Coinsult

Advanced Manual Smart Contract Audit

Project: MetaPet
Website: https://metapet.finance

🟢 **Low-risk**	🟡 **Medium-risk**	🔴 **High-risk**
4 low-risk code issues found	0 medium-risk code issues found	0 high-risk code issues found

Contract address

0x2633982e09F44b0dE2C480e9e4Eb72b3F0e54d47

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

Rank	Address	Quantity (Token)	Percentage
1	0xe06bdb4c45ead0856bf0181d7ea145bc0fbfa1ce	10,000,000	100.0000%

Source code

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

https://bscscan.com/address/0x2633982e09F44b0dE2C480e9e4Eb72b3F0e54d47#code

Manual Code Review

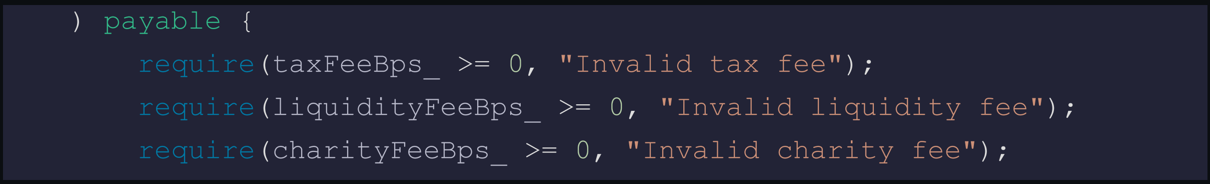
● Low-risk

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

- Contract has expressions that are tautologies or contradictions.

  fees are uint16, fees >= 0 will be always true.

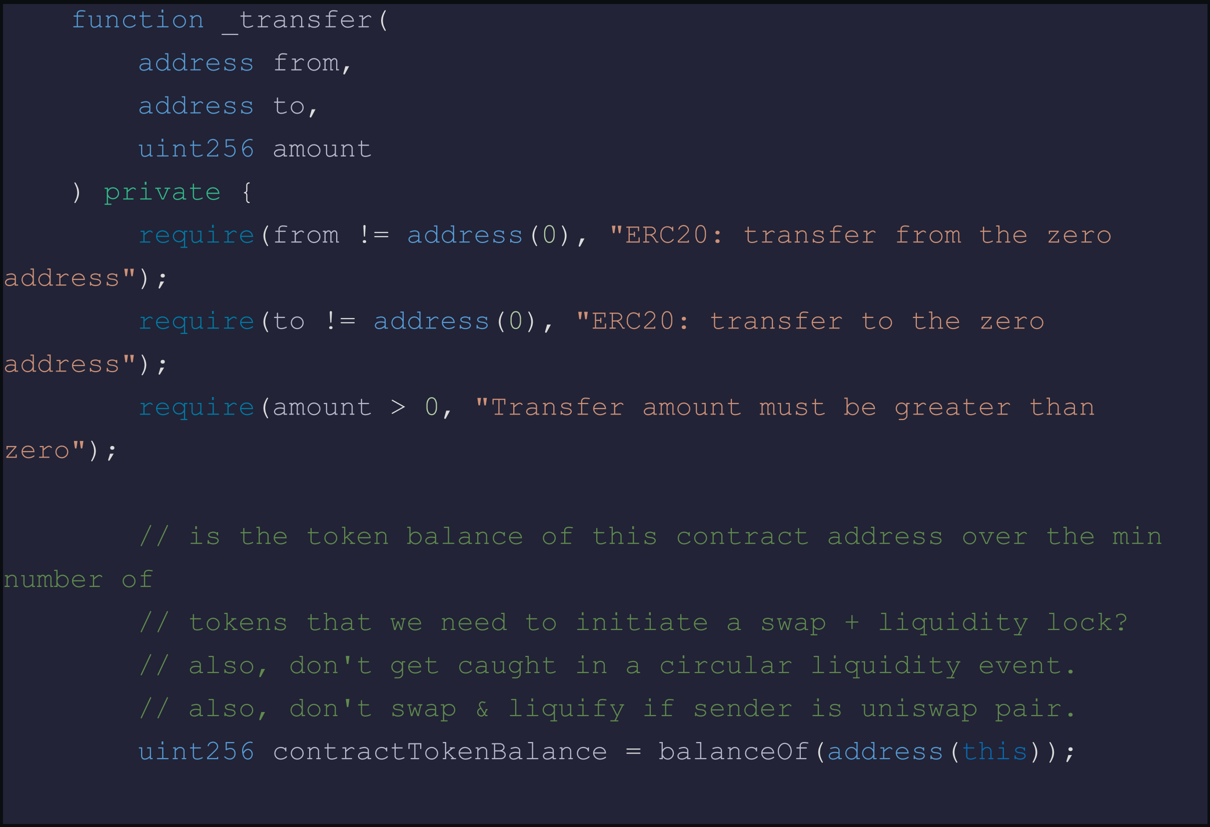
  Note: Contract already deployed so no risk involved anymore



```
    ) payable {
        require(taxFeeBps_ >= 0, "Invalid tax fee");
        require(liquidityFeeBps_ >= 0, "Invalid liquidity fee");
        require(charityFeeBps_ >= 0, "Invalid charity fee");
```

