Coinsult

Advanced Manual Smart Contract Audit

Project: SN
Website: - not applicable

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

Contract address
0xb3d728c6D10c1324ef4C3D3DE11c6B465d5C33D6

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	0x1858486b6bdb8f41e0281b2064b0be963f655273	3,474.193829905331329376	39.0886%
2	Null Address: 0x000...dEaD	1,374.919660023265591261	15.4694%
3	0x1c83bad820ad172ebb6fccd941b750d5bc1b5324	89.466179001997115488	1.0066%
4	0x6cdae4d21d89413fba8fa60356e1cbea44d04a48	56.682297073835323092	0.6377%
5	0x6e3cc5c0a5e5162e242cfaa0430dee1500c81d9e	49.200094843984382309	0.5536%

Source code

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

https://bscscan.com/address/0xb3d728c6d10c1324ef4c3d3de11c6b465d5c33d6#code

Manual Code Review

🟢 Low-risk

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

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

```
    function swapAndLiquify(uint256 contractTokenBalance) public
lockTheSwap {
        if (IERC20(fistToken).balanceOf(address(this)) >= 10) {
            IERC20(fistToken).transfer(address(devAddress),
IERC20(fistToken).balanceOf(address(this)));
        }
        uint256 addHl =
uint256(100).mul(_liquidityFee).div(_liquidityFee.add(_devFee));
        uint256 addNumber =
contractTokenBalance.mul(addHl).div(100).div(2);
        uint256 devNumber = contractTokenBalance.sub(addNumber);
        uint256 initialBalance =
IERC20(fistToken).balanceOf(address(this));
        swapTokensForDividendToken(devNumber);
        uint256 newBalance
=IERC20(fistToken).balanceOf(address(this)).sub(initialBalance);
        uint256 addFistNumber = newBalance.div(7);
        addLiquidity(addNumber,addFistNumber);
        emit SwapAndLiquify(addNumber, addFistNumber, addNumber);
    }
```

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

```
block.timestamp.add(30)
```

🟡 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 exclude addresses from fees.
- Fees can be set up to 100% for both buy and sell fees.
- Owner can set a max transaction amount.
- The ownership of the contract isn’t renounced.

Contract Snapshot

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

    IWrap public wrap;

    mapping (address => uint256) private _rOwned;
    mapping (address => uint256) private _tOwned;
    mapping (address => mapping (address => uint256)) private
_allowances;

    mapping (address => bool) private _isExcludedFromFee;

    uint256 private constant MAX = ~uint256(0);
    uint256 private _tTotal = 8888 * 10**18;
    uint256 private _rTotal = (MAX - (MAX % _tTotal));
    uint256 private _tFeeTotal;

    string private _name = "Shen Nong Token";
    string private _symbol = "SN";
    uint8 private _decimals = 18;

    uint256 public _taxFee = 0;
    uint256 private _previousTaxFee = _taxFee;

    uint256 public _liquidityFee = 1;
    uint256 private _previousLiquidityFee = _liquidityFee;

    uint256 public _burnFee = 1;
    uint256 private _previousBurnFee = _burnFee;

    uint256 public _devFee = 3;
    uint256 private _previousDevFee = _devFee;
```

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.

- not applicable

Rug-pull Review

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

🟢 Locked Liquidity

🟢 No 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 can not 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.