

REPORT 62439506D7F7FE001976FA49




Created Tue Mar 29 2022 23:23:50 GMT+0000 (Coordinated Universal Time)  
Number of analyses 6  
User 62430564ede4f2af258222bd

## REPORT SUMMARY

Analyses ID	Main source file	Detected vulnerabilities
<a href="#">07d81e4a-a389-4823-a6a0-818b7e575025</a>	contracts/utils/Deviation.sol	1
<a href="#">a87e1608-22f3-4940-80e0-1cd49dfdbb03</a>	contracts/utils/GlobalRateLimitedMinter.sol	1
<a href="#">46c980ac-05f7-4646-ac98-e1b54b0002ed</a>	contracts/utils/MultiRateLimited.sol	1
<a href="#">ffbdfcf6-389b-48e0-8e75-75d98470e3c2</a>	contracts/oracle/ScalingPriceOracle.sol	1
<a href="#">cbd296ae-7719-4723-a3f0-aaf9f05f34d5</a>	contracts/oracle/OraclePassThrough.sol	2
<a href="#">86a9de86-1407-4ccd-ad87-85366aa15b38</a>	contracts/peg/NonCustodialPSM.sol	1

Started	Tue Mar 29 2022 23:26:28 GMT+0000 (Coordinated Universal Time)
Finished	Wed Mar 30 2022 00:11:28 GMT+0000 (Coordinated Universal Time)
Mode	Deep
Client Tool	Mythx-Cli-0.6.22
Main Source File	Contracts/Utils/Deviation.Sol

DETECTED VULNERABILITIES

 HIGH	 MEDIUM	 LOW
0	0	1

ISSUES

LOW

A floating pragma is set.

SWC-103

The current pragma Solidity directive is ""^0.8.4"". 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/Utils/Deviation.sol  
Locations

```
1 | // SPDX-License-Identifier: GPL-3.0-or-later
2 | pragma solidity ^0.8.4;
3 |
4 | import {Constants} from "../Constants.sol";
```

Started	Tue Mar 29 2022 23:26:28 GMT+0000 (Coordinated Universal Time)
Finished	Tue Mar 29 2022 23:26:33 GMT+0000 (Coordinated Universal Time)
Mode	Deep
Client Tool	Mythx-CLI-0.6.22
Main Source File	Contracts/Utils/GlobalRateLimitedMinter.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

contracts/Utils/MultiRateLimited.sol

Locations

```
212 | };
213 |
214 | uint256 elapsed = block.timestamp - rateLimitData.lastBufferUsedTime;
215 | return
216 | uint112(
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

contracts/Utils/MultiRateLimited.sol

Locations

```
216 | uint112(
217 |   Math.min(
218 |     rateLimitData.bufferStored +
219 |     rateLimitData.rateLimitPerSecond * elapsed,
220 |     rateLimitData.bufferCap
221 |   )
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

contracts/utils/MultiRateLimited.sol

Locations

```
217 | Math.min(  
218 | rateLimitData.bufferStored +  
219 | (rateLimitData.rateLimitPerSecond * elapsed),  
220 | rateLimitData.bufferCap  
221 | )
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/utils/MultiRateLimited.sol

Locations

```
339 |  
340 | rateLimitPerAddress[rateLimitedAddress].bufferStored = uint112(  
341 | newBuffer - amount  
342 | );
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/utils/MultiRateLimited.sol

Locations

```
349 | rateLimitedAddress,  
350 | amount,  
351 | newBuffer - amount  
352 | );
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/utils/RateLimited.sol

Locations

```
81 | /// @dev replenishes at rateLimitPerSecond per second up to bufferCap  
82 | function buffer() public view returns (uint256) {  
83 | uint256 elapsed = block.timestamp - lastBufferUsedTime;  
84 | return  
85 | Math.min(bufferStored + (rateLimitPerSecond * elapsed), bufferCap);
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

contracts/Utils/RateLimited.sol

Locations

```
83 | uint256 elapsed = block.timestamp - lastBufferUsedTime;
84 | return
85 | Math.min(bufferStored + (rateLimitPerSecond * elapsed), bufferCap);
86 | }
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

contracts/Utils/RateLimited.sol

Locations

```
83 | uint256 elapsed = block.timestamp - lastBufferUsedTime;
84 | return
85 | Math.min(bufferStored + (rateLimitPerSecond * elapsed), bufferCap);
86 | }
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/Utils/RateLimited.sol

Locations

```
104 | require(usedAmount <= newBuffer, "RateLimited: rate limit hit");
105 |
106 | bufferStored = newBuffer - usedAmount;
107 |
108 | lastBufferUsedTime = block.timestamp;
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

contracts/Utils/RateLimited.sol

Locations

```
126 |
127 | /// ensure that bufferStored cannot be gt buffer cap
128 | bufferStored = Math.min(newBuffer + amount, _bufferCap);
129 | }
```

## UNKNOWN Arithmetic operation "++" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/Strings.sol

Locations

```
23 | uint256 digits;  
24 | while (temp != 0) {  
25 |     digits++;  
26 |     temp /= 10;  
27 | }
```

## UNKNOWN Arithmetic operation "/"=" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/Strings.sol

Locations

```
24 | while (temp != 0) {  
25 |     digits++;  
26 |     temp /= 10;  
27 | }  
28 | bytes memory buffer = new bytes(digits);
```

## UNKNOWN Arithmetic operation "-=" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/Strings.sol

Locations

```
28 | bytes memory buffer = new bytes(digits);  
29 | while (value != 0) {  
30 |     digits -= 1;  
31 |     buffer[digits] = bytes1(uint8(48 + uint256(value % 10)));  
32 |     value /= 10;
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/Strings.sol

Locations

```
29 | while (value != 0) {  
30 |     digits -= 1;  
31 |     buffer[digits] = bytes1(uint8(48 + uint256(value % 10)));  
32 |     value /= 10;  
33 | }
```

## UNKNOWN Arithmetic operation "%" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/Strings.sol

Locations

```
29 | while (value != 0) {  
30 |     digits -= 1;  
31 |     buffer[digits] = bytes1(uint8(48 + uint256(value % 10)));  
32 |     value /= 10;  
33 | }
```

## UNKNOWN Arithmetic operation "/"=" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/Strings.sol

Locations

```
30 | digits -= 1;  
31 | buffer[digits] = bytes1(uint8(48 + uint256(value % 10)));  
32 | value /= 10;  
33 | }  
34 | return string(buffer);
```

## UNKNOWN Arithmetic operation "++" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/Strings.sol

Locations

```
45 | uint256 length = 0;  
46 | while (temp != 0) {  
47 |     length++;  
48 |     temp >>= 8;  
49 | }
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/Strings.sol

Locations

```
55 | */  
56 | function toHexString(uint256 value, uint256 length) internal pure returns (string memory) {  
57 |     bytes memory buffer = new bytes(2 * length + 2);  
58 |     buffer[0] = "0";  
59 |     buffer[1] = "x";
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/Strings.sol

Locations

```
55 | */  
56 | function toHexString(uint256 value, uint256 length) internal pure returns (string memory) {  
57 |     bytes memory buffer = new bytes(2 * length + 2);  
58 |     buffer[0] = "0";  
59 |     buffer[1] = "x";
```



## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/Strings.sol

Locations

```
58 | buffer[0] = "0";
59 | buffer[1] = "x";
60 | for (uint256 i = 2 * length + 1; i > 1; --i) {
61 |     buffer[i] = _HEX_SYMBOLS[value & 0xf];
62 |     value >>= 4;
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/Strings.sol

Locations

```
58 | buffer[0] = "0";
59 | buffer[1] = "x";
60 | for (uint256 i = 2 * length + 1; i > 1; --i) {
61 |     buffer[i] = _HEX_SYMBOLS[value & 0xf];
62 |     value >>= 4;
```

## UNKNOWN Arithmetic operation "--" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/Strings.sol

Locations

```
58 | buffer[0] = "0";
59 | buffer[1] = "x";
60 | for (uint256 i = 2 * length + 1; i > 1; --i) {
61 |     buffer[i] = _HEX_SYMBOLS[value & 0xf];
62 |     value >>= 4;
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/math/Math.sol

Locations

```
28 | function average(uint256 a, uint256 b) internal pure returns (uint256) {
29 | // (a + b) / 2 can overflow.
30 | return a & b + (a ^ b) / 2;
31 | }
```

## UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/math/Math.sol

Locations

```
28 | function average(uint256 a, uint256 b) internal pure returns (uint256) {
29 | // (a + b) / 2 can overflow.
30 | return (a & b) + (a ^ b) / 2;
31 | }
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/math/Math.sol

Locations

```
39 | function ceilDiv(uint256 a, uint256 b) internal pure returns (uint256) {
40 | // (a + b - 1) / b can overflow on addition, so we distribute.
41 | return a / b + (a % b == 0 ? 0 : 1);
42 | }
43 | }
```

## UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/math/Math.sol

Locations

```
39 | function ceilDiv(uint256 a, uint256 b) internal pure returns (uint256) {
40 | // (a + b - 1) / b can overflow on addition, so we distribute.
41 | return a / b + (a % b == 0 ? 0 : 1);
42 | }
43 | }
```

## UNKNOWN Arithmetic operation "%" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/math/Math.sol

Locations

```
39 | function ceilDiv(uint256 a, uint256 b) internal pure returns (uint256) {
40 |     // (a + b - 1) / b can overflow on addition, so we distribute.
41 |     return a / b + (a % b == 0 ? 0 : 1);
42 | }
43 | }
```

LOW

A floating pragma is set.

The current pragma Solidity directive is ""^0.8.4"". 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

contracts/utils/GlobalRateLimitedMinter.sol

Locations

```
1 | // SPDX-License-Identifier: GPL-3.0-or-later
2 | pragma solidity ^0.8.4;
3 |
4 | import {MultiRateLimited} from "./MultiRateLimited.sol";
```

## UNKNOWN Out of bounds array access

The index access expression can cause an exception in case of use of invalid array index value.

SWC-110

Source file

node\_modules/@openzeppelin/contracts/utils/Strings.sol

Locations

```
29 | while (value != 0) {
30 |     digits -= 1;
31 |     buffer[digits] = bytes1(uint8(48 + uint256(value % 10)));
32 |     value /= 10;
33 | }
```

## UNKNOWN Out of bounds array access

The index access expression can cause an exception in case of use of invalid array index value.

SWC-110

Source file

node\_modules/@openzeppelin/contracts/utils/Strings.sol

Locations

```
56 | function toHexString(uint256 value, uint256 length) internal pure returns (string memory) {  
57 |     bytes memory buffer = new bytes(2 * length + 2);  
58 |     buffer[0] = "0";  
59 |     buffer[1] = "x";  
60 |     for (uint256 i = 2 * length + 1; i > 1; --i) {
```

## UNKNOWN Out of bounds array access

The index access expression can cause an exception in case of use of invalid array index value.

SWC-110

Source file

node\_modules/@openzeppelin/contracts/utils/Strings.sol

Locations

```
57 |     bytes memory buffer = new bytes(2 * length + 2);  
58 |     buffer[0] = "0";  
59 |     buffer[1] = "x";  
60 |     for (uint256 i = 2 * length + 1; i > 1; --i) {  
61 |         buffer[i] = _HEX_SYMBOLS[value & 0xf];
```

## UNKNOWN Out of bounds array access

The index access expression can cause an exception in case of use of invalid array index value.

SWC-110

Source file

node\_modules/@openzeppelin/contracts/utils/Strings.sol

Locations

```
59 |     buffer[1] = "x";  
60 |     for (uint256 i = 2 * length + 1; i > 1; --i) {  
61 |         buffer[i] = _HEX_SYMBOLS[value & 0xf];  
62 |         value >>= 4;  
63 |     }
```

## UNKNOWN Out of bounds array access

The index access expression can cause an exception in case of use of invalid array index value.

SWC-110

Source file

node\_modules/@openzeppelin/contracts/utils/Strings.sol

Locations