- 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");

        // 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);
        }

        //indicates if fee should be deducted from transfer
        bool takeFee = true;

        //if any account belongs to _isExcludedFromFee account then
remove the fee
        if (_isExcludedFromFee[from] || _isExcludedFromFee[to]) {
            takeFee = false;
        }

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

- Boolean constants can be used directly and do not need to be compare to true or false

  Remove the equality to the boolean constant.

```
    function ensureMaxTxAmount(
        address from,
        address to,
        uint256 amount
    ) private view {
        if (
            _isExcludedFromMaxTx[from] == false && // default will be
false
            _isExcludedFromMaxTx[to] == false // default will be false
        ) {
                require(amount <= _maxTxAmount, "Transfer amount
exceeds the maxTxAmount.");
```

```
        }
    }
```

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

```
    function setTaxFeePercent(uint256 taxFeeBps) external onlyOwner {
        _taxFee = taxFeeBps;
        require(
            _taxFee + _liquidityFee + _charityFee <= 10**4 / 4,
            "Total fee is over 25%"
        );
    }

    function setLiquidityFeePercent(uint256 liquidityFeeBps)
        external
        onlyOwner
    {
        _liquidityFee = liquidityFeeBps;
        require(
            _taxFee + _liquidityFee + _charityFee <= 10**4 / 4,
            "Total fee is over 25%"
        );
    }
```

🟡 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.

