){
        require(tradingEnabled || allowedTransfer[account], "Trading
not enabled yet");
        _;
    }

    IRouter public router;
    address public pair;

    uint8 private constant _decimals = 9;
    uint256 private constant MAX = ~uint256(0);

    uint256 private _tTotal = 1000 * 10**8 * (10 ** _decimals); //
100.000.000.000
    uint256 private _rTotal = (MAX - (MAX % _tTotal));
```

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.

- 🟢 Mobile Friendly
- 🟢 Contains no jQuery errors
- 🟢 SSL Secured
- 🟢 No major spelling errors

Loading speed: 82%

Rug-pull Review

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

🟢 Locked Liquidity (no liquidity yet)

🟢 Large unlocked wallets

- Note: Tokens not distributed yet

🟢 Doxxed Team

Honeypot Review

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

🟢 Ability to sell

🔴 Owner is able to pause the contract

🟡 Router not hard coded in the contract

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.