```
59 | buffer[1] = "x";
60 | for (uint256 i = 2 * length + 1; i > 1; --i) {
61 |     buffer[i] = HEX_SYMBOLS[ value >> 4 ];
62 |     value >>= 4;
63 | }
```

Started	Tue Mar 29 2022 23:26:38 GMT+0000 (Coordinated Universal Time)
Finished	Wed Mar 30 2022 00:11:40 GMT+0000 (Coordinated Universal Time)
Mode	Deep
Client Tool	Mythx-CLI-0.6.22
Main Source File	Contracts/Utils/MultiRateLimited.sol

## DETECTED VULNERABILITIES

HIGH	MEDIUM	LOW
0	0	1

## ISSUES

LOW	A floating pragma is set.
SWC-103	The current pragma Solidity directive is ""^0.8.4"". 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/utils/MultiRateLimited.sol

Locations

```
1 | // SPDX-License-Identifier: GPL-3.0-or-later
2 | pragma solidity ^0.8.4;
3 |
4 | import {CoreRef} from "../refs/CoreRef.sol";
```

Started	Tue Mar 29 2022 23:26:38 GMT+0000 (Coordinated Universal Time)
Finished	Tue Mar 29 2022 23:31:04 GMT+0000 (Coordinated Universal Time)
Mode	Deep
Client Tool	Mythx-Cli-0.6.22
Main Source File	Contracts/Oracle/ScalingPriceOracle.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

contracts/external/Decimal.sol

Locations

```
28 | // ===== Constants =====
29 |
30 | uint256 private constant BASE = 10**18;
31 |
32 | // ===== Structs =====
```

## UNKNOWN Arithmetic operation "++" discovered

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

SWC-101

Source file

contracts/external/Decimal.sol

Locations

```
106 |
107 | D256 memory temp = D256({value: self.value});
108 | for (uint256 i = 1; i < b; i++) {
109 |     temp = mul(temp, self);
110 | }
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeContract.sol

Locations

```
33 |
34 | contract BokkyPooBahsDateTimeContract {
35 |     uint256 public constant SECONDS_PER_DAY = 24 * 60 * 60;
36 |     uint256 public constant SECONDS_PER_HOUR = 60 * 60;
37 |     uint256 public constant SECONDS_PER_MINUTE = 60;
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeContract.sol

Locations

```
33 |
34 | contract BokkyPooBahsDateTimeContract {
35 |     uint256 public constant SECONDS_PER_DAY = 24 * 60 * 60;
36 |     uint256 public constant SECONDS_PER_HOUR = 60 * 60;
37 |     uint256 public constant SECONDS_PER_MINUTE = 60;
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeContract.sol

Locations

```
34 | contract BokkyPooBahsDateTimeContract {
35 |     uint256 public constant SECONDS_PER_DAY = 24 * 60 * 60;
36 |     uint256 public constant SECONDS_PER_HOUR = 60 * 60;
37 |     uint256 public constant SECONDS_PER_MINUTE = 60;
38 |     int256 public constant OFFSET19700101 = 2440588;
```



## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
31 |
32 | library BokkyPooBahsDateTimeLibrary {
33 |     uint256 constant SECONDS_PER_DAY = 24 * 60 * 60;
34 |     uint256 constant SECONDS_PER_HOUR = 60 * 60;
35 |     uint256 constant SECONDS_PER_MINUTE = 60;
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
31 |
32 | library BokkyPooBahsDateTimeLibrary {
33 |     uint256 constant SECONDS_PER_DAY = 24 * 60 * 60;
34 |     uint256 constant SECONDS_PER_HOUR = 60 * 60;
35 |     uint256 constant SECONDS_PER_MINUTE = 60;
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
32 | library BokkyPooBahsDateTimeLibrary {
33 |     uint256 constant SECONDS_PER_DAY = 24 * 60 * 60;
34 |     uint256 constant SECONDS_PER_HOUR = 60 * 60;
35 |     uint256 constant SECONDS_PER_MINUTE = 60;
36 |     int256 constant OFFSET19700101 = 2440588;
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
67 | int256 _day = int256(day);
68 |
69 | int256 __days = _day -
70 | 32075 +
71 | (1461 * (_year + 4800 + ((_month - 14) / 12))) /
72 | 4 +
73 | (367 * (_month - 2 - ((_month - 14) / 12) * 12)) /
74 | 12 -
75 | (3 * ((_year + 4900 + ((_month - 14) / 12) / 100)) /
76 | 4 -
77 | OFFSET19700101;
78 |
79 | _days = uint256(__days);
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
67 | int256 _day = int256(day);
68 |
69 | int256 __days = _day -
70 | 32075 +
71 | (1461 * (_year + 4800 + ((_month - 14) / 12))) /
72 | 4 +
73 | (367 * (_month - 2 - ((_month - 14) / 12) * 12)) /
74 | 12 -
75 | (3 * ((_year + 4900 + ((_month - 14) / 12) / 100)) /
76 | 4 -
77 | OFFSET19700101;
```

UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
67 | int256 _day = int256(day);
68 |
69 | int256 __days = _day -
70 | 32075 +
71 | 1461 * (_year + 4800 + ((_month - 14) / 12)) /
72 | 4 +
73 | 367 * (_month - 2 - (((_month - 14) / 12) * 12)) /
74 | 12 -
75 | (3 * ((_year + 4900 + (_month - 14) / 12) / 100)) /
76 | 4 -
```

UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
67 | int256 _day = int256(day);
68 |
69 | int256 __days = _day -
70 | 32075 +
71 | 1461 * (_year + 4800 + ((_month - 14) / 12)) /
72 | 4 +
73 | (367 * (_month - 2 - (((_month - 14) / 12) * 12)) /
74 | 12 -
```

UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
67 | int256 _day = int256(day);
68 |
69 | int256 __days = _day -
70 | 32075 +
71 | (1461 * (_year + 4800 + (_month - 14) / 12)) /
72 | 4 +
```

UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
69 | int256 __days = _day -
70 | 32075 +
71 | (1461 * (_year + 4800 + ((_month - 14) / 12)) /
72 | 4 +
73 | (367 * (_month - 2 - (((_month - 14) / 12) * 12)) /
74 | 12 -
```

UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
69 | int256 __days = _day -
70 | 32075 +
71 | (1461 * (_year + 4800 + ((_month - 14) / 12)) /
72 | 4 +
73 | (367 * (_month - 2 - (((_month - 14) / 12) * 12)) /
```

UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
69 | int256 __days = _day -
70 | 32075 +
71 | (1461 * (_year + 4800 + ((_month - 14) / 12)) /
72 | 4 +
73 | (367 * (_month - 2 - (((_month - 14) / 12) * 12)) /
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
69 | int256 __days = _day -
70 | 32075 +
71 | (1461 * (_year + 4800 + (_month - 14) / 12)) /
72 | 4 +
73 | (367 * (_month - 2 - ((month - 14) / 12) * 12)) /
```

## UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
69 | int256 __days = _day -
70 | 32075 +
71 | (1461 * (_year + 4800 + (_month - 14) / 12)) /
72 | 4 +
73 | (367 * (_month - 2 - ((month - 14) / 12) * 12)) /
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
69 | int256 __days = _day -
70 | 32075 +
71 | (1461 * (_year + 4800 + (_month - 14) / 12)) /
72 | 4 +
73 | (367 * (_month - 2 - ((month - 14) / 12) * 12)) /
```

## UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
71 | (1461 * (_year + 4800 + (_month - 14) / 12)) /  
72 | 4 +  
73 | (367 * (_month - 2 - ((_month - 14) / 12) * 12)) /  
74 | 12 -  
75 | (3 * ((_year + 4900 + (_month - 14) / 12) / 100)) /  
76 | 4 -
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
71 | (1461 * (_year + 4800 + (_month - 14) / 12)) /  
72 | 4 +  
73 | (367 * (_month - 2 - ((_month - 14) / 12) * 12)) /  
74 | 12 -  
75 | (3 * ((_year + 4900 + (_month - 14) / 12) / 100)) /
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
71 | (1461 * (_year + 4800 + (_month - 14) / 12)) /  
72 | 4 +  
73 | (367 * (_month - 2 - ((_month - 14) / 12) * 12)) /  
74 | 12 -  
75 | (3 * ((_year + 4900 + (_month - 14) / 12) / 100)) /
```

UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
71 | (1461 * (_year + 4800 + (_month - 14) / 12)) /
72 | 4 +
73 | (367 * (_month - 2 - (((_month - 14) / 12) * 12)) /
74 | 12 -
75 | (3 * (( _year + 4900 + (_month - 14) / 12) / 100)) /
```

UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
71 | (1461 * (_year + 4800 + (_month - 14) / 12)) /
72 | 4 +
73 | (367 * (_month - 2 - (( _month - 14) / 12) * 12)) /
74 | 12 -
75 | (3 * (( _year + 4900 + (_month - 14) / 12) / 100)) /
```

UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
71 | (1461 * (_year + 4800 + (_month - 14) / 12)) /
72 | 4 +
73 | (367 * (_month - 2 - (( _month - 14) / 12) * 12)) /
74 | 12 -
75 | (3 * (( _year + 4900 + (_month - 14) / 12) / 100)) /
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
71 | (1461 * (_year + 4800 + (_month - 14) / 12)) /  
72 | 4 +  
73 | (367 * (_month - 2 - ((month - 14) / 12) * 12)) /  
74 | 12 -  
75 | (3 * ((year + 4900 + (_month - 14) / 12) / 100)) /
```

## UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
73 | (367 * (_month - 2 - ((month - 14) / 12) * 12)) /  
74 | 12 -  
75 | 3 * ((year + 4900 + (_month - 14) / 12) / 100) /  
76 | 4 -  
77 | OFFSET19700101;
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
73 | (367 * (_month - 2 - ((month - 14) / 12) * 12)) /  
74 | 12 -  
75 | (3 * ((year + 4900 + (_month - 14) / 12) / 100)) /  
76 | 4 -  
77 | OFFSET19700101;
```



## UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
73 | (367 * (_month - 2 - ((_month - 14) / 12) * 12)) /  
74 | 12 -  
75 | (3 * ((_year + 4900 + (_month - 14) / 12) / 100)) /  
76 | 4 -  
77 | OFFSET19700101;
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
73 | (367 * (_month - 2 - ((_month - 14) / 12) * 12)) /  
74 | 12 -  
75 | (3 * (( _year + 4900 + (_month - 14) / 12) / 100)) /  
76 | 4 -  
77 | OFFSET19700101;
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
73 | (367 * (_month - 2 - ((_month - 14) / 12) * 12)) /  
74 | 12 -  
75 | (3 * (( _year + 4900 + (_month - 14) / 12) / 100)) /  
76 | 4 -  
77 | OFFSET19700101;
```

## UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
73 | (367 * (_month - 2 - (( _month - 14 ) / 12 ) * 12)) /  
74 | 12 -  
75 | (3 * (( _year + 4900 + ( _month - 14 ) / 12 ) / 100)) /  
76 | 4 -  
77 | OFFSET19700101;
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
73 | (367 * (_month - 2 - (( _month - 14 ) / 12 ) * 12)) /  
74 | 12 -  
75 | (3 * (( _year + 4900 + ( _month - 14 ) / 12 ) / 100)) /  
76 | 4 -  
77 | OFFSET19700101;
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
108 | int256 __days = int256(_days);  
109 |  
110 | int256 L = __days + 68569 + OFFSET19700101;  
111 | int256 N = (4 * L) / 146097;  
112 | L = L - (146097 * N + 3) / 4;
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
108 | int256 __days = int256(_days);
109 |
110 | int256 L = __days + 68569 + OFFSET19700101;
111 | int256 N = (4 * L) / 146097;
112 | L = L - (146097 * N + 3) / 4;
```

## UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
109 |
110 | int256 L = __days + 68569 + OFFSET19700101;
111 | int256 N = (4 * L) / 146097;
112 | L = L - (146097 * N + 3) / 4;
113 | int256 _year = (4000 * (L + 1)) / 1461001;
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
109 |
110 | int256 L = __days + 68569 + OFFSET19700101;
111 | int256 N = (4 * L) / 146097;
112 | L = L - (146097 * N + 3) / 4;
113 | int256 _year = (4000 * (L + 1)) / 1461001;
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
110 | int256 L = __days + 68569 + OFFSET19700101;  
111 | int256 N = (4 * L) / 146097;  
112 | L = L - (146097 * N + 3) / 4;  
113 | int256 _year = (4000 * (L + 1)) / 1461001;  
114 | L = L - (1461 * _year) / 4 + 31;
```

## UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
110 | int256 L = __days + 68569 + OFFSET19700101;  
111 | int256 N = (4 * L) / 146097;  
112 | L = L - (146097 * N + 3) / 4;  
113 | int256 _year = (4000 * (L + 1)) / 1461001;  
114 | L = L - (1461 * _year) / 4 + 31;
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
110 | int256 L = __days + 68569 + OFFSET19700101;  
111 | int256 N = (4 * L) / 146097;  
112 | L = L - (146097 * N + 3) / 4;  
113 | int256 _year = (4000 * (L + 1)) / 1461001;  
114 | L = L - (1461 * _year) / 4 + 31;
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
110 | int256 L = __days + 68569 + OFFSET19700101;  
111 | int256 N = (4 * L) / 146097;  
112 | L = L - (146097 * N + 3) / 4;  
113 | int256 _year = (4000 * (L + 1)) / 1461001;  
114 | L = L - (1461 * _year) / 4 + 31;
```

## UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
111 | int256 N = (4 * L) / 146097;  
112 | L = L - (146097 * N + 3) / 4;  
113 | int256 _year = (4000 * (L + 1)) / 1461001;  
114 | L = L - (1461 * _year) / 4 + 31;  
115 | int256 _month = (80 * L) / 2447;
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
111 | int256 N = (4 * L) / 146097;  
112 | L = L - (146097 * N + 3) / 4;  
113 | int256 _year = (4000 * (L + 1)) / 1461001;  
114 | L = L - (1461 * _year) / 4 + 31;  
115 | int256 _month = (80 * L) / 2447;
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
111 | int256 N = (4 * L) / 146097;  
112 | L = L - (146097 * N + 3) / 4;  
113 | int256 _year = (4000 * (L + 1)) / 1461001;  
114 | L = L - (1461 * _year) / 4 + 31;  
115 | int256 _month = (80 * L) / 2447;
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
112 | L = L - (146097 * N + 3) / 4;  
113 | int256 _year = (4000 * (L + 1)) / 1461001;  
114 | L = L - (1461 * _year) / 4 + 31;  
115 | int256 _month = (80 * L) / 2447;  
116 | int256 _day = L - (2447 * _month) / 80;
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
112 | L = L - (146097 * N + 3) / 4;  
113 | int256 _year = (4000 * (L + 1)) / 1461001;  
114 | L = L - (1461 * _year) / 4 + 31;  
115 | int256 _month = (80 * L) / 2447;  
116 | int256 _day = L - (2447 * _month) / 80;
```

UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
112 | L = L - (146097 * N + 3) / 4;  
113 | int256 _year = (4000 * (L + 1)) / 1461001;  
114 | L = L - (1461 * _year) / 4 + 31;  
115 | int256 _month = (80 * L) / 2447;  
116 | int256 _day = L - (2447 * _month) / 80;
```

UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
112 | L = L - (146097 * N + 3) / 4;  
113 | int256 _year = (4000 * (L + 1)) / 1461001;  
114 | L = L - (1461 * _year) / 4 + 31;  
115 | int256 _month = (80 * L) / 2447;  
116 | int256 _day = L - (2447 * _month) / 80;
```

UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
113 | int256 _year = (4000 * (L + 1)) / 1461001;  
114 | L = L - (1461 * _year) / 4 + 31;  
115 | int256 _month = (80 * L) / 2447;  
116 | int256 _day = L - (2447 * _month) / 80;  
117 | L = _month / 11;
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
113 | int256 _year = (4000 * (L + 1)) / 1461001;
114 | L = L - (1461 * _year) / 4 + 31;
115 | int256 _month = (80 * L) / 2447;
116 | int256 _day = L - (2447 * _month) / 80;
117 | L = _month / 11;
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
114 | L = L - (1461 * _year) / 4 + 31;
115 | int256 _month = (80 * L) / 2447;
116 | int256 _day = L - (2447 * _month) / 80;
117 | L = _month / 11;
118 | _month = _month + 2 - 12 * L;
```

## UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
114 | L = L - (1461 * _year) / 4 + 31;
115 | int256 _month = (80 * L) / 2447;
116 | int256 _day = L - (2447 * _month) / 80;
117 | L = _month / 11;
118 | _month = _month + 2 - 12 * L;
```



## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
114 | L = L - (1461 * _year) / 4 + 31;  
115 | int256 _month = (80 * L) / 2447;  
116 | int256 _day = L - (2447 * _month) / 80;  
117 | L = _month / 11;  
118 | _month = _month + 2 - 12 * L;
```

## UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
115 | int256 _month = (80 * L) / 2447;  
116 | int256 _day = L - (2447 * _month) / 80;  
117 | L = _month / 11;  
118 | _month = _month + 2 - 12 * L;  
119 | _year = 100 * (N - 49) + _year + L;
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
116 | int256 _day = L - (2447 * _month) / 80;  
117 | L = _month / 11;  
118 | _month = _month + 2 - 12 * L;  
119 | _year = 100 * (N - 49) + _year + L;
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
116 | int256 _day = L - (2447 * _month) / 80;  
117 | L = _month / 11;  
118 | _month = _month + 2 - 12 * L;  
119 | _year = 100 * (N - 49) + _year + L;
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
116 | int256 _day = L - (2447 * _month) / 80;  
117 | L = _month / 11;  
118 | _month = _month + 2 - 12 * L;  
119 | _year = 100 * (N - 49) + _year + L;
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
117 | L = _month / 11;  
118 | _month = _month + 2 - 12 * L;  
119 | _year = 100 * (N - 49) + _year + L;  
120 |  
121 | year = uint256(_year);
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
117 | L = _month / 11;  
118 | _month = _month + 2 - 12 * L;  
119 | _year = 100 * (N - 49) + _year + L;  
120 |  
121 | year = uint256(_year);
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
117 | L = _month / 11;  
118 | _month = _month + 2 - 12 * L;  
119 | _year = 100 * (N - 49) + _year + L;  
120 |  
121 | year = uint256(_year);
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
117 | L = _month / 11;  
118 | _month = _month + 2 - 12 * L;  
119 | _year = 100 * (N - 49) + _year + L;  
120 |  
121 | year = uint256(_year);
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
129 | uint256 day  
130 | ) internal pure returns (uint256 timestamp) {  
131 | timestamp = _daysFromDate(year, month, day) * SECONDS_PER_DAY;  
132 | }
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
141 | ) internal pure returns (uint256 timestamp) {  
142 |     timestamp =  
143 |     daysFromDate(year, month, day) *  
144 |     SECONDS_PER_DAY +  
145 |     hour *  
146 |     SECONDS_PER_HOUR +  
147 |     minute *  
148 |     SECONDS_PER_MINUTE +  
149 |     second;  
150 | }
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
141 | ) internal pure returns (uint256 timestamp) {  
142 |     timestamp =  
143 |     daysFromDate(year, month, day) *  
144 |     SECONDS_PER_DAY +  
145 |     hour *  
146 |     SECONDS_PER_HOUR +  
147 |     minute *  
148 |     SECONDS_PER_MINUTE +  
149 |     second;  
150 | }
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
141 | ) internal pure returns (uint256 timestamp) {  
142 |     timestamp =  
143 |     daysFromDate(year, month, day) *  
144 |     SECONDS_PER_DAY +  
145 |     hour *  
146 |     SECONDS_PER_HOUR +  
147 |     minute *  
148 |     SECONDS_PER_MINUTE +
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
141 | ) internal pure returns (uint256 timestamp) {
142 |     timestamp =
143 |     daysFromDate(year, month, day) *
144 |     SECONDS_PER_DAY +
145 |     hour *
146 |     SECONDS_PER_HOUR +
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
143 | _daysFromDate(year, month, day) *
144 | SECONDS_PER_DAY +
145 | hour *
146 | SECONDS_PER_HOUR +
147 | minute *
148 | SECONDS_PER_MINUTE +
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
145 | hour *
146 | SECONDS_PER_HOUR +
147 | minute *
148 | SECONDS_PER_MINUTE +
149 | second;
150 | }
```

## UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
159 | )
160 | {
161 | (year, month, day) = _daysToDate(timestamp / SECONDS_PER_DAY);
162 | }
```

## UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
174 | )
175 | {
176 | (year, month, day) = _daysToDate(timestamp / SECONDS_PER_DAY);
177 | uint256 secs = timestamp % SECONDS_PER_DAY;
178 | hour = secs / SECONDS_PER_HOUR;
```

## UNKNOWN Arithmetic operation "%" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
175 | {
176 | (year, month, day) = _daysToDate(timestamp / SECONDS_PER_DAY);
177 | uint256 secs = timestamp % SECONDS_PER_DAY;
178 | hour = secs / SECONDS_PER_HOUR;
179 | secs = secs % SECONDS_PER_HOUR;
```

## UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
176 | (year, month, day) = _daysToDate(timestamp / SECONDS_PER_DAY);
177 | uint256 secs = timestamp % SECONDS_PER_DAY;
178 | hour = secs / SECONDS_PER_HOUR;
179 | secs = secs % SECONDS_PER_HOUR;
180 | minute = secs / SECONDS_PER_MINUTE;
```

## UNKNOWN Arithmetic operation "%" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
177 | uint256 secs = timestamp % SECONDS_PER_DAY;
178 | hour = secs / SECONDS_PER_HOUR;
179 | secs = secs % SECONDS_PER_HOUR;
180 | minute = secs / SECONDS_PER_MINUTE;
181 | second = secs % SECONDS_PER_MINUTE;
```

## UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
178 | hour = secs / SECONDS_PER_HOUR;
179 | secs = secs % SECONDS_PER_HOUR;
180 | minute = secs / SECONDS_PER_MINUTE;
181 | second = secs % SECONDS_PER_MINUTE;
182 | }
```

## UNKNOWN Arithmetic operation "%" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
179 | secs = secs % SECONDS_PER_HOUR;  
180 | minute = secs / SECONDS_PER_MINUTE;  
181 | second = secs % SECONDS_PER_MINUTE;  
182 | }
```

## UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
218 | uint256 month;  
219 | uint256 day;  
220 | (year, month, day) = _daysToDate(timestamp / SECONDS_PER_DAY);  
221 | leapYear = _isLeapYear(year);  
222 | }
```

## UNKNOWN Arithmetic operation "%" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
223 |  
224 | function _isLeapYear(uint256 year) internal pure returns (bool leapYear) {  
225 | leapYear = ((year % 4 == 0) && (year % 100 != 0)) || (year % 400 == 0);  
226 | }
```

## UNKNOWN Arithmetic operation "%" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
223 |  
224 | function _isLeapYear(uint256 year) internal pure returns (bool leapYear) {  
225 | leapYear = ((year % 4 == 0) && (year % 100 != 0)) || (year % 400 == 0);  
226 | }
```



