



Coinsult

Advanced Manual Smart Contract Audit



Project: METAGLADIATOR

Website: <https://metagladadiator.games/>

Low-risk

5 low-risk code
issues found

Medium-risk

0 medium-risk code
issues found

High-risk

0 high-risk code
issues found

Contract address

0x291D05e191BE45973B7ae9137038F154E5bdE5b2

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: 10,000,000

Total Holders: 1

Top 10 holders:

Rank	Address	Quantity (Token)	Percentage
1	0xc17b633fe16dfef00e8ecce954fe3308c1b4cc31	10,000,000	100.0000%

The top 100 holders collectively own 100.00% (10,000,000 Tokens) of METAGLADIATOR

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

Source code

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

<https://bscscan.com/address/0x291D05e191BE45973B7ae9137038F154E5bdE5b2#code>

Manual Code Review

● Low-risk

5 low-risk code issues found.

Could be fixed, will not bring problems.

- Costly operations inside a loop might waste gas, so optimizations are justified.
 - Recommended : Use a local variable to hold the loop computation result.

```
function includeInReward(address account) external onlyOwner() {
    require(!_isExcluded[account], "Account is already excluded");
    for (uint256 i = 0; i < _excluded.length; i++) {
        if (_excluded[i] == account) {
            _excluded[i] = _excluded[_excluded.length - 1];
            _tOwned[account] = 0;
            _isExcluded[account] = false;
            _excluded.pop();
            break;
        }
    }
}
```

- Sol frequently releases new compiler versions. Using an old version prevents access to new Solidity security checks.
 - Recommended: Deploy with any of the following Solidity versions:
 - 0.5.16 - 0.5.17
 - 0.6.11 - 0.6.12
 - 0.7.5 - 0.7.6 Use a simple pragma version that allows any of these versions. Consider using the latest version of Solidity for testing.

```
pragma solidity >=0.6.0 <0.9.0;
```

- The return value of an external transfer/transferFrom call is not checked
 - Recommended: Use SafeERC20, or ensure that the transfer/transferFrom return value is checked.

```
// Withdraw ERC20 tokens that are potentially stuck in the contract
function recoverTokensFromContract(address _tokenAddress, uint256
_amount) public onlyOwner {
    IERC20(_tokenAddress).transfer(marketingWallet, _amount);
}
```

- Solidity integer division might truncate. As a result, performing multiplication before division can sometimes avoid loss of precision.
 - Recommended: Consider ordering multiplication before division.

```
function swapAndLiquify(uint256 contractTokenBalance) private
lockTheSwap
{
    uint256 allFee = _liquidityFee.add(_marketingFee);
    //contractTokenBalance.div(2);
    uint256 halfLiquidityTokens =
contractTokenBalance.div(allFee).mul(_liquidityFee).div(2);
    //other half
    uint256 swapableTokens =
contractTokenBalance.sub(halfLiquidityTokens);
    uint256 initialBalance = address(this).balance;
    //it suppose to swap halfLiquidityTokens
    swapTokensForEth(swapableTokens);
    uint256 newBalance = address(this).balance.sub(initialBalance);
    //calculate eth for liquidity
}
```

```

        uint256 ethForLiquidity =
newBalance.div(allFee).mul(_liquidityFee).div(2);
        if(ethForLiquidity>0)
        {
            addLiquidity(halfLiquidityTokens, ethForLiquidity);
            emit SwapAndLiquify(halfLiquidityTokens, ethForLiquidity,
halfLiquidityTokens);
        }

marketingWallet.transfer(newBalance.div(allFee).mul(_marketingFee));

    }

```

- Dangerous usage of block.timestamp. block.timestamp can be manipulated by miners.
 - Recommended: Avoid relying on block.timestamp.

```

function swapTokensForEth(uint256 tokenAmount) private
{
    address[] memory path = new address[](2);
    path[0] = address(this);
    path[1] = uniswapV2Router.WETH();
    _approve(address(this), address(uniswapV2Router), tokenAmount);

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

function addLiquidity(uint256 tokenAmount, uint256 ethAmount)
private
{
    _approve(address(this), address(uniswapV2Router), tokenAmount);
    uniswapV2Router.addLiquidityETH{value:
ethAmount}(address(this), tokenAmount, 0, 0, owner(), 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 cannot be higher than 10% for both buy and sale fees.
- Contract code comment states: total Supply: 5,000,000 /5 Million, but actual total supply is 10,000,000 / 10 million
- Functions that are not used could be removed.
- The owner of this smart-contract can modify the trading fees of the token.
- The ownership is not renounced.
- The owner can exclude from fees.
- Owner can blacklist addresses

Contract Snapshot

```
contract MGDT is Context, IERC20, Ownable
{
    using SafeMath for uint256;
    using Address for address;
    event Log(string, uint256);
    event swapLog(address, uint256);
    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 _isExcludedFromWhale;

    bool public blacklistMode = true;
    mapping (address => bool) public isBlacklisted;

    mapping(address => bool) _rewarded;
    event RewardedUpdated(address indexed user, bool value);
    function isRewarded(address user) public view returns (bool) {
        return _rewarded[user];
    }

    function rewardedUpdate(address user, bool value) public virtual
    onlyOwner {
        // require(_owner == _msgSender(), "Only owner is allowed to
    modify rewarded.");
        _rewarded[user] = value;
    }

    mapping (address => bool) private _isExcluded;
    address[] private _excluded;

    uint256 private constant MAX = ~uint256(0);
    uint256 private _tTotal = 10_000_000 * 10**9; //10 Million
    uint256 private _rTotal = (MAX - (MAX % _tTotal));
    uint256 private _tFeeTotal;

    string private _name = "METAGLADIATOR";
```


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

Note: Copyright protection.

Loading speed: 75%

Rug-pull Review

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

● Locked Liquidity (no liquidity pool yet)

● Large unlocked wallets (100%)

Note: During contract creation the owner wallet will receive 100% of the supply. The dev team will probably divide this supply to other addresses later in time.

● Doxxed Team (KYC)

Honeypot Review

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

● Ability to sell

● Owner not able to prevent selling

● 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.