Coinsult

Advanced Manual Smart Contract Audit

Project: smart ocean

Website: No 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

0x1472d3c58eD3Ea6BDBBE3517e1632299951ce2E1

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	0xa4c652033ae6e03261e87a0584d7928ece6e53af	42,000,001	46.1538%
2	PancakeSwap V2: SOT 53	16,013,378.068411088628946266	17.5971%
3	0xc87ed85229a95817d7c4339abbf1845cb3936a86	6,909,046.73452256008477798	7.5924%
4	0x04f013114e066cf5932cf4ae310f9bd0db884f02	5,000,000	5.4945%
5	0xe3f9c02c2097c480cadf8d6f97d848bc6d5400d7	3,163,804.510149005845971663	3.4767%

Source Code

Coinsult was comissioned by smart ocean to perform an audit based on the following smart contract:

https://bscscan.com/address/0x1472d3c58ed3ea6bdbbe3517e1632299951ce2e1#code

Manual Code Review

In this audit report we will highlight all these issues:

🟢 Low-Risk

5 low-risk code issues found

🟡 Medium-Risk

0 medium-risk code issues found

🔴 High-Risk

0 high-risk code issues found

The detailed report continues on the next page...

🟢 **Low-Risk:** Could be fixed, will not bring problems.

Avoid relying on block.timestamp

block.timestamp can be manipulated by miners.

```
if(startTimeForSwap &gt; block.timestamp &amp;&amp; (to == address(uniswapV2Pair) || from == address
```

Recommendation

Do not use `block.timestamp`, now or `blockhash` as a source of randomness

Exploit scenario

```
contract Game {

    uint reward_determining_number;

    function guessing() external{
      reward_determining_number = uint256(block.blockhash(10000)) % 10;
    }
}
```

Eve is a miner. Eve calls `guessing` and re-orders the block containing the transaction. As a result, Eve wins the game.

🟢 **Low-Risk:** Could be fixed, will not bring problems.

Too many digits

Literals with many digits are difficult to read and review.

```
_mint(msg.sender, 91000000 * 1e18);
```

Recommendation

Use: Ether suffix, Time suffix, or The scientific notation

Exploit scenario

```
contract MyContract{
    uint 1_ether = 10000000000000000000;
}
```

While `1_ether` looks like `1 ether`, it is `10 ether`. As a result, it's likely to be used incorrectly.

🟢 **Low-Risk:** Could be fixed, will not bring problems.

No zero address validation for some functions

Detect missing zero address validation.

```
function setInvertDate(address b) public onlyOwner {
    _invertData = b;
}
```

Recommendation

Check that the new address is not zero.

Exploit scenario

```
contract C {

  modifier onlyAdmin {
    if (msg.sender != owner) throw;
    _;
  }

  function updateOwner(address newOwner) onlyAdmin external {
    owner = newOwner;
  }
}
```

Bob calls `updateOwner` without specifying the `newOwner`, soBob loses ownership of the contract.

● **Low-Risk:** Could be fixed, will not bring problems.

Missing events arithmetic

Detect missing events for critical arithmetic parameters.

```
function setThreshold(uint256 value) public onlyOwner {
    _threshold = value;
}
```

Recommendation

Emit an event for critical parameter changes.

Exploit scenario

```
contract C {

  modifier onlyAdmin {
    if (msg.sender != owner) throw;
    _;
  }

  function updateOwner(address newOwner) onlyAdmin external {
    owner = newOwner;
  }
}
```

`updateOwner()` has no event, so it is difficult to track off-chain changes in the buy price.

🟢 **Low-Risk:** Could be fixed, will not bring problems.

Conformance to Solidity naming conventions

Allow _ at the beginning of the mixed_case match for private variables and unused parameters.

```
uint256[] public _rate=[20,10,10,5,5];
```

Recommendation

Follow the Solidity naming convention.

Rule exceptions

- Allow constant variable name/symbol/decimals to be lowercase (`ERC20`).
- Allow _ at the beginning of the `mixed_case` match for private variables and unused parameters.

Owner privileges

- Owner can change max transaction amount
- Owner can set fees higher than 25%
- Owner can exclude from fees
- Owner can pause the contract

Extra notes by the team

No notes

Contract Snapshot

```
contract SO is ERC20, Ownable {
using SafeMath for uint256;
using Address for address;

address public _router;
IUniswapV2Router02 public uniswapV2Router;
IUniswapV2Pair public uniswapV2Pair;
uint256 public startTimeForSwap = 1650882660;

address private burnWallet;
address private poolWallet;
address private technologyWallet;
address private marketingWallet;
address private collectWallet;
address private shareholderWallet = 0x1462f536EDC09606883E1Ec38bDAb18044fD829B;

uint256 private shareholder;
uint256 private burn;
uint256 private pool;
uint256 private technology;
uint256 private marketing;
uint256 private tax;
uint256 private resLimit;
```

Project Overview

● Not KYC verified by Coinsult