

REPORT 615D0D904BB54B00185221BA




Created	Wed Oct 06 2021 02:44:32 GMT+0000 (Coordinated Universal Time)
Number of analyses	1
User	615cffe43f2c3497112e999

REPORT SUMMARY

Analyses ID	Main source file	Detected vulnerabilities
3f0996e2-35e4-432f-bcf0-0e8eb7a1e850	Timelock.sol	1

Started	Wed Oct 06 2021 02:44:33 GMT+0000 (Coordinated Universal Time)
Finished	Wed Oct 06 2021 03:09:15 GMT+0000 (Coordinated Universal Time)
Mode	Deep
Client Tool	Remythx
Main Source File	Timelock.sol

DETECTED VULNERABILITIES

 HIGH	 MEDIUM	 LOW
0	0	1

ISSUES

UNKNOWN Arithmetic operation "+" discovered
This plugin produces issues to support false positive discovery within MythX.
SWC-101

Source file
Timelock.sol
Locations

```
21 | */
22 | function tryAdd(uint256 a, uint256 b) internal pure returns (bool, uint256) {
23 |     uint256 c = a + b;
24 |     if (c < a) return (false, 0);
25 |     return (true, c);
```

UNKNOWN Arithmetic operation "-" discovered
This plugin produces issues to support false positive discovery within MythX.
SWC-101

Source file
Timelock.sol
Locations

```
33 | function trySub(uint256 a, uint256 b) internal pure returns (bool, uint256) {
34 |     if (b > a) return (false, 0);
35 |     return (true, a - b);
36 | }
```

UNKNOWN Arithmetic operation "*" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

Timelock.sol

Locations

```
46 | // See: https://github.com/OpenZeppelin/openzeppelin-contracts/pull/522
47 | if (a == 0) return (true, 0);
48 | uint256 c = a * b;
49 | if (c / a != b) return (false, 0);
50 | return (true, c);
```

UNKNOWN Arithmetic operation "/" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

Timelock.sol

Locations

```
47 | if (a == 0) return (true, 0);
48 | uint256 c = a * b;
49 | if (c/a != b) return (false, 0);
50 | return (true, c);
51 | }
```

UNKNOWN Arithmetic operation "/" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

Timelock.sol

Locations

```
58 | function tryDiv(uint256 a, uint256 b) internal pure returns (bool, uint256) {
59 | if (b == 0) return (false, 0);
60 | return (true, a / b);
61 | }
```

UNKNOWN Arithmetic operation "%" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

Timelock.sol

Locations

```
68 | function tryMod(uint256 a, uint256 b) internal pure returns (bool, uint256) {
69 | if (b == 0) return (false, 0);
70 | return (true, a % b);
71 | }
```

UNKNOWN Arithmetic operation "+" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

Timelock.sol

Locations

```
82 | */
83 | function add(uint256 a, uint256 b) internal pure returns (uint256) {
84 |     uint256 c = a + b;
85 |     require(c >= a, "SafeMath: addition overflow");
86 |     return c;
```

UNKNOWN Arithmetic operation "-" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

Timelock.sol

Locations

```
99 | function sub(uint256 a, uint256 b) internal pure returns (uint256) {
100 |     require(b <= a, "SafeMath: subtraction overflow");
101 |     return a - b;
102 | }
```

UNKNOWN Arithmetic operation "*" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

Timelock.sol

Locations

```
114 | function mul(uint256 a, uint256 b) internal pure returns (uint256) {
115 |     if (a == 0) return 0;
116 |     uint256 c = a * b;
117 |     require(c / a == b, "SafeMath: multiplication overflow");
118 |     return c;
```

UNKNOWN Arithmetic operation "/" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

Timelock.sol

Locations

```
115 | if (a == 0) return 0;
116 | uint256 c = a * b;
117 | require(c / a == b, "SafeMath: multiplication overflow");
118 | return c;
119 | }
```

UNKNOWN Arithmetic operation "/" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

Timelock.sol

Locations

```
133 | function div(uint256 a, uint256 b) internal pure returns (uint256) {
134 |     require(b > 0, "SafeMath: division by zero");
135 |     return a / b;
136 | }
```

UNKNOWN Arithmetic operation "%" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

Timelock.sol

Locations

```
150 | function mod(uint256 a, uint256 b) internal pure returns (uint256) {
151 |     require(b > 0, "SafeMath: modulo by zero");
152 |     return a % b;
153 | }
```

UNKNOWN Arithmetic operation "-" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

Timelock.sol

Locations

```
168 | function sub(uint256 a, uint256 b, string memory errorMessage) internal pure returns (uint256) {
169 |     require(b <= a, errorMessage);
170 |     return a - b;
171 | }
```

UNKNOWN Arithmetic operation "/" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

Timelock.sol

Locations

```
188 | function div(uint256 a, uint256 b, string memory errorMessage) internal pure returns (uint256) {
189 |     require(b > 0, errorMessage);
190 |     return a / b;
191 | }
```

UNKNOWN Arithmetic operation "%" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

Timelock.sol

Locations

```
208 | function mod(uint256 a, uint256 b, string memory errorMessage) internal pure returns (uint256) {
209 |     require(b > 0, errorMessage);
210 |     return a % b;
211 | }
212 | }
```

LOW

A floating pragma is set.

The current pragma Solidity directive is ">=0.6.0<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.

SWC-103

Source file

Timelock.sol

Locations

```
1 | pragma solidity >=0.6.0 <0.8.0;
2 |
3 | /**
```