SOOHO

Critical	High	Medium	Low	Note
0	0	1	1	2

Issues

Severity	Issue	Analyzer	Code Lines
Medium	SWC-102	Achilles	1
Low	SWC-103	Achilles	1
Note	SWC-108	Achilles	21, 22

Code

1. SWC-102 / lines: 1 Medium Achilles

⊖   A security vulnerability has been detected.

```
1    pragma solidity ^0.6.0;
2
```
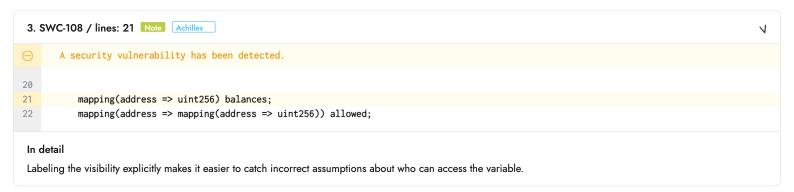
In detail

Using an outdated compiler version can be problematic especially if there are publicly disclosed bugs and issues that affect the current compiler version.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 Locking the pragma helps to ensure that contracts do not accidentally get deployed using, for example, an outdated compiler version that might introduce bugs that affect the contract system negatively.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-108 / lines: 21 Note Achilles

⊖   A security vulnerability has been detected.

```
20
21        mapping(address => uint256) balances;
22        mapping(address => mapping(address => uint256)) allowed;
```

In detail

Labeling the visibility explicitly makes it easier to catch incorrect assumptions about who can access the variable.

4. SWC-108 / lines: 22 Note Achilles

A security vulnerability has been detected.

```
21          mapping(address => uint256) balances;
22          mapping(address => mapping(address => uint256)) allowed;
23
```

In detail

Labeling the visibility explicitly makes it easier to catch incorrect assumptions about who can access the variable.

```
21          mapping(address => uint256) balances;
22          mapping(address => mapping(address => uint256)) allowed;
```

In detail

Labeling the visibility explicitly makes it easier to catch incorrect assumptions about who can access the variable.