MythX

REPORT 64F7861A96EA2500194A491E

Created	Tue Sep 05 2023 19:48:42 GMT+0000 (Coordinated Universal Time)
Number of analyses	1
User	62b1a8425ec4948f52c83856

REPORT SUMMARY

Analyses ID	Main source file	Detected vulnerabilities
45e99a61-7179-4635-86aa-5c5f22ff71bb	/contracts/erc721nesdrop.sol	5

Analysis 45e99a61-7179-4635-86aa-5c5f22ff71bb

MythX

Started	Tue Sep 05 2023 19:48:46 GMT+0000 (Coordinated Universal Time)
Finished	Tue Sep 05 2023 20:04:33 GMT+0000 (Coordinated Universal Time)
Mode	Standard
Client Tool	Mythx-Vscode-Extension
Main Source File	/Contracts/Erc721nesdrop.Sol

DETECTED VULNERABILITIES

HIGH	MEDIUM	LOW
0	0	5

ISSUES

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

/contracts/erc721nesdrop.sol

Locations

```
// SPDX-License-Identifier: Apache-2.0
pragma solidity ^0.8.0;

import "@thirdweb-dev/contracts/custom/ERC721Drop.sol";
```

LOW SWC-108 State variable visibility is not set.

It is best practice to set the visibility of state variables explicitly. The default visibility for "multiplier" is internal. Other possible visibility settings are public and private.

Source file

/contracts/erc721nesdrop.sol

Locations

```

contract ERC721NESDrop is ERC721Drop {
uint256 multiplier;

// Event published when a token is staked.
```

LOW

SWC-120

Potential use of "block.number" as source of randonmness.

The environment variable "block.number" looks like it might be used as a source of randomness. Note that the values of variables like coinbase, gaslimit, block number and timestamp are predictable and can be manipulated by a malicious miner. Also keep in mind that attackers know hashes of earlier blocks. Don't use any of those environment variables as sources of randomness and be aware that use of these variables introduces a certain level of trust into miners.

Source file

`/contracts/erc721nesdrop.sol`

Locations

```
return block.number - tokenToWhenStaked[tokenId];
} else {
return 0;
}
}
```

LOW

SWC-120

Potential use of "block.number" as source of randonmness.

The environment variable "block.number" looks like it might be used as a source of randomness. Note that the values of variables like coinbase, gaslimit, block number and timestamp are predictable and can be manipulated by a malicious miner. Also keep in mind that attackers know hashes of earlier blocks. Don't use any of those environment variables as sources of randomness and be aware that use of these variables introduces a certain level of trust into miners.

Source file

`/contracts/erc721nesdrop.sol`

Locations

```

/**
* @dev Unstakes a token and records the start block number or time stamp.
*/
function unstake(uint256 tokenId) public {
```