c35c7bfc1078139b84f7e8ef5a81ee55ea152d39d1d578eef855a03bbf866e93

File: NFTEX.sol | Language:solidity | Size:10670 bytes | Date:2021-04-25T07:20:33.890Z

Critical	High	Medium	Low	Note
0	0	1	1	8

Issues

Severity	Issue	Analyzer	Code Lines
Medium	SWC-102	Achilles	1
Low	SWC-103	Achilles	1
Note	SWC-116	Achilles	84, 151, 162, 182, 203, 220, 243, 288

Code

1. SWC-102 / lines: 1 Medium Achilles

```
A security vulnerability has been detected.

1  pragma solidity ^0.6.0;
2
```

In detail

Using an outdated compiler version can be problematic especially if there are publicly disclosed bugs and issues that affect the current compiler version.

2. SWC-103 / lines: 1 Low Achilles

```
A security vulnerability has been detected.

1  pragma solidity ^0.6.0;
2
```

In detail

Contracts should be deployed with the same compiler version and flags that they have been tested with thoroughly. Locking the pragma helps to ensure that contracts do not accidentally get deployed using, for example, an outdated compiler version that might introduce bugs that affect the contract system negatively.

3. SWC-116 / lines: 84 Note Achilles

```
A security vulnerability has been detected.

83                 (_startPrice - o.endPrice) / (o.endBlock - _startBlock);
84             uint256 tickPrice = (block.number - _startBlock) * tickPerBlock;
85             if (tickPrice >= _startPrice - _endPrice) {
```

In detail

Contracts often need access to the current timestamp to trigger time-dependent events. As Ethereum is decentralized, nodes can synchronize time only to some degree. Moreover, malicious miners can alter the timestamp of their blocks, especially if they can gain advantages by doing so. However, miners can't set timestamp smaller than the previous one (otherwise the block will be rejected), nor can they set the timestamp too far ahead in the future. Taking all of the above into consideration, developers can't rely on the preciseness of the provided timestamp.

4. SWC-116 / lines: 151 Note Achilles

A security vulnerability has been detected.

```
150     ) internal {
151         require(_endBlock > block.number, "Duration must be more than zero");
152
```

In detail

Contracts often need access to the current timestamp to trigger time-dependent events. As Ethereum is decentralized, nodes can synchronize time only to some degree. Moreover, malicious miners can alter the timestamp of their blocks, especially if they can gain advantages by doing so. However, miners can't set timestamp smaller than the previous one (otherwise the block will be rejected), nor can they set the timestamp too far ahead in the future. Taking all of the above into consideration, developers can't rely on the preciseness of the provided timestamp.

5. SWC-116 / lines: 162 Note Achilles

A security vulnerability has been detected.

```
161             _endPrice,
162             block.number,
163             _endBlock,
```

In detail

Contracts often need access to the current timestamp to trigger time-dependent events. As Ethereum is decentralized, nodes can synchronize time only to some degree. Moreover, malicious miners can alter the timestamp of their blocks, especially if they can gain advantages by doing so. However, miners can't set timestamp smaller than the previous one (otherwise the block will be rejected), nor can they set the timestamp too far ahead in the future. Taking all of the above into consideration, developers can't rely on the preciseness of the provided timestamp.

6. SWC-116 / lines: 182 Note Achilles

A security vulnerability has been detected.

```
181     ) internal view returns (bytes32) {
182         return keccak256(abi.encodePacked(block.number, _token, _id, _seller));
183     }
```

In detail

Contracts often need access to the current timestamp to trigger time-dependent events. As Ethereum is decentralized, nodes can synchronize time only to some degree. Moreover, malicious miners can alter the timestamp of their blocks, especially if they can gain advantages by doing so. However, miners can't set timestamp smaller than the previous one (otherwise the block will be rejected), nor can they set the timestamp too far ahead in the future. Taking all of the above into consideration, developers can't rely on the preciseness of the provided timestamp.

7. SWC-116 / lines: 203 Note Achilles

A security vulnerability has been detected.

```
202         require(endBlock != 0, "Canceled order");
203         require(block.number <= endBlock, "It's over");
204         require(o.seller != msg.sender, "Can not bid to your order");
```

In detail

Contracts often need access to the current timestamp to trigger time-dependent events. As Ethereum is decentralized, nodes can synchronize time only to some degree. Moreover, malicious miners can alter the timestamp of their blocks, especially if they can gain advantages by doing so. However, miners can't set timestamp smaller than the previous one (otherwise the block will be rejected), nor can they set the timestamp too far ahead in the future. Taking all of the above into consideration, developers can't rely on the preciseness of the provided timestamp.

8. SWC-116 / lines: 220 Note Achilles

A security vulnerability has been detected.

```
219
220         if (block.number > endBlock - 20) {
221             //20blocks = 5 mins in Etherium.
```

In detail

Contracts often need access to the current timestamp to trigger time-dependent events. As Ethereum is decentralized, nodes can synchronize time only to some degree. Moreover, malicious miners can alter the timestamp of their blocks, especially if they can gain advantages by doing so. However, miners can't set timestamp smaller than the previous one (otherwise the block will be rejected), nor can they set the timestamp too far ahead in the future. Taking all of the above into consideration, developers can't rely on the preciseness of the provided timestamp.

9. SWC-116 / lines: 243 Note Achilles

A security vulnerability has been detected.

```
242         require(endBlock != 0, "Canceled order");
243         require(endBlock > block.number, "It's over");
244         require(o.orderType < 2, "It's a English Auction");
```

In detail

Contracts often need access to the current timestamp to trigger time-dependent events. As Ethereum is decentralized, nodes can synchronize time only to some degree. Moreover, malicious miners can alter the timestamp of their blocks, especially if they can gain advantages by doing so. However, miners can't set timestamp smaller than the previous one (otherwise the block will be rejected), nor can they set the timestamp too far ahead in the future. Taking all of the above into consideration, developers can't rely on the preciseness of the provided timestamp.

10. SWC-116 / lines: 288 Note Achilles

A security vulnerability has been detected.

```
287         require(o.orderType == 2, "This function is for English Auction");
288         require(block.number > o.endBlock, "Not yet");
289
```

In detail

Contracts often need access to the current timestamp to trigger time-dependent events. As Ethereum is decentralized, nodes can synchronize time only to some degree. Moreover, malicious miners can alter the timestamp of their blocks, especially if they can gain advantages by doing so. However, miners can't set timestamp smaller than the previous one (otherwise the block will be rejected), nor can they set the timestamp too far ahead in the future. Taking all of the above into consideration, developers can't rely on the preciseness of the provided timestamp.