MythX

REPORT 6068858FB857B00011F1971F

Created	Sat Apr 03 2021 15:11:11 GMT+0000 (Coordinated Universal Time)
Number of analyses	1
User	admin@kobito.finance

REPORT SUMMARY

Analyses ID	Main source file	Detected vulnerabilities
c76fa84b-932e-4c1f-89c6-c8171ddd052d	contracts/KobitoReferral.sol	6

Analysis c76fa84b-932e-4c1f-89c6-c8171ddd052d MythX

Started	Sat Apr 03 2021 15:11:19 GMT+0000 (Coordinated Universal Time)
Finished	Sat Apr 03 2021 15:26:36 GMT+0000 (Coordinated Universal Time)
Mode	Standard
Client Tool	Remythx
Main Source File	Contracts/KobitoReferral.Sol

DETECTED VULNERABILITIES

HIGH	MEDIUM	LOW
0	4	2

ISSUES

MEDIUM

SWC-000

Function could be marked as external.

The function definition of "recordReferral" is marked "public". However, it is never directly called by another function in the same contract or in any of its descendants. Consider to mark it as "external" instead.

Source file

contracts/libs/IBEP20.sol

Locations

```
* @dev Returns the amount of tokens owned by `account`.
*/
function balanceOf(address account) external view returns (uint256);

/**
* @dev Moves `amount` tokens from the caller's account to `recipient`.
*
* Returns a boolean value indicating whether the operation succeeded.
*
* Emits a {Transfer} event.
*/
function transfer(address recipient, uint256 amount) external returns (bool);

/**
* @dev Returns the remaining number of tokens that `spender` will be
* allowed to spend on behalf of `owner` through {transferFrom}. This is
* zero by default.
*
```

MEDIUM

SWC-000

Function could be marked as external.

The function definition of "getReferrer" is marked "public". However, it is never directly called by another function in the same contract or in any of its descendants. Consider to mark it as "external" instead.

Source file

`contracts/libs/IBEP20.sol`

Locations

```
/**
* @dev Returns the remaining number of tokens that `spender` will be
* allowed to spend on behalf of `owner` through {transferFrom}. This is
* zero by default.
*
* This value changes when {approve} or {transferFrom} are called.
*/
function allowance(address _owner, address spender) external view returns (uint256);
```

LOW

SWC-123

Requirement violation.

A requirement was violated in a nested call and the call was reverted as a result. Make sure valid inputs are provided to the nested call (for instance, via passed arguments).

Source file

`contracts/libs/IBEP20.sol`

Locations

```
* @dev Returns the amount of tokens in existence.
*/
function totalSupply() external view returns (uint256);

/**
* @dev Returns the token decimals.
*/
function decimals() external view returns (uint8);

/**
* @dev Returns the token symbol.
*/
function symbol() external view returns (string memory);

/**
* @dev Returns the token name.
*/
function name() external view returns (string memory);

/**
* @dev Returns the bep token owner.
*/
function getOwner() external view returns (address);

/**
* @dev Returns the amount of tokens owned by `account`.
*/
function balanceOf(address account) external view returns (uint256);

/**
* @dev Moves `amount` tokens from the caller's account to `recipient`.
*
* Returns a boolean value indicating whether the operation succeeded.
*
* Emits a {Transfer} event.
*/
function transfer(address recipient, uint256 amount) external returns (bool);

/**
* @dev Returns the remaining number of tokens that `spender` will be
* allowed to spend on behalf of `owner` through {transferFrom}. This is
* zero by default.
*
* This value changes when {approve} or {transferFrom} are called.
*/
function allowance(address _owner, address spender) external view returns (uint256);

/**
* @dev Sets `amount` as the allowance of `spender` over the caller's tokens.
*
* Returns a boolean value indicating whether the operation succeeded.
*
* IMPORTANT: Beware that changing an allowance with this method brings the risk
* that someone may use both the old and the new allowance by unfortunate

* transaction ordering. One possible solution to mitigate this race
* condition is to first reduce the spender's allowance to 0 and set the
```