

REPORT 64F7861A96EA2500194A491E

Created Tue Sep 05 2023 19:48:42 GMT+0000 (Coordinated Universal Time)

Number of analyses 1

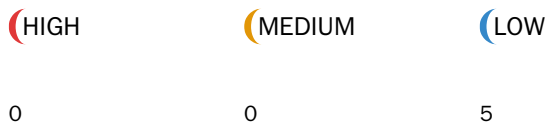
User 62b1a8425ec4948f52c83856

REPORT SUMMARY

Analyses ID	Main source file	Detected vulnerabilities
45e99a61-7179-4635-86aa-5c5f22ff71bb	/contracts/erc721.nesdrop.sol	5

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



ISSUES

LOW

A floating pragma is set.

SWC-103

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

```
1 // SPDX-License-Identifier: Apache-2.0
2 pragma solidity ^0.8.0;
3
4 import "@thirdweb-dev/contracts/custom/ERC721Drop.sol";
```

LOW

State variable visibility is not set.

SWC-108

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

```
5
6 contract ERC721NESDrop is ERC721Drop {
7     uint256 multiplier;
8
9     // Event published when a token is staked.
```

LOW Potential use of "block.number" as source of randomness.

SWC-120

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

```
42 | return block.number - tokenToWhenStaked[tokenId];
43 | } else {
44 |   return 0;
45 | }
46 | }
```

LOW Potential use of "block.number" as source of randomness.

SWC-120

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

```
78 |
79 | /**
80 |  * @dev Unstakes a token and records the start block number or time stamp.
81 |  */
82 | function unstake(uint256 tokenId) public {
```