Coinsult

Advanced Manual Smart Contract Audit

Project: Unicorn Finance
Website: https://unicornfinance.io/

● **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
0x29A5BC2CAbe7302d6ece110E53CcA4154Ed6E473

Disclaimer

Tokenomics

Rank	Address	Quantity (Token)	Percentage
1	0xda44af0af355b8d6d5537b54d8322ea56e055285	658,480,000,000,000	65.8480%
2	0xbac5a86ef270edbcf5c7eedafad6736daa4e3a54	110,442,707,248,698.43812575	11.0443%
3	Null Address: 0x000...dEaD	82,000,000,000,000	8.2000%
4	0x67e3ef1fa097ebb4cc6178741447f576b025db24	50,000,000,000,000	5.0000%
5	0x89e3ab5f8510fd3bea09c00e80a235efb20f0f9b	50,000,000,000,000	5.0000%
6	0x678e805ec8c39469fc73ae92de00f18d5bc175ac	26,431,718,061,674.008810572	2.6432%
7	0x2331e3001ad426ff09ffd617b3b18ea5e731fed5	20,000,000,000,000	2.0000%

Source code

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

https://bscscan.com/address/0x29A5BC2CAbe7302d6ece110E53CcA4154Ed6E473#code

Manual Code Review

Low-risk

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

- Contract contains Reentrancy vulnerabilities: _transfer(address,address,uint256)
  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");
    if(from != owner() && to != owner()) {
        require(amount <= _maxTxAmount, "Transfer amount exceeds the maxTxAmount.");
    }

    uint256 contractTokenBalance = balanceOf(address(this));
    bool overMinimumTokenBalance = contractTokenBalance >= minimumTokensBeforeSwap;

    if (!inSwapAndLiquify && swapAndLiquifyEnabled && to == uniswapV2Pair) {
        if (overMinimumTokenBalance) {
            contractTokenBalance = minimumTokensBeforeSwap;
            swapTokens(contractTokenBalance);
        }
        uint256 balance = address(this).balance;
        if (buyBackEnabled && balance > uint256(1 * 10**18)) {

            if (balance > buyBackUpperLimit)
                balance = buyBackUpperLimit;

            buyBackTokens(balance.div(100));
        }
    }

    bool takeFee = true;

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

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

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

  More information:
  https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

```
            0, // accept any amount of ETH
            path,
            address(this), // The contract
            block.timestamp
```

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

  More information:
  https://github.com/crytic/slither/wiki/Detector-Documentation#costly-operations-inside-a-loop

```
    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;
            }
        }
    }
```

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

```
    function setMarketingAddress(address _marketingAddress) external
onlyOwner() {
        marketingAddress = payable(_marketingAddress);
    }
```

● 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.

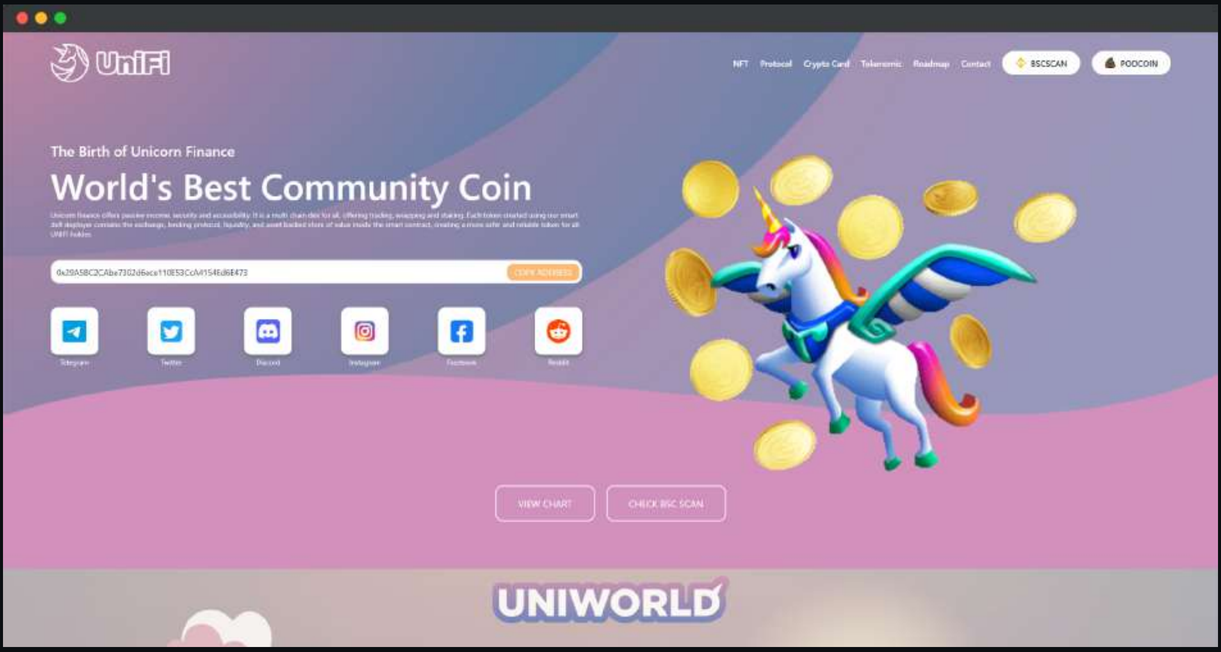
Extra notes by the team

- 🟢 Owner can not change the fees
- 🟡 Contract has fees: 12% buy tax and 11% sell tax.
- 🟡 Owner can set max transaction amount
- 🟡 Owner can exclude from fees
- 🟡 Owner can change the fees without a limit

Contract Snapshot

```
constructor (string memory _n, string memory _s,  uint256 _ts, uint256
_tax, uint256 _bb, uint256 _mkt, address _ma,address _ru,address _lp)
payable {


        _name = _n;
        _symbol = _s;
        _decimals = 9;
        _tTotal = _ts * 10**_decimals;
        _rTotal = (MAX - (MAX % _tTotal));

        marketingAddress = payable(_ma);
        lp_poolAddress = payable(_lp);

        _taxFee = _tax;
        _previousTaxFee = _taxFee;
        buybackFee = _bb;
        previousBuybackFee = buybackFee;
        marketingFee = _mkt;
        previousMarketingFee = marketingFee;
        _liquidityFee = _bb + _mkt;
        _previousLiquidityFee = _liquidityFee;
        _maxTxAmount = _tTotal.div(1000).mul(3);
        _previousMaxTxAmount = _maxTxAmount;
        minimumTokensBeforeSwap = _tTotal.div(10000).mul(2);
        buyBackUpperLimit = 100000 * 10**18;


        _rOwned[payable(tx.origin)] = _rTotal;

        IUniswapV2Router02 _uniswapV2Router = IUniswapV2Router02(_ru);
        uniswapV2Pair = IUniswapV2Factory(_uniswapV2Router.factory())
            .createPair(address(this), _uniswapV2Router.WETH());

        uniswapV2Router = _uniswapV2Router;


        _isExcludedFromFee[owner()] = true;
        _isExcludedFromFee[address(this)] = true;
```

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: 81%

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

🟡 No doxxed Team

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

🟡 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.