Coinsult

Advanced Manual Smart Contract Audit

Project: MetaCourse
Website: https://www.metacoursetoken.com

Low-risk	Medium-risk	High-risk
7 low-risk code issues found	0 medium-risk code issues found	0 high-risk code issues found

Contract address
0x5466B664a8B18A4545b15d41D01BaDA764c918dB

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 checks the contract for coding issues, we do not guarantee the use case of the code. We do not check the logic of the functions and if they are used for what they were intended to be used for.

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	Null Address: 0x000...dEaD	6,000,000,000,000,000	60.0000%
2	0x567cbbfa47859c20759b6d940c601c962cc9853c	300,000,000,000,000	3.0000%
3	0x0da912bfbba86766af34a0b4b84e8f8a4f5be779	293,456,005,001,336.26461505088282593	2.9346%
4	PancakeSwap V2: COURSE	292,226,655,505,475.34816152460861009	2.9223%
5	0xac03cc9b72c2754a8fdf56195c81cf7dddbc0dff	211,278,602,807,231.4346230332365099	2.1128%
6	0x7161a729f02010c077a441c57c89925954a2a31b	200,000,071,844,839.19959811391219834	2.0000%
7	0xfde6ccb8e7d2483e23c0c54098094b1398c6f5e4	200,000,071,844,839.19959811391219834	2.0000%
8	0x17e5371003d2eb35569c891fc7c82fa2e2a4cec9	200,000,015,965,519.82213291420271074	2.0000%
9	0xc3f5b24ba184d8da5b87614eca82026ccb860518	200,000,015,965,519.82213291420271074	2.0000%
10	0x58618f88d5d2dd68a416ca175af7ec2b10a8f4bc	200,000,000,000,000	2.0000%

Source code

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

https://bscscan.com/address/0x5466B664a8B18A4545b15d41D01BaDA764c918dB#code

Manual Code Review

🟢 Low-risk

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

- Literals with many digits are difficult to read and review.
  Recommendation: Use Ether suffix, Time suffix, or The scientific notation

```
    uint256 private _tTotal = 10000000000000000 * 10**17;
```

- Missing events for critical arithmetic parameters
  Emit an event for critical parameter changes.

```
    function setNumTokensSellToAddToLiquidity(uint256 newAmt, uint256
decimal) external onlyOwner() {
        numTokensSellToAddToLiquidity = newAmt*10**decimal;
    }

    function setMaxTxAmount(uint256 maxTxAmount, uint256 decimal)
external onlyOwner() {
        require(maxTxAmount > 0, "Cannot set transaction amount as
zero");
        _maxTxAmount = maxTxAmount * 10**decimal;
    }
```

- No zero address validation
  Check that the new address is not the zero address.

```
    function setRouterAddress(address newRouter) public onlyOwner() {
        IUniswapV2Router02 _newPancakeRouter =
IUniswapV2Router02(newRouter);
        uniswapV2Pair =
IUniswapV2Factory(_newPancakeRouter.factory()).createPair(address(this)
, _newPancakeRouter.WETH());
        uniswapV2Router = _newPancakeRouter;
    }
```

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

```
    function _transfer(
        address from,
        address to,
        uint256 amount
    ) private {
        require(from != address(0), "ERC20: transfer from the zero address");
        require(amount > 0, "Transfer amount must be greater than zero");

        // 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 uniswap pair.
        uint256 contractTokenBalance = balanceOf(address(this));
        bool overMinTokenBalance = contractTokenBalance >=
numTokensSellToAddToLiquidity;
        if (
            overMinTokenBalance &&
            !inSwapAndLiquify &&
            from != uniswapV2Pair &&
            swapAndLiquifyEnabled
        ) {
            contractTokenBalance = numTokensSellToAddToLiquidity;
            //add liquidity
            swapAndLiquify(contractTokenBalance);
        }

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

- Costly operations inside a loop

  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;
            }
        }
    }
```

- Unused return

  Ensure that all the return values of the function calls are used.

```
    function addLiquidity(uint256 tokenAmount, uint256 ethAmount)
private {
        // approve token transfer to cover all possible scenarios
        _approve(address(this), address(uniswapV2Router), tokenAmount);

        // add the liquidity
        uniswapV2Router.addLiquidityETH{value: ethAmount}(
            address(this),
            tokenAmount,
            0, // slippage is unavoidable
            0, // slippage is unavoidable
            owner(),
            block.timestamp
        );
    }
```