UNKNOWN Arithmetic operation "%" discovered

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

SWC-101

Source file  
contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol  
Locations

```
223 |  
224 | function _isLeapYear(uint256 year) internal pure returns (bool leapYear) {  
225 |     leapYear = ((year % 4 == 0) && (year % 100 != 0)) || (year % 400 == 0);  
226 | }
```

UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file  
contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol  
Locations

```
242 | uint256 month;  
243 | uint256 day;  
244 | (year, month, day) = _daysToDate(timestamp / SECONDS_PER_DAY);  
245 | daysInMonth = _getDaysInMonth(year, month);  
246 | }
```

UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file  
contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol  
Locations

```
274 | returns (uint256 dayOfWeek)  
275 | {  
276 |     uint256 _days = timestamp / SECONDS_PER_DAY;  
277 |     dayOfWeek = ((_days + 3) % 7) + 1;  
278 | }
```

UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file  
contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol  
Locations

```
275 | {  
276 |     uint256 _days = timestamp / SECONDS_PER_DAY;  
277 |     dayOfWeek = ((_days + 3) % 7) + 1;  
278 | }
```

## UNKNOWN Arithmetic operation "%" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
275 | {  
276 | uint256 _days = timestamp / SECONDS_PER_DAY;  
277 | dayOfWeek = ((_days + 3) % 7) + 1;  
278 | }
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
275 | {  
276 | uint256 _days = timestamp / SECONDS_PER_DAY;  
277 | dayOfWeek = (((_days + 3) % 7) + 1);  
278 | }
```

## UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
281 | uint256 month;  
282 | uint256 day;  
283 | (year, month, day) = _daysToDate(timestamp / SECONDS_PER_DAY);  
284 | }
```

## UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
287 | uint256 year;  
288 | uint256 day;  
289 | (year, month, day) = _daysToDate(timestamp / SECONDS_PER_DAY);  
290 | }
```

## UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
293 | uint256 year;  
294 | uint256 month;  
295 | (year, month, day) = _daysToDate(timestamp / SECONDS_PER_DAY);  
296 | }
```

## UNKNOWN Arithmetic operation "%" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
297 |  
298 | function getHour(uint256 timestamp) internal pure returns (uint256 hour) {  
299 |     uint256 secs = timestamp % SECONDS_PER_DAY;  
300 |     hour = secs / SECONDS_PER_HOUR;  
301 | }
```

## UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
298 | function getHour(uint256 timestamp) internal pure returns (uint256 hour) {  
299 |     uint256 secs = timestamp % SECONDS_PER_DAY;  
300 |     hour = secs / SECONDS_PER_HOUR;  
301 | }
```

## UNKNOWN Arithmetic operation "%" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
306 | returns (uint256 minute)  
307 | {  
308 |     uint256 secs = timestamp % SECONDS_PER_HOUR;  
309 |     minute = secs / SECONDS_PER_MINUTE;  
310 | }
```

## UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
307 | {  
308 |     uint256 secs = timestamp % SECONDS_PER_HOUR;  
309 |     minute = secs / SECONDS_PER_MINUTE;  
310 | }
```

## UNKNOWN Arithmetic operation "%" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
315 | returns (uint256 second)  
316 | {  
317 |     second = timestamp % SECONDS_PER_MINUTE;  
318 | }
```

## UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
326 | uint256 month;  
327 | uint256 day;  
328 | (year, month, day) = _daysToDate(timestamp / SECONDS_PER_DAY);  
329 | year += _years;  
330 | uint256 daysInMonth = _getDaysInMonth(year, month);
```

## UNKNOWN Arithmetic operation "+=" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
327 | uint256 day;  
328 | (year, month, day) = _daysToDate(timestamp / SECONDS_PER_DAY);  
329 | year += _years;  
330 | uint256 daysInMonth = _getDaysInMonth(year, month);  
331 | if (day > daysInMonth) {
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
333 | }
334 | newTimestamp =
335 |   daysFromDate(year, month, day) *
336 |   SECONDS_PER_DAY +
337 |   timestamp % SECONDS_PER_DAY;
338 | require(newTimestamp >= timestamp);
339 | }
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
333 | }
334 | newTimestamp =
335 |   daysFromDate(year, month, day) *
336 |   SECONDS_PER_DAY +
337 |   (timestamp % SECONDS_PER_DAY);
338 | require(newTimestamp >= timestamp);
```

## UNKNOWN Arithmetic operation "%" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
335 | _daysFromDate(year, month, day) *
336 | SECONDS_PER_DAY +
337 | (timestamp % SECONDS_PER_DAY);
338 | require(newTimestamp >= timestamp);
339 | }
```

## UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
347 | uint256 month;  
348 | uint256 day;  
349 | (year, month, day) = _daysToDate(timestamp / SECONDS_PER_DAY);  
350 | month += _months;  
351 | year += (month - 1) / 12;
```

## UNKNOWN Arithmetic operation "+=" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
348 | uint256 day;  
349 | (year, month, day) = _daysToDate(timestamp / SECONDS_PER_DAY);  
350 | month += _months;  
351 | year += (month - 1) / 12;  
352 | month = ((month - 1) % 12) + 1;
```

## UNKNOWN Arithmetic operation "+=" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
349 | (year, month, day) = _daysToDate(timestamp / SECONDS_PER_DAY);  
350 | month += _months;  
351 | year += (month - 1) / 12;  
352 | month = ((month - 1) % 12) + 1;  
353 | uint256 daysInMonth = _getDaysInMonth(year, month);
```

## UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
349 | (year, month, day) = _daysToDate(timestamp / SECONDS_PER_DAY);
350 | month += _months;
351 | year += (month - 1) / 12;
352 | month = ((month - 1) % 12) + 1;
353 | uint256 daysInMonth = _getDaysInMonth(year, month);
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
349 | (year, month, day) = _daysToDate(timestamp / SECONDS_PER_DAY);
350 | month += _months;
351 | year += (month - 1) / 12;
352 | month = ((month - 1) % 12) + 1;
353 | uint256 daysInMonth = _getDaysInMonth(year, month);
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
350 | month += _months;
351 | year += (month - 1) / 12;
352 | month = (month - 1) % 12 + 1;
353 | uint256 daysInMonth = _getDaysInMonth(year, month);
354 | if (day > daysInMonth) {
```

## UNKNOWN Arithmetic operation "%" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
350 | month += _months;
351 | year += (month - 1) / 12;
352 | month = ((month - 1) % 12) + 1;
353 | uint256 daysInMonth = _getDaysInMonth(year, month);
354 | if (day > daysInMonth) {
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
350 | month += _months;
351 | year += (month - 1) / 12;
352 | month = ((month - 1) % 12) + 1;
353 | uint256 daysInMonth = _getDaysInMonth(year, month);
354 | if (day > daysInMonth) {
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
356 | }
357 | newTimestamp =
358 | _daysFromDate year month day *
359 | SECONDS_PER_DAY +
360 | timestamp % SECONDS_PER_DAY;
361 | require(newTimestamp >= timestamp);
362 | }
```



## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
356 | }
357 | newTimestamp =
358 |   daysFromDate(year, month, day) *
359 |   SECONDS_PER_DAY +
360 |   (timestamp % SECONDS_PER_DAY);
361 | require(newTimestamp >= timestamp);
```

## UNKNOWN Arithmetic operation "%" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
358 |   _daysFromDate(year, month, day) *
359 |   SECONDS_PER_DAY +
360 |   (timestamp % SECONDS_PER_DAY);
361 | require(newTimestamp >= timestamp);
362 | }
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
367 | returns (uint256 newTimestamp)
368 | {
369 |   newTimestamp = timestamp + _days * SECONDS_PER_DAY;
370 |   require(newTimestamp >= timestamp);
371 | }
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
367 | returns (uint256 newTimestamp)
368 | {
369 |     newTimestamp = timestamp + days * SECONDS_PER_DAY;
370 |     require(newTimestamp >= timestamp);
371 | }
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
376 | returns (uint256 newTimestamp)
377 | {
378 |     newTimestamp = timestamp + _hours * SECONDS_PER_HOUR;
379 |     require(newTimestamp >= timestamp);
380 | }
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
376 | returns (uint256 newTimestamp)
377 | {
378 |     newTimestamp = timestamp + _hours * SECONDS_PER_HOUR;
379 |     require(newTimestamp >= timestamp);
380 | }
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
385 | returns (uint256 newTimestamp)
386 | {
387 |   newTimestamp = timestamp + _minutes * SECONDS_PER_MINUTE;
388 |   require(newTimestamp >= timestamp);
389 | }
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
385 | returns (uint256 newTimestamp)
386 | {
387 |   newTimestamp = timestamp + _minutes * SECONDS_PER_MINUTE;
388 |   require(newTimestamp >= timestamp);
389 | }
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
394 | returns (uint256 newTimestamp)
395 | {
396 |   newTimestamp = timestamp + _seconds;
397 |   require(newTimestamp >= timestamp);
398 | }
```

## UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
406 | uint256 month;  
407 | uint256 day;  
408 | (year, month, day) = _daysToDate(timestamp / SECONDS_PER_DAY);  
409 | year -= _years;  
410 | uint256 daysInMonth = _getDaysInMonth(year, month);
```

## UNKNOWN Arithmetic operation "-=" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
407 | uint256 day;  
408 | (year, month, day) = _daysToDate(timestamp / SECONDS_PER_DAY);  
409 | year -= _years;  
410 | uint256 daysInMonth = _getDaysInMonth(year, month);  
411 | if (day > daysInMonth) {
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
413 | }  
414 | newTimestamp =  
415 | _daysFromDate(year, month, day) *  
416 | SECONDS_PER_DAY +  
417 | timestamp % SECONDS_PER_DAY;  
418 | require(newTimestamp <= timestamp);  
419 | }
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
413 | }  
414 | newTimestamp =  
415 | _daysFromDate(year, month, day) *  
416 | SECONDS_PER_DAY +  
417 | (timestamp % SECONDS_PER_DAY);  
418 | require(newTimestamp <= timestamp);
```

## UNKNOWN Arithmetic operation "%" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
415 | _daysFromDate(year, month, day) *  
416 | SECONDS_PER_DAY +  
417 | (timestamp % SECONDS_PER_DAY);  
418 | require(newTimestamp <= timestamp);  
419 | }
```

## UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
427 | uint256 month;  
428 | uint256 day;  
429 | (year, month, day) = _daysToDate(timestamp / SECONDS_PER_DAY);  
430 | uint256 yearMonth = year * 12 + (month - 1) - _months;  
431 | year = yearMonth / 12;
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
428 | uint256 day;  
429 | (year, month, day) = _daysToDate(timestamp / SECONDS_PER_DAY);  
430 | uint256 yearMonth = year * 12 + (month - 1) - _months;  
431 | year = yearMonth / 12;  
432 | month = (yearMonth % 12) + 1;
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
428 | uint256 day;  
429 | (year, month, day) = _daysToDate(timestamp / SECONDS_PER_DAY);  
430 | uint256 yearMonth = year * 12 + (month - 1) - _months;  
431 | year = yearMonth / 12;  
432 | month = (yearMonth % 12) + 1;
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
428 | uint256 day;  
429 | (year, month, day) = _daysToDate(timestamp / SECONDS_PER_DAY);  
430 | uint256 yearMonth = year * 12 + (month - 1) - _months;  
431 | year = yearMonth / 12;  
432 | month = (yearMonth % 12) + 1;
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
428 | uint256 day;  
429 | (year, month, day) = _daysToDate(timestamp / SECONDS_PER_DAY);  
430 | uint256 yearMonth = year * 12 + (month - 1) - _months;  
431 | year = yearMonth / 12;  
432 | month = (yearMonth % 12) + 1;
```

## UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
429 | (year, month, day) = _daysToDate(timestamp / SECONDS_PER_DAY);  
430 | uint256 yearMonth = year * 12 + (month - 1) - _months;  
431 | year = yearMonth / 12;  
432 | month = (yearMonth % 12) + 1;  
433 | uint256 daysInMonth = _getDaysInMonth(year, month);
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
430 | uint256 yearMonth = year * 12 + (month - 1) - _months;  
431 | year = yearMonth / 12;  
432 | month = yearMonth % 12 + 1;  
433 | uint256 daysInMonth = _getDaysInMonth(year, month);  
434 | if (day > daysInMonth) {
```

