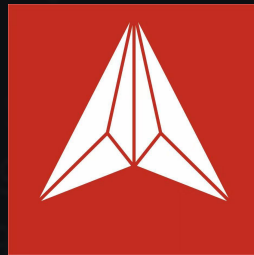




Coinsult

Advanced Manual Smart Contract Audit



Project: ACCAudit

Website: -

Low-risk

5 low-risk code
issues found

Medium-risk

0 medium-risk code
issues found

High-risk

0 high-risk code
issues found

Contract address

0x5dc53496e8dd50887785d75d432cba6a86f82cad

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	0xa7a86c2393cebd9bc9600378acf7a1faddf8f683	2,370,000	30.0000%
2	0x639e461b391da033058e942acc1343e7829b67b9	1,185,000	15.0000%
3	0x4747a90fb23668c2bd332df47ec0c28530d89fb9	1,110,352	14.0551%
4	0x12abcdaf2bb92252f6a6d9f55601143656921452	1,085,000	13.7342%
5	0x97fae71dce0dac6fbeeaa855aa67643af46d6aabb	395,000	5.0000%
6	0xb61eb71a492f47a5fd8e927081526abc2211f9ca	395,000	5.0000%
7	PancakeSwap V2: BSC-USD-ACC 3	294,366.296659682339183157	3.7262%
8	0x5a320171c1524d1519b2d16cc86582be49b3cb0d	59,314.277682327810864632	0.7508%
9	0x5dc53496e8dd50887785d75d432cba6a86f82cad	47,043.215034912638356975	0.5955%
10	Null Address: 0x000...dEaD	33,679.996439655145236458	0.4263%

Source code

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

<https://bscscan.com/address/0x5dc53496e8dd50887785d75d432cba6a86f82cad#code>

Manual Code Review

● Low-risk

5 low-risk code issues found.

Could be fixed, will not bring problems.

- isContractSwitch is created but never changed so it can be removed from the function

```
bool public isContractSwitch;  
  
    if (isContractSwitch && _isContract(from)) {  
        require(whiteContract[msg.sender], "not allowed");  
    }
```

- Uninitialized state variables

Initialize all the variables. If a variable is meant to be initialized to true, explicitly set it to true to improve code readability.

```
bool public isContractSwitch;  
bool public isFee;  
bool public isExcludedEx;  
bool public isSell;  
bool public isBuy;
```

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

```
function swapTokensForUSDT(uint256 tokenAmount,address to) private
{

    address[] memory path = new address[] (2);
    path[0] = address(this);
    path[1] = USDT;

    _approve(address(this), address(uniswapV2Router), tokenAmount);

    uniswapV2Router.swapExactTokensForTokensSupportingFeeOnTransferTokens(
        tokenAmount,
        0,
        path,
        to,
        block.timestamp
    );
}
```

- Instead of initializing a new var 'takeFee', use `isFee` since it is already false

```
if(!isFee){
    takeFee = false;
}

FeeParam memory param;
if(takeFee) {
```

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

```
uint256 public _maxTxAmount = 100000 * 10**18;
```

● **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 changed
- Owner can exclude addresses from fees.
- The ownership of the contract isn't renounced.
- Owner can blacklist

```
function setBlack(address addr_, bool com_) public onlyOwner {  
    black[addr_] = com_;  
}
```

- Owner can forbid buying and selling.

```
    if(isFee && exPairs[from]){  
        param.user = to;  
        if(!_isExcludedFromFee[param.user]){  
            require(!isBuy, "forbid buying");  
        }  
        _getBuyParam(amount,param);  
    }  
    if(isFee && exPairs[to]){  
        param.user = from;  
        if(!_isExcludedFromFee[param.user]){  
            require(!isSell, "forbid selling");  
        }  
        _getSellParam(amount,param);  
    }  
}
```

Contract Snapshot

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

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

    mapping (address => bool) private _isExcludedFromFee;
    mapping(address => bool) public whiteContract;

    uint8 private _decimals = 18;
    uint256 private _tTotal = 7900000 * 10**18;

    string private _name = "ACCToken";
    string private _symbol = "ACC";

    uint256 public totalBuyFee = 20;
    uint256 public totalSellFee = 10;
    uint256 public transferFee = 10;
    bool public isContractSwitch;
    bool public isFee;
    bool public isExcludedEx;
    bool public isSell;
    bool public isBuy;

    IUniswapV2Router02 public immutable uniswapV2Router;
    address public uniswapV2Pair;

    mapping(address => bool) public exPairs;

    bool inSwapAndLiquify;

    uint256 public _maxTxAmount = 100000 * 10**18;
```