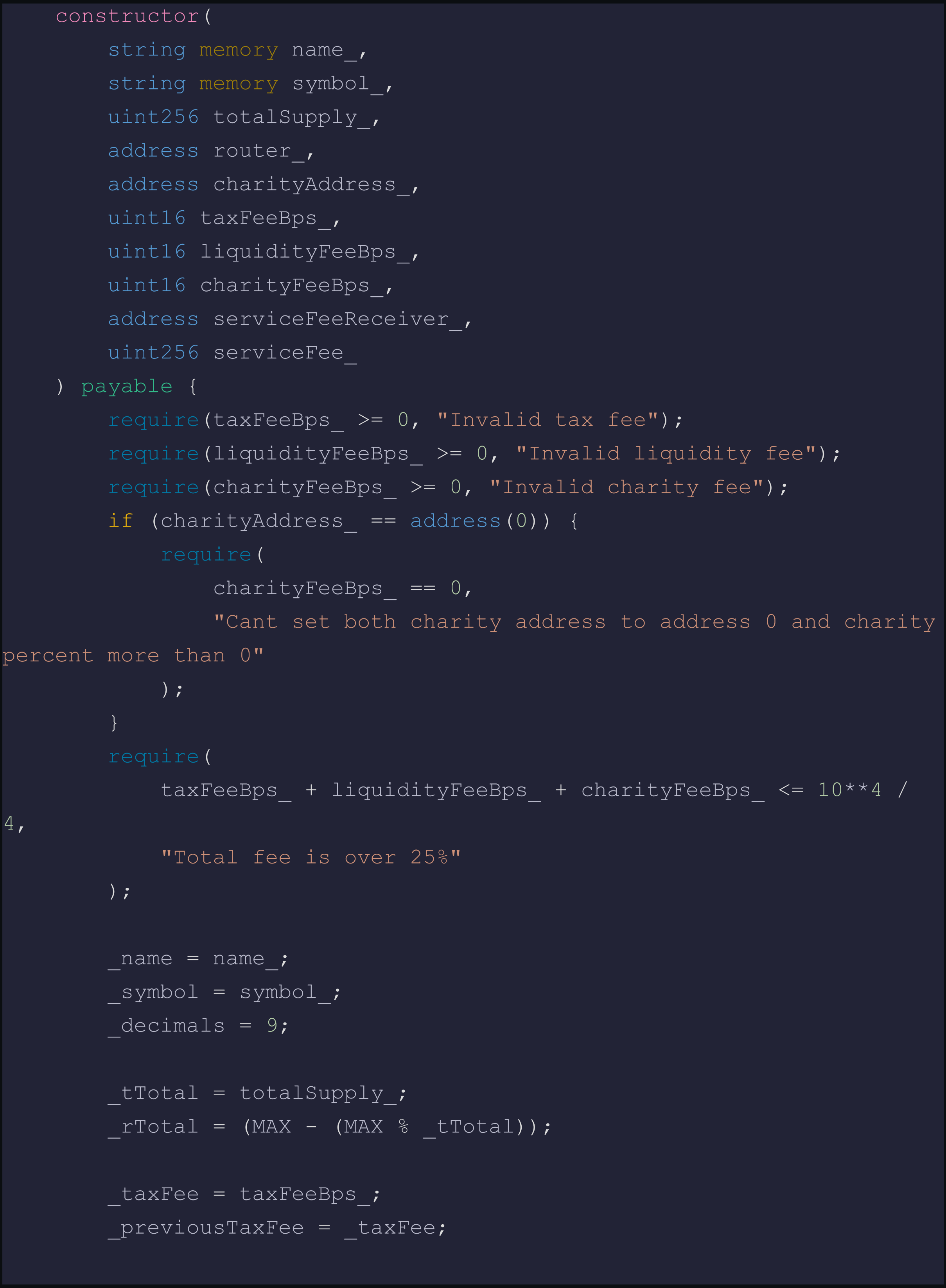
Extra notes by the team

🟢 Fees cannot be set higher than 25%

🟡 Owner can exclude from fees

🟡 The ownership of the contract isn't renounced

Contract Snapshot



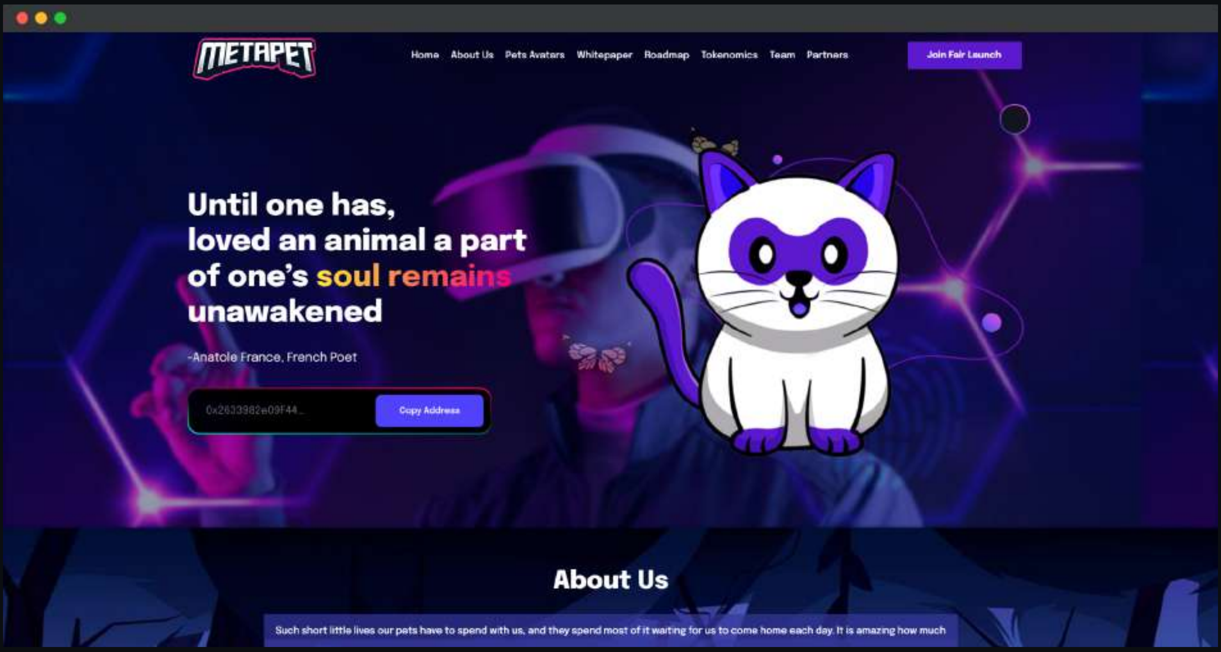
```
    constructor(
        string memory name_,
        string memory symbol_,
        uint256 totalSupply_,
        address router_,
        address charityAddress_,
        uint16 taxFeeBps_,
        uint16 liquidityFeeBps_,
        uint16 charityFeeBps_,
        address serviceFeeReceiver_,
        uint256 serviceFee_
    ) payable {
        require(taxFeeBps_ >= 0, "Invalid tax fee");
        require(liquidityFeeBps_ >= 0, "Invalid liquidity fee");
        require(charityFeeBps_ >= 0, "Invalid charity fee");
        if (charityAddress_ == address(0)) {
            require(
                charityFeeBps_ == 0,
                "Cant set both charity address to address 0 and charity
percent more than 0"
            );
        }
        require(
            taxFeeBps_ + liquidityFeeBps_ + charityFeeBps_ <= 10**4 /
4,
            "Total fee is over 25%"
        );

        _name = name_;
        _symbol = symbol_;
        _decimals = 9;

        _tTotal = totalSupply_;
        _rTotal = (MAX - (MAX % _tTotal));

        _taxFee = taxFeeBps_;
        _previousTaxFee = _taxFee;
```

Website Review



Until one has, loved an animal a part of one’s soul remains unawakened

-Anatole France, French Poet

About Us

Such short little lives our pets have to spend with us, and they spend most of it waiting for us to come home each day. It is amazing how much

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 liquidity yet)

🟡 Large unlocked wallets (tokens not distributed yet)

🟡 No doxxed Team (yet)

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

🟢 Correct router cannot be changed

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.