## UNKNOWN Arithmetic operation "%" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
430 | uint256 yearMonth = year * 12 + (month - 1) * _months;
431 | year = yearMonth / 12;
432 | month = (yearMonth % 12) + 1;
433 | uint256 daysInMonth = _getDaysInMonth(year, month);
434 | if (day > daysInMonth) {
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
436 | }
437 | newTimestamp =
438 | daysFromDate(year, month, day) *
439 | SECONDS_PER_DAY +
440 | timestamp % SECONDS_PER_DAY;
441 | require(newTimestamp <= timestamp);
442 | }
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
436 | }
437 | newTimestamp =
438 | daysFromDate(year, month, day) *
439 | SECONDS_PER_DAY +
440 | (timestamp % SECONDS_PER_DAY);
441 | require(newTimestamp <= timestamp);
```



## UNKNOWN Arithmetic operation "%" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
438 | _daysFromDate(year, month, day) *  
439 | SECONDS_PER_DAY +  
440 | (timestamp % SECONDS_PER_DAY);  
441 | require(newTimestamp <= timestamp);  
442 | }
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
447 | returns (uint256 newTimestamp)  
448 | {  
449 | newTimestamp = timestamp - _days * SECONDS_PER_DAY;  
450 | require(newTimestamp <= timestamp);  
451 | }
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
447 | returns (uint256 newTimestamp)  
448 | {  
449 | newTimestamp = timestamp - _days * SECONDS_PER_DAY;  
450 | require(newTimestamp <= timestamp);  
451 | }
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
456 | returns (uint256 newTimestamp)
457 | {
458 |   newTimestamp = timestamp - _hours * SECONDS_PER_HOUR;
459 |   require(newTimestamp <= timestamp);
460 | }
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
456 | returns (uint256 newTimestamp)
457 | {
458 |   newTimestamp = timestamp - _hours * SECONDS_PER_HOUR;
459 |   require(newTimestamp <= timestamp);
460 | }
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
465 | returns (uint256 newTimestamp)
466 | {
467 |   newTimestamp = timestamp - _minutes * SECONDS_PER_MINUTE;
468 |   require(newTimestamp <= timestamp);
469 | }
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
465 | returns (uint256 newTimestamp)
466 | {
467 |     newTimestamp = timestamp - minutes * SECONDS_PER_MINUTE;
468 |     require(newTimestamp <= timestamp);
469 | }
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
474 | returns (uint256 newTimestamp)
475 | {
476 |     newTimestamp = timestamp - _seconds;
477 |     require(newTimestamp <= timestamp);
478 | }
```

## UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
491 | uint256 toDay;
492 | (fromYear, fromMonth, fromDay) = _daysToDate(
493 |     fromTimestamp / SECONDS_PER_DAY
494 | );
495 | (toYear, toMonth, toDay) = _daysToDate(toTimestamp / SECONDS_PER_DAY);
```

## UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
493 | fromTimestamp / SECONDS_PER_DAY
494 | );
495 | (toYear, toMonth, toDay) = _daysToDate(toTimestamp / SECONDS_PER_DAY);
496 | _years = toYear - fromYear;
497 | }
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
494 | );
495 | (toYear, toMonth, toDay) = _daysToDate(toTimestamp / SECONDS_PER_DAY);
496 | _years = toYear - fromYear;
497 | }
```

## UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
510 | uint256 toDay;
511 | (fromYear, fromMonth, fromDay) = _daysToDate(
512 |   fromTimestamp / SECONDS_PER_DAY
513 | );
514 | (toYear, toMonth, toDay) = _daysToDate(toTimestamp / SECONDS_PER_DAY);
```

## UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
512 | fromTimestamp / SECONDS_PER_DAY
513 | );
514 | (toYear, toMonth, toDay) = _daysToDate(toTimestamp / SECONDS_PER_DAY);
515 | _months = toYear * 12 + toMonth - fromYear * 12 - fromMonth;
516 | }
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
513 | );
514 | (toYear, toMonth, toDay) = _daysToDate(toTimestamp / SECONDS_PER_DAY);
515 | _months = toYear * 12 + toMonth - fromYear * 12 - fromMonth;
516 | }
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
513 | );
514 | (toYear, toMonth, toDay) = _daysToDate(toTimestamp / SECONDS_PER_DAY);
515 | _months = toYear * 12 + toMonth - fromYear * 12 - fromMonth;
516 | }
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
513 | );
514 | (toYear, toMonth, toDay) = _daysToDate(toTimestamp / SECONDS_PER_DAY);
515 | _months = toYear * 12 + toMonth - fromYear * 12 - fromMonth;
516 | }
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
513 | );  
514 | (toYear, toMonth, toDay) = _daysToDate(toTimestamp / SECONDS_PER_DAY);  
515 | _months = toYear * 12 + toMonth - fromYear * 12 - fromMonth;  
516 | }
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
513 | );  
514 | (toYear, toMonth, toDay) = _daysToDate(toTimestamp / SECONDS_PER_DAY);  
515 | _months = toYear * 12 + toMonth - fromYear * 12 - fromMonth;  
516 | }
```

## UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
522 | {  
523 |     require(fromTimestamp <= toTimestamp);  
524 |     _days = (toTimestamp - fromTimestamp) / SECONDS_PER_DAY;  
525 | }
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
522 | {  
523 |     require(fromTimestamp <= toTimestamp);  
524 |     _days = (toTimestamp - fromTimestamp) / SECONDS_PER_DAY;  
525 | }
```

## UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
531 | {  
532 |     require(fromTimestamp <= toTimestamp);  
533 |     _hours = (toTimestamp - fromTimestamp) / SECONDS_PER_HOUR;  
534 | }
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
531 | {  
532 |     require(fromTimestamp <= toTimestamp);  
533 |     _hours = (toTimestamp - fromTimestamp) / SECONDS_PER_HOUR;  
534 | }
```

## UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
540 | {  
541 |     require(fromTimestamp <= toTimestamp);  
542 |     _minutes = (toTimestamp - fromTimestamp) / SECONDS_PER_MINUTE;  
543 | }
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
540 | {  
541 |     require(fromTimestamp <= toTimestamp);  
542 |     _minutes = (toTimestamp - fromTimestamp) / SECONDS_PER_MINUTE;  
543 | }
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
549 | {  
550 |     require(fromTimestamp <= toTimestamp);  
551 |     _seconds = toTimestamp - fromTimestamp;  
552 | }  
553 | }
```

## UNKNOWN Compiler-rewritable "<uint> - 1" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
349 | (year, month, day) = _daysToDate(timestamp / SECONDS_PER_DAY);  
350 | month += _months;  
351 | year += (month - 1) / 12;  
352 | month = ((month - 1) % 12) + 1;  
353 | uint256 daysInMonth = _getDaysInMonth(year, month);
```

## UNKNOWN Compiler-rewritable "<uint> - 1" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
350 | month += _months;  
351 | year += (month - 1) / 12;  
352 | month = ((month - 1) % 12) + 1;  
353 | uint256 daysInMonth = _getDaysInMonth(year, month);  
354 | if (day > daysInMonth) {
```



## UNKNOWN Compiler-rewritable "<uint> - 1" discovered

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

SWC-101

Source file

contracts/external/calendar/BokkyPooBahsDateTimeLibrary.sol

Locations

```
428 | uint256 day;  
429 | (year, month, day) = _daysToDate(timestamp / SECONDS_PER_DAY);  
430 | uint256 yearMonth = year * 12 + (month - 1) - _months;  
431 | year = yearMonth / 12;  
432 | month = (yearMonth % 12) + 1;
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/oracle/ScalingPriceOracle.sol

Locations

```
111 | int256 oraclePriceInt = oraclePrice.toInt256();  
112 |  
113 | int256 timeDelta = Math.min(block.timestamp - startTime, TIMEFRAME).toInt256();  
114 | int256 pricePercentageChange = oraclePriceInt * monthlyChangeRateBasisPoints / Constants.BP_INT;  
115 | int256 priceDelta = pricePercentageChange * timeDelta / TIMEFRAME.toInt256();
```

## UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

contracts/oracle/ScalingPriceOracle.sol

Locations

```
112 |  
113 | int256 timeDelta = Math.min(block.timestamp - startTime, TIMEFRAME).toInt256();  
114 | int256 pricePercentageChange = oraclePriceInt * monthlyChangeRateBasisPoints / Constants.BP_INT;  
115 | int256 priceDelta = pricePercentageChange * timeDelta / TIMEFRAME.toInt256();
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

contracts/oracle/ScalingPriceOracle.sol

Locations

```
112 |  
113 | int256 timeDelta = Math.min(block.timestamp - startTime, TIMEFRAME).toInt256();  
114 | int256 pricePercentageChange = oraclePriceInt * monthlyChangeRateBasisPoints / Constants.BP_INT;  
115 | int256 priceDelta = pricePercentageChange * timeDelta / TIMEFRAME.toInt256();
```

## UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

contracts/oracle/ScalingPriceOracle.sol

Locations

```
113 | int256 timeDelta = Math.min(block.timestamp - startTime, TIMEFRAME).toInt256();
114 | int256 pricePercentageChange = oraclePriceInt * monthlyChangeRateBasisPoints / Constants.BP_INT;
115 | int256 priceDelta = pricePercentageChange * timeDelta / TIMEFRAME.toInt256();
116 |
117 | return (oraclePriceInt + priceDelta).toUint256();
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

contracts/oracle/ScalingPriceOracle.sol

Locations

```
113 | int256 timeDelta = Math.min(block.timestamp - startTime, TIMEFRAME).toInt256();
114 | int256 pricePercentageChange = oraclePriceInt * monthlyChangeRateBasisPoints / Constants.BP_INT;
115 | int256 priceDelta = pricePercentageChange * timeDelta / TIMEFRAME.toInt256();
116 |
117 | return (oraclePriceInt + priceDelta).toUint256();
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

contracts/oracle/ScalingPriceOracle.sol

Locations

```
115 | int256 priceDelta = pricePercentageChange * timeDelta / TIMEFRAME.toInt256();
116 |
117 | return (oraclePriceInt + priceDelta).toUint256();
118 | }
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/oracle/ScalingPriceOracle.sol

Locations

```
121 | /// @return percentageChange percentage change in basis points over past month
122 | function getMonthlyAPR() public view returns (int256 percentageChange) {
123 |     int256 delta = int128(currentMonth) - int128(previousMonth);
124 |     percentageChange = (delta * Constants.BP_INT) / int128(previousMonth);
125 | }
```

## UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

contracts/oracle/ScalingPriceOracle.sol

Locations

```
122 | function getMonthlyAPR() public view returns (int256 percentageChange) {
123 |     int256 delta = int128(currentMonth) - int128(previousMonth);
124 |     percentageChange = (delta * Constants.BP_INT) / int128(previousMonth);
125 | }
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

contracts/oracle/ScalingPriceOracle.sol

Locations

```
122 | function getMonthlyAPR() public view returns (int256 percentageChange) {
123 |     int256 delta = int128(currentMonth) - int128(previousMonth);
124 |     percentageChange = (delta * Constants.BP_INT) / int128(previousMonth);
125 | }
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/utils/Deviation.sol

