Coinsult

Advanced Manual Smart Contract Audit

Project: Easter Shiba Inu
Website: https://eastershibainu.xyz/

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
0x1D2AFeE2D60dbD4C457AeE09F477a3B5ee0b605D

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: 76,000,000
Total Holders: 2
Top 10 holders:

Rank	Address	Quantity (Token)	Percentage
1	0x01a5435500808a9c585d2961ae914c1faa32b4db	49,122,000	64.6342%
2	0x40d651b41688d0dd884b3f102b6ea7d9119f6d4d	26,878,000	35.3658%

The top 100 holders collectively own 100.00% (41,200,000.00 Tokens) of Eater Shiba Inu

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

Source code

Coinsult was commissioned by Easter Shiba Inu to perform an audit based on the following smart contract:

https://bscscan.com/address/0x1d2afee2d60dbd4c457aee09f477a3b5ee0b605d#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");

        // 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);
        }
```

- Contradiction in the contract.
  Fix the incorrect comparison by changing the value type or the comparison. More information: Slither

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

        require(charityFeeBps_ >= 0, "Invalid charity fee");
```

- Use of non-alpanumeric thicker

```
        $ESI
```

- Function which sends eth to arbitrary destination
  Ensure that an arbitrary user cannot withdraw unauthorized funds. More information: Slither

```
    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
        );
    }
```

🟡 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 change fees but not more than 25%.

🟡 A lot of commented code is in the contract, this could be removed to increase the readability.

🟡 The ownership is not renounced.

🟡 Owner can whitelist addresses from fee.

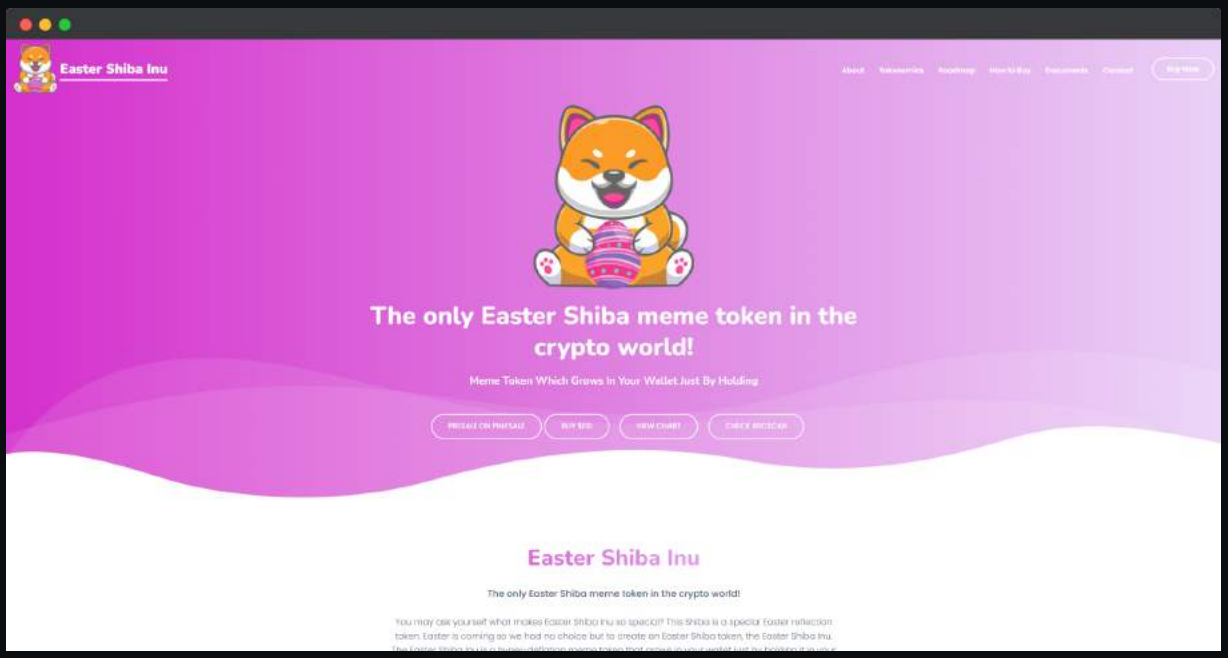
🟡 No liquidity router hard coded in the contract.

Contract Snapshot

```
contract LiquidityGeneratorToken is IERC20, Ownable, BaseToken {
    using SafeMath for uint256;
    using Address for address;

    uint256 public constant VERSION = 1;

    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;
    uint256 private _rTotal;
    uint256 private _tFeeTotal;

    string private _name;
    string private _symbol;
    uint8 private _decimals;

    uint256 public _taxFee;
    uint256 private _previousTaxFee = _taxFee;

    uint256 public _liquidityFee;
    uint256 private _previousLiquidityFee = _liquidityFee;

    uint256 public _charityFee;
    uint256 private _previousCharityFee = _charityFee;

    IUniswapV2Router02 public uniswapV2Router;
    address public uniswapV2Pair;
    address public _charityAddress;

    bool inSwapAndLiquify;
    bool public swapAndLiquifyEnabled;
```

```
    uint256 private numTokensSellToAddToLiquidity;

    event MinTokensBeforeSwapUpdated(uint256 minTokensBeforeSwap);
    event SwapAndLiquifyEnabledUpdated(bool enabled);
    event SwapAndLiquify(
        uint256 tokensSwapped,
        uint256 ethReceived,
        uint256 tokensIntoLiqudity
    );
```

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
- 🟡 Some minor spelling errors

Loading speed: 94%

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

🔴 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

🟡 No router 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.