SOOHO

ae9e51d0efe7c3a22f7c01fbf3f53d94b76cfddb9f886c52e90f0d4640b43888

File: ConstSwap.sol | Language:solidity | Size:32333 bytes | Date:2022-01-21T11:40:32.527Z

Critical	High	Medium	Low	Note
0	0	0	1	1

Issues

Severity	Issue	Analyzer	Code Lines
Low	SWC-103	Achilles	2
Note	SWC-116	Achilles	451

Code

1. SWC-103 / lines: 2 Low Achilles

A security vulnerability has been detected.

```
1   // SPDX-License-Identifier: MIT
2   pragma solidity ^0.8.0;
3
```

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 Locking the pragma helps to ensure that contracts do not accidentally get deployed using, for example, an outdated compiler version that might introduce bugs that affect the contract system negatively.

2. SWC-116 / lines: 451 Note Achilles

A security vulnerability has been detected.

```
450
451         uint256 blockTimestamp = block.timestamp;
452         uint256 timeElapsed = blockTimestamp.sub(_lastRebalance[tokenIndex]);
```

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.