Locations

```
20 | returns (uint256)
21 | {
22 |     int256 delta = a - b;
23 |     int256 basisPoints = (delta * Constants.BP_INT) / a;
```

## UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

contracts/utils/Deviation.sol

Locations

```
21 | {  
22 |   int256 delta = a - b;  
23 |   int256 basisPoints = (delta * Constants.BP_INT) / a;  
24 |  
25 |   return (basisPoints < 0 ? basisPoints * -1 : basisPoints).toUint256();
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

contracts/utils/Deviation.sol

Locations

```
21 | {  
22 |   int256 delta = a - b;  
23 |   int256 basisPoints = (delta * Constants.BP_INT) / a;  
24 |  
25 |   return (basisPoints < 0 ? basisPoints * -1 : basisPoints).toUint256();
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

contracts/utils/Deviation.sol

Locations

```
23 |   int256 basisPoints = (delta * Constants.BP_INT) / a;  
24 |  
25 |   return (basisPoints < 0 ? basisPoints * -1 : basisPoints).toUint256();  
26 | }
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/utils/Timed.sol

Locations

```
44 | /// @return remaining  
45 | function remainingTime() public view returns (uint256) {  
46 |   return duration - timeSinceStart(); // duration always >= timeSinceStart which is on [0,d]  
47 | }
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/utils/Timed.sol

Locations

```
55 | }
56 | uint256 _duration = duration;
57 | uint256 timePassed = block.timestamp - startTime; // block timestamp always >= startTime
58 | return timePassed > _duration ? _duration : timePassed;
59 | }
```

## UNKNOWN Arithmetic operation "++" discovered

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

SWC-101

Source file

node\_modules/@chainlink/contracts/src/v0.8/Chainlink.sol

Locations

```
128 | self.buf.encodeString(key);
129 | self.buf.startArray();
130 | for (uint256 i = 0; i < values.length; i++) {
131 |     self.buf.encodeString(values[i]);
132 | }
```

## UNKNOWN Arithmetic operation "\*\*\*" discovered

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

SWC-101

Source file

node\_modules/@chainlink/contracts/src/v0.8/ChainlinkClient.sol

Locations

```
18 | using Chainlink for Chainlink.Request;
19 |
20 | uint256 internal constant LINK_DIVISIBILITY = 10**18;
21 | uint256 private constant AMOUNT_OVERRIDE = 0;
22 | address private constant SENDER_OVERRIDE = address(0);
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

node\_modules/@chainlink/contracts/src/v0.8/ChainlinkClient.sol

Locations

```
96 | } internal returns (bytes32 requestId) {  
97 |     uint256 nonce = s_requestCount;  
98 |     s_requestCount = nonce + 1;  
99 |     bytes memory encodedRequest = abi.encodeWithSelector(  
100 |         ChainlinkRequestInterface.oracleRequest.selector,
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

node\_modules/@chainlink/contracts/src/v0.8/ChainlinkClient.sol

Locations

```
140 | } internal returns (bytes32 requestId) {  
141 |     uint256 nonce = s_requestCount;  
142 |     s_requestCount = nonce + 1;  
143 |     bytes memory encodedRequest = abi.encodeWithSelector(  
144 |         OperatorInterface.operatorRequest.selector,
```

## UNKNOWN Arithmetic operation "%" discovered

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

SWC-101

Source file

node\_modules/@chainlink/contracts/src/v0.8/vendor/BufferChainlink.sol

Locations

```
28 | */  
29 | function init(buffer memory buf, uint256 capacity) internal pure returns (buffer memory) {  
30 |     if (capacity % 32 != 0) {  
31 |         capacity += 32 - (capacity % 32);  
32 |     }
```

## UNKNOWN Arithmetic operation "+=" discovered

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

SWC-101

Source file

node\_modules/@chainlink/contracts/src/v0.8/vendor/BufferChainlink.sol

Locations

```
29 | function init(buffer memory buf, uint256 capacity) internal pure returns (buffer memory) {  
30 |     if (capacity % 32 != 0) {  
31 |         capacity += 32 - (capacity % 32);  
32 |     }  
33 |     // Allocate space for the buffer data
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

node\_modules/@chainlink/contracts/src/v0.8/vendor/BufferChainlink.sol

Locations

```
29 | function init(buffer memory buf, uint256 capacity) internal pure returns (buffer memory) {  
30 |     if (capacity % 32 != 0) {  
31 |         capacity += 32 - (capacity % 32);  
32 |     }  
33 |     // Allocate space for the buffer data
```

## UNKNOWN Arithmetic operation "%" discovered

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

SWC-101

Source file

node\_modules/@chainlink/contracts/src/v0.8/vendor/BufferChainlink.sol

Locations

```
29 | function init(buffer memory buf, uint256 capacity) internal pure returns (buffer memory) {  
30 |     if (capacity % 32 != 0) {  
31 |         capacity += 32 - (capacity % 32);  
32 |     }  
33 |     // Allocate space for the buffer data
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

node\_modules/@chainlink/contracts/src/v0.8/vendor/BufferChainlink.sol

Locations

```
98 | require(len <= data.length);
99 |
100 | if (off + len > buf.capacity) {
101 |     resize(buf, max(buf.capacity, len + off) * 2);
102 | }
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

node\_modules/@chainlink/contracts/src/v0.8/vendor/BufferChainlink.sol

Locations

```
99 |
100 | if (off + len > buf.capacity) {
101 |     resize(buf, max(buf.capacity, len + off) * 2);
102 | }
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

node\_modules/@chainlink/contracts/src/v0.8/vendor/BufferChainlink.sol

Locations

```
99 |
100 | if (off + len > buf.capacity) {
101 |     resize(buf, max(buf.capacity, len + off) * 2);
102 | }
```

## UNKNOWN Arithmetic operation "-=" discovered

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

SWC-101

Source file

node\_modules/@chainlink/contracts/src/v0.8/vendor/BufferChainlink.sol

Locations

```
119 |
120 | // Copy word-length chunks while possible
121 | for (; len >= 32; len -= 32) {
122 |     assembly {
123 |         mstore(dest, mload(src))
```



## UNKNOWN Arithmetic operation "+=" discovered

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

SWC-101

Source file

node\_modules/@chainlink/contracts/src/v0.8/vendor/BufferChainlink.sol

Locations

```
123 | mstore(dest, mload(src))
124 | }
125 | dest += 32;
126 | src += 32;
127 | }
```

## UNKNOWN Arithmetic operation "+=" discovered

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

SWC-101

Source file

node\_modules/@chainlink/contracts/src/v0.8/vendor/BufferChainlink.sol

Locations

```
124 | }
125 | dest += 32;
126 | src += 32;
127 | }
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

node\_modules/@chainlink/contracts/src/v0.8/vendor/BufferChainlink.sol

Locations

```
129 | // Copy remaining bytes
130 | unchecked {
131 | uint256 mask = 256**32 - len - 1;
132 | assembly {
133 | let srcpart := and(mload(src), not(mask))
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

node\_modules/@chainlink/contracts/src/v0.8/vendor/BufferChainlink.sol

Locations

```
129 | // Copy remaining bytes
130 | unchecked {
131 |     uint256 mask = (256**32 - len) - 1;
132 |     assembly {
133 |         let srcpart := and(mload(src), not(mask))
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

node\_modules/@chainlink/contracts/src/v0.8/vendor/BufferChainlink.sol

Locations

```
129 | // Copy remaining bytes
130 | unchecked {
131 |     uint256 mask = (256**32 - len) - 1;
132 |     assembly {
133 |         let srcpart := and(mload(src), not(mask))
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

node\_modules/@chainlink/contracts/src/v0.8/vendor/BufferChainlink.sol

Locations

```
181 | ) internal pure returns (buffer memory) {
182 |     if (off >= buf.capacity) {
183 |         resize(buf, buf.capacity * 2);
184 |     }
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

node\_modules/@chainlink/contracts/src/v0.8/vendor/BufferChainlink.sol

Locations

```
226 | uint256 len
227 | ) private pure returns (buffer memory) {
228 | if (len + off > buf.capacity) {
229 |     resize(buf, (len + off) * 2);
230 | }
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

node\_modules/@chainlink/contracts/src/v0.8/vendor/BufferChainlink.sol

Locations

```
227 | ) private pure returns (buffer memory) {
228 | if (len + off > buf.capacity) {
229 |     resize(buf, len + off * 2);
230 | }
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

node\_modules/@chainlink/contracts/src/v0.8/vendor/BufferChainlink.sol

Locations

```
227 | ) private pure returns (buffer memory) {
228 | if (len + off > buf.capacity) {
229 |     resize(buf, len + off * 2);
230 | }
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

node\_modules/@chainlink/contracts/src/v0.8/vendor/BufferChainlink.sol

Locations

```
231 |
232 | unchecked {
233 |     uint256 mask = 256**len - 1;
234 |     // Right-align data
235 |     data = data >> (8 * (32 - len));
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

node\_modules/@chainlink/contracts/src/v0.8/vendor/BufferChainlink.sol

Locations

```
231 |
232 | unchecked {
233 |     uint256 mask = (256**len) - 1;
234 |     // Right-align data
235 |     data = data >> (8 * (32 - len));
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

node\_modules/@chainlink/contracts/src/v0.8/vendor/BufferChainlink.sol

Locations

```
233 |     uint256 mask = (256**len) - 1;
234 |     // Right-align data
235 |     data = data >> (8 * (32 - len));
236 |     assembly {
237 |         // Memory address of the buffer data
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

node\_modules/@chainlink/contracts/src/v0.8/vendor/BufferChainlink.sol

Locations

```
233 |     uint256 mask = (256**len) - 1;
234 |     // Right-align data
235 |     data = data >> (8 * (32 - len));
236 |     assembly {
237 |         // Memory address of the buffer data
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

node\_modules/@chainlink/contracts/src/v0.8/vendor/BufferChainlink.sol

Locations

```
302 | uint256 len
303 | ) private pure returns (buffer memory) {
304 | if (len + off > buf.capacity) {
305 |     resize(buf, (len + off) * 2);
306 | }
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

node\_modules/@chainlink/contracts/src/v0.8/vendor/BufferChainlink.sol

Locations

```
303 | ) private pure returns (buffer memory) {
304 | if (len + off > buf.capacity) {
305 |     resize(buf, len + off * 2);
306 | }
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

node\_modules/@chainlink/contracts/src/v0.8/vendor/BufferChainlink.sol

Locations

```
303 | ) private pure returns (buffer memory) {
304 | if (len + off > buf.capacity) {
305 |     resize(buf, (len + off) * 2);
306 | }
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

node\_modules/@chainlink/contracts/src/v0.8/vendor/BufferChainlink.sol

Locations

```
306 | }
307 |
308 | uint256 mask = 256**len - 1;
309 | assembly {
310 |     // Memory address of the buffer data
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

node\_modules/@chainlink/contracts/src/v0.8/vendor/BufferChainlink.sol

Locations

```
306 | }  
307 |  
308 | uint256 mask = (256**len) - 1;  
309 | assembly {  
310 | // Memory address of the buffer data
```

## UNKNOWN Compiler-rewritable "<uint> - 1" discovered

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

SWC-101

Source file

node\_modules/@chainlink/contracts/src/v0.8/vendor/BufferChainlink.sol

Locations

```
129 | // Copy remaining bytes  
130 | unchecked {  
131 | uint256 mask = (256**32 - len) - 1;  
132 | assembly {  
133 | let srcpart := and(mload(src), not(mask))
```