- Remove commented code

  Remove commented code for better readability and optimization of the contract

```
    /**
     * @dev Replacement for Solidity's `transfer`: sends `amount` wei
to
     * `recipient`, forwarding all available gas and reverting on
errors.
     *
     * https://eips.ethereum.org/EIPS/eip-1884[EIP1884] increases the
gas cost
     * of certain opcodes, possibly making contracts go over the 2300
gas limit
     * imposed by `transfer`, making them unable to receive funds via
     * `transfer`. {sendValue} removes this limitation.
     *
     *
https://diligence.consensys.net/posts/2019/09/stop-using-soliditys-tran
sfer-now/[Learn more].
     *
     * IMPORTANT: because control is transferred to `recipient`, care
must be
     * taken to not create reentrancy vulnerabilities. Consider using
     * {ReentrancyGuard} or the
     *
https://solidity.readthedocs.io/en/v0.5.11/security-considerations.html
#use-the-checks-effects-interactions-pattern[checks-effects-interaction
s pattern].
     */
    function sendValue(address payable recipient, uint256 amount)
internal {
        require(address(this).balance >= amount, "Address: insufficient
CCTS balance");

        // solhint-disable-next-line avoid-low-level-calls,
avoid-call-value
        (bool success, ) = recipient.call{ value: amount }("");
        require(success, "Address: unable to send value, recipient may
have reverted");
    }
```

🟡 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

🟢 Owner can not change the buy and sell fees

🟡 Owner can exclude from fees

🟡 Owner can set max transaction amount without a limit

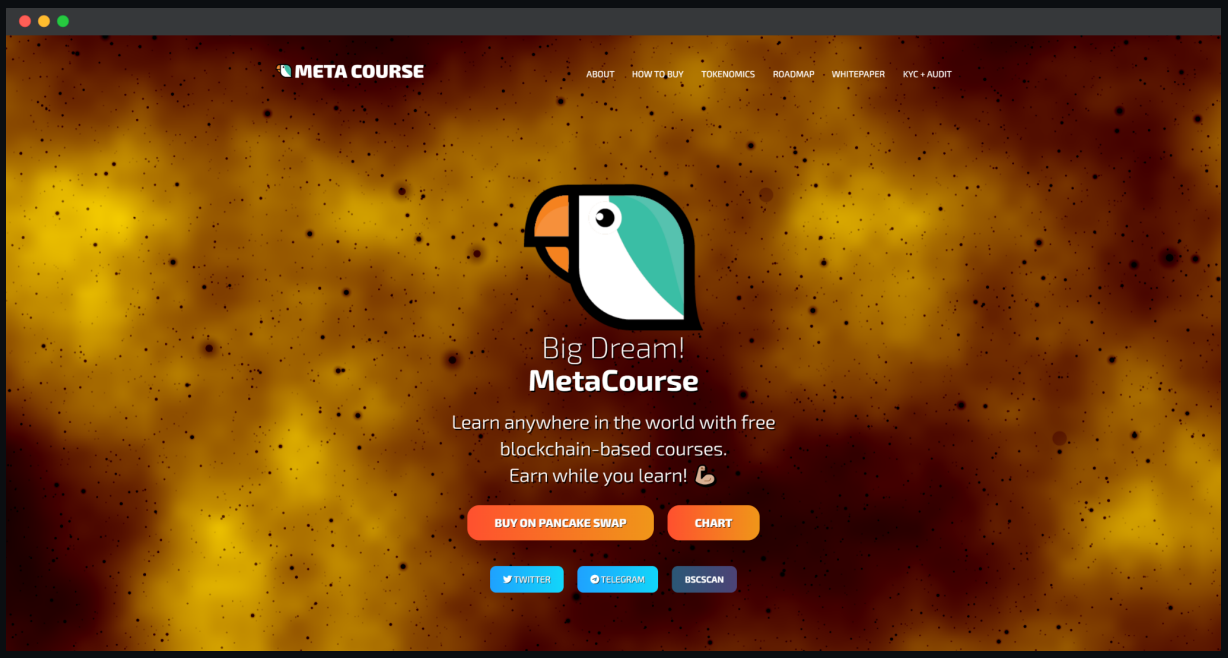
```
    function setMaxTxAmount(uint256 maxTxAmount, uint256 decimal)
external onlyOwner() {
        require(maxTxAmount > 0, "Cannot set transaction amount as
zero");
        _maxTxAmount = maxTxAmount * 10**decimal;
    }
```

🟡 The ownership of the contract isn't renounced

Contract Snapshot

```
contract METACOURSE is Context, IERC20, Ownable {
    using SafeMath for uint256;
    using Address for address;

    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;
    address[] private _excluded;

    uint256 private constant MAX = ~uint256(0);
    uint256 private _tTotal = 10000000000000000 * 10**17;
    uint256 private _rTotal = (MAX - (MAX % _tTotal));
    uint256 private _tFeeTotal;

    string private _name = "META COURSE";
    string private _symbol = "COURSE";
    uint8 private _decimals = 17;


    uint256 public _liquidityFee =0;
    uint256 public _devFee = 2;
    uint256 public _taxFee = 2;
    uint256 public _marketingFee = 4;
    uint256 public _burnFee = 0;

    uint256 private _previousDevFee = _devFee;
    uint256 private _previousTaxFee = _taxFee;
    uint256 private _previousLiquidityFee = _liquidityFee;
    uint256 private _previousmarketingFee = _marketingFee;
    uint256 private _previousBurnFee = _burnFee;
```

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: 88%

Rug-pull Review

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

🟢 Locked Liquidity

🟢 No large unlocked wallets

🟢 Doxxed Team (KYC at Coinsult)

Honeypot Review

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

🟢 Ability to sell

🟢 Owner is not able to pause 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.