MythX

REPORT 60B5EFCCF1A91F0018924DE6

Created	Tue Jun 01 2021 08:29:00 GMT+0000 (Coordinated Universal Time)
Number of analyses	1
User	5d48145d459fdc0011c58cba

REPORT SUMMARY

Analyses ID	Main source file	Detected vulnerabilities
27dfddd5-60d2-4286-a326-7d5d77a493b4	PolsStake_flat.sol	14

Analysis 27dfddd5-60d2-4286-a326-7d5d77a493b4 MythX

Started

Finished Tue Jun 01 2021 08:29:00 GMT+0000 (Coordinated Universal Time)

Mode Quick

Client Tool Mythx-Cli-0.6.22

Main Source File PolsStake_flat.Sol

DETECTED VULNERABILITIES

HIGH	MEDIUM	LOW
0	4	10

ISSUES

MEDIUM SWC-000 Function could be marked as external.

The function definition of "owner" 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

PolsStake_flat.sol

Locations

```
* @dev Returns the address of the current owner.
*/
function owner() public view virtual returns (address) {
return _owner;
}

/**
```

MEDIUM SWC-000 Function could be marked as external.

The function definition of "userClaimableRewardToken_msgSender" 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

PolsStake_flat.sol

Locations

```
}

function userClaimableRewardToken_msgSender() public view returns (uint256) {
return userClaimableRewardToken(msg.sender);
}

function userStakedTokenUnlockTime_msgSender() public view returns (uint256 unlockTime) {
```

MEDIUM

SWC-000

Function could be marked as external.

The function definition of "userStakedTokenUnlockTime_msgSender" 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

PolsStake_flat.sol

Locations

```
}

function userStakedTokenUnlockTime_msgSender() public view returns (uint256 unlockTime) {
return userStakedTokenUnlockTime(msg.sender);
}

/** public external view functions (also used internally) **************************/
```

MEDIUM

SWC-000

Function could be marked as external.

The function definition of "approveStakingToken" 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

PolsStake_flat.sol

Locations

```
* https://solidity-by-example.org/delegatecall/
*/
function approveStakingToken(uint256 _amount) public returns (bool) {
(bool success, ) =
address(stakingToken).delegatecall(
abi.encodeWithSignature("approve(address,uint256)", address(this), _amount)
);
return success;
}

/**
```

LOW

SWC-103

A floating pragma is set.

The current pragma Solidity directive is ""^0.8.0"". It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

Source file

PolsStake_flat.sol

Locations

```
// SPDX-License-Identifier: MIT

pragma solidity ^0.8.0;

/*
```

LOW

SWC-103

A floating pragma is set.

The current pragma Solidity directive is ""^0.8.0"". It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

Source file

PolsStake_flat.sol

Locations

```


pragma solidity ^0.8.0;

/**
```

LOW

SWC-103

A floating pragma is set.

The current pragma Solidity directive is ""^0.8.0"". It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

Source file

PolsStake_flat.sol

Locations

```


pragma solidity ^0.8.0;

// CAUTION
```

LOW

SWC-103

A floating pragma is set.

The current pragma Solidity directive is ""^0.8.0"". It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

Source file

PolsStake_flat.sol

Locations

```


pragma solidity ^0.8.0;

/**
```

LOW

SWC-103

A floating pragma is set.

The current pragma Solidity directive is ""^0.8.0"". It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

Source file

`PolsStake_flat.sol`

Locations

```
392
393
394  pragma solidity ^0.8.0;
395
396  /**
```

LOW

SWC-103

A floating pragma is set.

The current pragma Solidity directive is ""^0.8.0"". It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

Source file

`PolsStake_flat.sol`

Locations

```
473
474
475  pragma solidity ^0.8.0;
476
477  /**
```

LOW

SWC-103

A floating pragma is set.

The current pragma Solidity directive is ""^0.8.0"". It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

Source file

`PolsStake_flat.sol`

Locations

```
666
667
668  pragma solidity ^0.8.0;
669
```

LOW

SWC-103

A floating pragma is set.

The current pragma Solidity directive is ""^0.8.0"". It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

Source file

PolsStake_flat.sol

Locations

```


pragma solidity ^0.8.0;

/**
```

LOW

SWC-103

A floating pragma is set.

The current pragma Solidity directive is "">=0.7.6<0.9.0"". It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

Source file

PolsStake_flat.sol

Locations

```


pragma solidity >=0.7.6 <0.9.0;

// import "@openzeppelin/contracts/token/ERC20/IERC20.sol";
```

LOW

SWC-103

A floating pragma is set.

The current pragma Solidity directive is ""^0.8.0"". It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

Source file

PolsStake_flat.sol

Locations

```


pragma solidity ^0.8.0;

// import "@openzeppelin/contracts/utils/math/SafeCast.sol"; // OZ contracts v4
```