## UNKNOWN Compiler-rewritable "<uint> - 1" discovered

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

SWC-101

Source file

node\_modules/@chainlink/contracts/src/v0.8/vendor/BufferChainlink.sol

Locations

```
231 |  
232 | unchecked {  
233 | uint256 mask = (256**len) - 1;  
234 | // Right-align data  
235 | data = data >> (8 * (32 - len));
```

## UNKNOWN Compiler-rewritable "<uint> - 1" discovered

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

SWC-101

Source file

node\_modules/@chainlink/contracts/src/v0.8/vendor/BufferChainlink.sol

Locations

```
306 | }  
307 |  
308 | uint256 mask = [256**len] - 1;  
309 | assembly {  
310 | // Memory address of the buffer data
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

node\_modules/@chainlink/contracts/src/v0.8/vendor/CBORChainlink.sol

Locations

```
57 | encodeFixedNumeric(buf, MAJOR_TYPE_INT, uint64(uint256(value)));  
58 | } else {  
59 | encodeFixedNumeric(buf, MAJOR_TYPE_NEGATIVE_INT, uint64(uint256(-1 - value)));  
60 | }  
61 | }
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

node\_modules/@chainlink/contracts/src/v0.8/vendor/CBORChainlink.sol

Locations

```
73 | function encodeSignedBigNum(BufferChainlink.buffer memory buf, int input) internal pure {  
74 | buf.appendUint8(uint8((MAJOR_TYPE_TAG << 5) | TAG_TYPE_NEGATIVE_BIGNUM));  
75 | encodeBytes(buf, abi.encode(uint256(-1 - input)));  
76 | }
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol

Locations

```
158 | require(currentAllowance >= amount, "ERC20: transfer amount exceeds allowance");  
159 | unchecked {  
160 | _approve(sender, _msgSender(), currentAllowance - amount);  
161 | }
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol

Locations

```
177 | */
178 | function increaseAllowance(address spender, uint256 addedValue) public virtual returns (bool) {
179 |     _approve(_msgSender(), spender, allowances[_msgSender()][spender] + addedValue);
180 |     return true;
181 | }
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol

Locations

```
199 | require(currentAllowance >= subtractedValue, "ERC20: decreased allowance below zero");
200 | unchecked {
201 |     _approve(_msgSender(), spender, currentAllowance - subtractedValue);
202 | }
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol

Locations

```
232 | require(senderBalance >= amount, "ERC20: transfer amount exceeds balance");
233 | unchecked {
234 |     _balances[sender] = senderBalance - amount;
235 | }
236 | _balances[recipient] += amount;
```



## UNKNOWN Arithmetic operation "+=" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol

Locations

```
234 | _balances[sender] = senderBalance - amount;
235 | }
236 | balances[recipient] += amount;
237 |
238 | emit Transfer(sender, recipient, amount);
```

## UNKNOWN Arithmetic operation "+=" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol

Locations

```
255 | _beforeTokenTransfer(address(0), account, amount);
256 |
257 | totalSupply += amount;
258 | _balances[account] += amount;
259 | emit Transfer(address(0), account, amount);
```

## UNKNOWN Arithmetic operation "+=" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol

Locations

```
256 |
257 | _totalSupply += amount;
258 | balances[account] += amount;
259 | emit Transfer(address(0), account, amount);
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol

Locations

```
281 | require(accountBalance >= amount, "ERC20: burn amount exceeds balance");
282 | unchecked {
283 |   _balances[account] = accountBalance - amount;
284 | }
285 | _totalSupply -= amount;
```

## UNKNOWN Arithmetic operation "-=" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol

Locations

```
283 | _balances[account] = accountBalance - amount;
284 | }
285 | totalSupply -= amount;
286 |
287 | emit Transfer(account, address(0), amount);
```

## UNKNOWN Arithmetic operation "++" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/Strings.sol

Locations

```
23 | uint256 digits;
24 | while (temp != 0) {
25 |   digits++;
26 |   temp /= 10;
27 | }
```

## UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/Strings.sol

Locations

```
24 | while (temp != 0) {
25 |     digits++;
26 |     temp /= 10;
27 | }
28 | bytes memory buffer = new bytes(digits);
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/Strings.sol

Locations

```
28 | bytes memory buffer = new bytes(digits);
29 | while (value != 0) {
30 |     digits -= 1;
31 |     buffer[digits] = bytes1(uint8(48 + uint256(value % 10)));
32 |     value /= 10;
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/Strings.sol

Locations

```
29 | while (value != 0) {
30 |     digits -= 1;
31 |     buffer[digits] = bytes1(uint8(48 + uint256(value % 10)));
32 |     value /= 10;
33 | }
```

## UNKNOWN Arithmetic operation "%" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/Strings.sol

Locations

```
29 | while (value != 0) {  
30 |     digits -= 1;  
31 |     buffer[digits] = bytes1(uint8(48 + uint256(value % 10)));  
32 |     value /= 10;  
33 | }
```

## UNKNOWN Arithmetic operation "/"=" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/Strings.sol

Locations

```
30 |     digits -= 1;  
31 |     buffer[digits] = bytes1(uint8(48 + uint256(value % 10)));  
32 |     value /= 10;  
33 | }  
34 | return string(buffer);
```

## UNKNOWN Arithmetic operation "++" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/Strings.sol

Locations

```
45 | uint256 length = 0;  
46 | while (temp != 0) {  
47 |     length++;  
48 |     temp >>= 8;  
49 | }
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/Strings.sol

Locations

```
55 | */
56 | function toHexString(uint256 value, uint256 length) internal pure returns (string memory) {
57 |     bytes memory buffer = new bytes(2 * length + 2);
58 |     buffer[0] = "0";
59 |     buffer[1] = "x";
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/Strings.sol

Locations

```
55 | */
56 | function toHexString(uint256 value, uint256 length) internal pure returns (string memory) {
57 |     bytes memory buffer = new bytes(2 * length + 2);
58 |     buffer[0] = "0";
59 |     buffer[1] = "x";
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/Strings.sol

Locations

```
58 |     buffer[0] = "0";
59 |     buffer[1] = "x";
60 |     for (uint256 i = 2 * length + 1; i > 1; --i) {
61 |         buffer[i] = _HEX_SYMBOLS[value & 0xf];
62 |         value >>= 4;
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/Strings.sol

Locations

```
58 | buffer[0] = "0";
59 | buffer[1] = "x";
60 | for (uint256 i = 2 * length + 1; i > 1; --i) {
61 |     buffer[i] = _HEX_SYMBOLS[value & 0xf];
62 |     value >>= 4;
```

## UNKNOWN Arithmetic operation "--" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/Strings.sol

Locations

```
58 | buffer[0] = "0";
59 | buffer[1] = "x";
60 | for (uint256 i = 2 * length + 1; i > 1; --i) {
61 |     buffer[i] = _HEX_SYMBOLS[value & 0xf];
62 |     value >>= 4;
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/math/Math.sol

Locations

```
28 | function average(uint256 a, uint256 b) internal pure returns (uint256) {
29 |     // (a + b) / 2 can overflow.
30 |     return (a & b) + (a ^ b) / 2;
31 | }
```

## UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/math/Math.sol

Locations

```
28 | function average(uint256 a, uint256 b) internal pure returns (uint256) {
29 |     // (a + b) / 2 can overflow.
30 |     return (a & b) + (a ^ b) / 2;
31 | }
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/math/Math.sol

Locations

```
39 | function ceilDiv(uint256 a, uint256 b) internal pure returns (uint256) {
40 |     // (a + b - 1) / b can overflow on addition, so we distribute.
41 |     return a / b + (a % b == 0 ? 0 : 1);
42 | }
43 |
```

## UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/math/Math.sol

Locations

```
39 | function ceilDiv(uint256 a, uint256 b) internal pure returns (uint256) {
40 |     // (a + b - 1) / b can overflow on addition, so we distribute.
41 |     return a / b + (a % b == 0 ? 0 : 1);
42 | }
43 |
```

## UNKNOWN Arithmetic operation "%" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/math/Math.sol

Locations

```
39 | function ceilDiv(uint256 a, uint256 b) internal pure returns (uint256) {
40 |     // (a + b - 1) / b can overflow on addition, so we distribute.
41 |     return a / b + (a % b == 0 ? 0 : 1);
42 | }
43 |
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/math/SafeMath.sol

Locations

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

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/math/SafeMath.sol

Locations

```
36 |     unchecked {  
37 |         if (b > a) return (false, 0);  
38 |         return (true, a - b);  
39 |     }  
40 | }
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/math/SafeMath.sol

Locations

```
51 | // See: https://github.com/OpenZeppelin/openzeppelin-contracts/pull/522  
52 | if (a == 0) return (true, 0);  
53 | uint256 c = a * b;  
54 | if (c / a != b) return (false, 0);  
55 | return (true, c);
```



## UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/math/SafeMath.sol

Locations

```
52 | if (a == 0) return (true, 0);
53 | uint256 c = a * b;
54 | if (c / a != b) return (false, 0);
55 | return (true, c);
56 | }
```

## UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/math/SafeMath.sol

Locations

```
65 | unchecked {
66 |   if (b == 0) return (false, 0);
67 |   return (true, a / b);
68 | }
69 | }
```

## UNKNOWN Arithmetic operation "%" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/math/SafeMath.sol

Locations

```
77 | unchecked {
78 |   if (b == 0) return (false, 0);
79 |   return (true, a % b);
80 | }
81 | }
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/math/SafeMath.sol

Locations

```
92 | */
93 | function add(uint256 a, uint256 b) internal pure returns (uint256) {
94 |     return a + b;
95 | }
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/math/SafeMath.sol

Locations

```
106 | */
107 | function sub(uint256 a, uint256 b) internal pure returns (uint256) {
108 |     return a - b;
109 | }
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/math/SafeMath.sol

Locations

```
120 | */
121 | function mul(uint256 a, uint256 b) internal pure returns (uint256) {
122 |     return a * b;
123 | }
```

## UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/math/SafeMath.sol

Locations

```
134 | */
135 | function div(uint256 a, uint256 b) internal pure returns (uint256) {
136 |     return a / b;
137 | }
```

## UNKNOWN Arithmetic operation "%" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/math/SafeMath.sol

Locations

```
150 | */
151 | function mod(uint256 a, uint256 b) internal pure returns (uint256) {
152 |     return a % b;
153 | }
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/math/SafeMath.sol

Locations

```
173 | unchecked {
174 |     require(b <= a, errorMessage);
175 |     return a - b;
176 | }
177 | }
```

## UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/math/SafeMath.sol

Locations

```
196 | unchecked {
197 |     require(b > 0, errorMessage);
198 |     return a / b;
199 | }
200 | }
```

## UNKNOWN Arithmetic operation "%" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/math/SafeMath.sol

Locations

```
222 | unchecked {  
223 |     require(b > 0, errorMessage);  
224 |     return a % b;  
225 | }  
226 | }
```

LOW

A floating pragma is set.

The current pragma Solidity directive is ""^0.8.4"". 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

contracts/oracle/ScalingPriceOracle.sol

Locations

```
1 | // SPDX-License-Identifier: GPL-3.0-or-later  
2 | pragma solidity ^0.8.4;  
3 |  
4 | import {Timed} from "../utils/Timed.sol";
```

## UNKNOWN Out of bounds array access

The index access expression can cause an exception in case of use of invalid array index value.

SWC-110

Source file

node\_modules/@chainlink/contracts/src/v0.8/Chainlink.sol

Locations

```
129 | self.buf.startArray();  
130 | for (uint256 i = 0; i < values.length; i++) {  
131 |     self.buf.encodeString(values[i]);  
132 | }  
133 | self.buf.endSequence();
```

## UNKNOWN Out of bounds array access

The index access expression can cause an exception in case of use of invalid array index value.

SWC-110

Source file

node\_modules/@openzeppelin/contracts/utils/Strings.sol

Locations

