OXSCANS

Project Address

0x2ebf0710c8d875d1e9a33a52aabd6c85b406705f

01/26/2024

AUDIT REPORT

SAFETY SCORE: 84

1 - Arbitrary Jump/Storage Write

Result: Pass

2 - Centralization of Control

Result: Medium

Details: The contract contains functions that centralize control, such as `renounceOwnership`, `transferOwnership`, and `flagBot`. The `renounceOwnership` function allows the owner to renounce their ownership, which could lead to a loss of control over the contract. The `transferOwnership` function allows the owner to transfer control to a new address. The `flagBot` function allows the owner to flag addresses as bots, which could be misused to flag honest users.

Code:

```
function renounceOwnership(bool confirmRenounce) external virtual onlyOwner {
    require(confirmRenounce, "Please confirm renounce!");
    emit OwnershipTransferred(_owner, address(0));
    _owner = address(0);
}

function transferOwnership(address newOwner) public virtual onlyOwner {
    require(newOwner != address(0), "Ownable: new owner is the zero address");
    emit OwnershipTransferred(_owner, newOwner);
    _owner = newOwner;
}

function flagBot(address wallet) external onlyOwner {
    require(!flaggedAsBot[wallet], "Wallet is already flagged.");
    flaggedAsBot[wallet] = true;
}
```

Correction:

```
// No correction needed for renounceOwnership and transferOwnership as they are common patterns for ownership control.
// For flagBot, consider implementing a decentralized governance mechanism for flagging bots.
```

3 - Compiler Issues

Result: Pass

4 - Delegate Call to Untrusted Contract

Result: Pass

5 - Dependence on Predictable Variables

Result: Pass

6 - Ether/Token Theft

Result: Pass

7 - Flash Loans

Result: Pass

8 - Front Running

Result: Pass

9 - Improper Events

Result: Pass

10 - Improper Authorization Scheme

Result: Pass

11 - Integer Over/Underflow

Result: Pass

12 - Logical Issues

Result: Medium

Details: The `swapBack` function does not check for the success of the `swapTokensForEth` and `addLiquidity` functions, which could lead to unexpected behavior if these functions fail.

Code:

```
function swapBack() private {
    ...
    swapTokensForEth(contractBalance - liquidityTokens);
    ...
```

```
    addLiquidity(liquidityTokens, ethForLiquidity);
    ...
}
```

Correction:

```
function swapBack() private {
    ...
    bool swapSuccess = swapTokensForEth(contractBalance - liquidityTokens);
    require(swapSuccess, "Swap failed");
    ...
       (uint256 amountToken, uint256 amountETH, uint256 liquidity) = addLiquidity(liquidityTokens,
ethForLiquidity);
    require(liquidity > 0, "Add liquidity failed");
    ...
}
```

13 - Oracle Issues

Result: Pass

14 - Outdated Compiler Version

Result: Pass

15 - Race Conditions

Result: Pass

16 - Reentrancy

Result: Pass

17 - Signature Issues

Result: Pass

18 - Sybil Attack

Result: Pass

19 - Unbounded Loops

Result: Pass

20 - Unused Code

Result: Low

Details: The contract contains unused code, such as the `decimals` function in the `ERC20` contract, which always returns 18. This function could be removed to save gas if the token will always have 18 decimals.

Code:

```
function decimals() public view virtual override returns (uint8) {
    return 18;
}
```

Correction:

```
// Remove the decimals function if the token will always have 18 decimals.
```

LEGAL DISCLAIMER

Oxscans operates as an automated system for smart contract due diligence, acknowledging the possibility of bugs or vulnerabilities impacting token values. We do not hold specific obligations regarding your trading outcomes or the utilization of audit content. Users release Oxscans from any liability associated with content obtained through the tool.

AI GENERATED BY OXSCANS AI TECHNOLOGY

chat with us

Telegram

For more information. Visit below:

Twitter

Github