```
29 | while (value != 0) {
30 |     digits -= 1;
31 |     buffer[digits] = bytes1(uint8(48 + uint256(value % 10)));
32 |     value /= 10;
33 | }
```

## UNKNOWN Out of bounds array access

The index access expression can cause an exception in case of use of invalid array index value.

SWC-110

Source file

node\_modules/@openzeppelin/contracts/utils/Strings.sol

Locations

```
56 | function toHexString(uint256 value, uint256 length) internal pure returns (string memory) {
57 |     bytes memory buffer = new bytes(2 * length + 2);
58 |     buffer[0] = "0";
59 |     buffer[1] = "x";
60 |     for (uint256 i = 2 * length + 1; i > 1; --i) {
```

## UNKNOWN Out of bounds array access

The index access expression can cause an exception in case of use of invalid array index value.

SWC-110

Source file

node\_modules/@openzeppelin/contracts/utils/Strings.sol

Locations

```
57 | bytes memory buffer = new bytes(2 * length + 2);
58 | buffer[0] = "0";
59 | buffer[1] = "x";
60 | for (uint256 i = 2 * length + 1; i > 1; --i) {
61 |     buffer[i] = _HEX_SYMBOLS[value & 0xf];
```

## UNKNOWN Out of bounds array access

The index access expression can cause an exception in case of use of invalid array index value.

SWC-110

Source file

node\_modules/@openzeppelin/contracts/utils/Strings.sol

Locations

```
59 | buffer[1] = "x";
60 | for (uint256 i = 2 * length + 1; i > 1; --i) {
61 |     buffer[i] = _HEX_SYMBOLS[value & 0xf];
62 |     value >>= 4;
63 | }
```

## UNKNOWN Out of bounds array access

The index access expression can cause an exception in case of use of invalid array index value.

SWC-110

Source file




node\_modules/@openzeppelin/contracts/utils/Strings.sol

Locations

```
59 | buffer[1] = "x";
60 | for (uint256 i = 2 * length + 1; i > 1; --i) {
61 |     buffer[i] = _HEX_SYMBOLS[value & 0xf];
62 |     value >>= 4;
63 | }
```

Started	Tue Mar 29 2022 23:26:38 GMT+0000 (Coordinated Universal Time)
Finished	Wed Mar 30 2022 00:11:53 GMT+0000 (Coordinated Universal Time)
Mode	Deep
Client Tool	Mythx-Cli-0.6.22
Main Source File	Contracts/Oracle/OraclePassThrough.sol

DETECTED VULNERABILITIES

 HIGH	 MEDIUM	 LOW
0	0	2

ISSUES

LOW

A floating pragma is set.

SWC-103

The current pragma Solidity directive is ""^0.8.4"". 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/oracle/OraclePassThrough.sol  
Locations

```
1 | // SPDX-License-Identifier: GPL-3.0-or-later
2 | pragma solidity ^0.8.4;
3 |
4 | import {Decimal} from "../external/Decimal.sol";
```

LOW

Requirement violation.

A requirement was violated in a nested call and the call was reverted as a result. Make sure valid inputs are provided to the nested call (for instance, via passed arguments).

SWC-123

Source file

contracts/oracle/OraclePassThrough.sol

Locations

```
44 | /// @notice function to get the current oracle price for the entire system
45 | function getCurrentOraclePrice() external view override returns (uint256) {
46 |     return scalingPriceOracle.getCurrentOraclePrice();
47 | }
```

Source file

contracts/oracle/OraclePassThrough.sol

Locations

```
44 | /// @notice function to get the current oracle price for the entire system
45 | function getCurrentOraclePrice() external view override returns (uint256) {
46 |     return scalingPriceOracle.getCurrentOraclePrice();
47 | }
```



Started	Tue Mar 29 2022 23:26:38 GMT+0000 (Coordinated Universal Time)
Finished	Tue Mar 29 2022 23:26:48 GMT+0000 (Coordinated Universal Time)
Mode	Deep
Client Tool	Mythx-Cli-0.6.22
Main Source File	Contracts/Peg/NonCustodialPSM.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

contracts/external/Decimal.sol

Locations

```
28 | // ===== Constants =====
29 |
30 | uint256 private constant BASE = 10**18;
31 |
32 | // ===== Structs =====
```

## UNKNOWN Arithmetic operation "++" discovered

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

SWC-101

Source file

contracts/external/Decimal.sol

Locations

```
106 |
107 | D256 memory temp = D256({value: self.value});
108 | for (uint256 i = 1; i < b; i++) {
109 |     temp = mul(temp, self);
110 | }
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/peg/NonCustodialPSM.sol

Locations

```
288 | amountVoltOut
289 | );
290 | uint256 amountFeiToMint = amountVoltOut - amountFeiToTransfer;
291 |
292 | if (amountFeiToTransfer != 0) {
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

contracts/peg/NonCustodialPSM.sol

Locations

```
339 | function getMaxMintAmountOut() external view override returns (uint256) {
340 |     return
341 |         volt().balanceOf(address(this)) +
342 |         rateLimitedMinter.individualBuffer(address(this));
343 | }
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/peg/NonCustodialPSM.sol

Locations

```
361 |
362 | amountVoltOut = adjustedAmountIn
363 |     .mul(Constants.BASIS_POINTS_GRANULARITY - mintFeeBasisPoints)
364 |     .div(Constants.BASIS_POINTS_GRANULARITY)
365 |     .asUint256();
```

## UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

contracts/peg/NonCustodialPSM.sol

Locations

```
381 | /// get amount of VOLT being provided being redeemed after fees
382 | Decimal.D256 memory adjustedAmountIn = Decimal.from(
383 |     amountVoltIn *
384 |     (Constants.BASIS_POINTS_GRANULARITY - redeemFeeBasisPoints)) /
385 |     Constants.BASIS_POINTS_GRANULARITY
386 | );
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

contracts/peg/NonCustodialPSM.sol

Locations

```
381 | /// get amount of VOLT being provided being redeemed after fees
382 | Decimal.D256 memory adjustedAmountIn = Decimal.from(
383 |     (amountVoltIn *
384 |     (Constants.BASIS_POINTS_GRANULARITY - redeemFeeBasisPoints)) /
385 |     Constants.BASIS_POINTS_GRANULARITY
386 | );
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/peg/NonCustodialPSM.sol

Locations

```
382 | Decimal.D256 memory adjustedAmountIn = Decimal.from(
383 |     (amountVoltIn *
384 |     (Constants.BASIS_POINTS_GRANULARITY - redeemFeeBasisPoints)) /
385 |     Constants.BASIS_POINTS_GRANULARITY
386 | );
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

contracts/refs/OracleRef.sol

Locations

```
109 | uint256 scalingFactor;  
110 | if (decimalsNormalizer < 0) {  
111 | scalingFactor = 10**(-1 * decimalsNormalizer).toUint256();  
112 | _peg = _peg.div(scalingFactor);  
113 | } else {
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

contracts/refs/OracleRef.sol

Locations

```
109 | uint256 scalingFactor;  
110 | if (decimalsNormalizer < 0) {  
111 | scalingFactor = 10**(-1 * decimalsNormalizer).toUint256();  
112 | _peg = _peg.div(scalingFactor);  
113 | } else {
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

contracts/refs/OracleRef.sol

Locations

```
112 | _peg = _peg.div(scalingFactor);  
113 | } else {  
114 | scalingFactor = 10**decimalsNormalizer.toUint256();  
115 | _peg = _peg.mul(scalingFactor);  
116 | }
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

contracts/refs/OracleRef.sol

Locations

```
142 |
143 | if (oldDoInvert != newDoInvert) {
144 |     _setDecimalsNormalizer(-1 * decimalsNormalizer);
145 | }
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/refs/OracleRef.sol

Locations

```
159 | function _setDecimalsNormalizerFromToken(address token) internal {
160 |     int256 feiDecimals = 18;
161 |     int256 _decimalsNormalizer = feiDecimals -
162 |     int256(uint256(IERC20Metadata(token).decimals()));
163 |
164 |     if (doInvert) {
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

contracts/refs/OracleRef.sol

Locations

```
163 |
164 | if (doInvert) {
165 |     _decimalsNormalizer = -1 * _decimalsNormalizer;
166 | }
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/utils/MultiRateLimited.sol

Locations

```
212 | ];  
213 |  
214 | uint256 elapsed = block.timestamp - rateLimitData.lastBufferUsedTime;  
215 | return  
216 | uint112(
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

contracts/utils/MultiRateLimited.sol

Locations

```
216 | uint112(  
217 | Math.min(  
218 | rateLimitData.bufferStored +  
219 | rateLimitData.rateLimitPerSecond * elapsed ,  
220 | rateLimitData.bufferCap  
221 | )
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

contracts/utils/MultiRateLimited.sol

Locations

```
217 | Math.min(  
218 | rateLimitData.bufferStored +  
219 | (rateLimitData.rateLimitPerSecond * elapsed),  
220 | rateLimitData.bufferCap  
221 | )
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/utils/MultiRateLimited.sol

Locations

```
339 |  
340 | rateLimitPerAddress[rateLimitedAddress].bufferStored = uint112(  
341 | newBuffer - amount  
342 | );
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/utils/MultiRateLimited.sol

Locations

```
349 | rateLimitedAddress,  
350 | amount,  
351 | newBuffer - amount  
352 | );
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/utils/RateLimited.sol

Locations

```
81 | /// @dev replenishes at rateLimitPerSecond per second up to bufferCap  
82 | function buffer() public view returns (uint256) {  
83 | uint256 elapsed = block.timestamp - lastBufferUsedTime;  
84 | return  
85 | Math.min(bufferStored + (rateLimitPerSecond * elapsed), bufferCap);
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

contracts/utils/RateLimited.sol

Locations

```
83 | uint256 elapsed = block.timestamp - lastBufferUsedTime;  
84 | return  
85 | Math.min(bufferStored + (rateLimitPerSecond * elapsed), bufferCap);  
86 | }
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

contracts/Utils/RateLimited.sol

Locations

```
83 | uint256 elapsed = block.timestamp - lastBufferUsedTime;
84 | return
85 | Math.min(bufferStored + (rateLimitPerSecond * elapsed), bufferCap);
86 | }
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/Utils/RateLimited.sol

Locations

```
104 | require(usedAmount <= newBuffer, "RateLimited: rate limit hit");
105 |
106 | bufferStored = newBuffer - usedAmount;
107 |
108 | lastBufferUsedTime = block.timestamp;
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

contracts/Utils/RateLimited.sol

Locations

```
126 |
127 | /// ensure that bufferStored cannot be gt buffer cap
128 | bufferStored = Math.min(newBuffer + amount, _bufferCap);
129 | }
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/token/ERC20/Utils/SafeERC20.sol

Locations

```
63 | uint256 value
64 | ) internal {
65 | uint256 newAllowance = token.allowance(address(this), spender) + value;
66 | _callOptionalReturn(token, abi.encodeWithSelector(token.approve.selector, spender, newAllowance));
67 | }
```



## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol

Locations

```
75 | uint256 oldAllowance = token.allowance(address(this), spender);
76 | require(oldAllowance >= value, "SafeERC20: decreased allowance below zero");
77 | uint256 newAllowance = oldAllowance - value;
78 | _callOptionalReturn(token, abi.encodeWithSelector(token.approve.selector, spender, newAllowance));
79 | }
```

## UNKNOWN Arithmetic operation "++" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/Strings.sol

Locations

```
23 | uint256 digits;
24 | while (temp != 0) {
25 |     digits++;
26 |     temp /= 10;
27 | }
```

## UNKNOWN Arithmetic operation "/"=" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/Strings.sol

Locations

```
24 | while (temp != 0) {
25 |     digits++;
26 |     temp /= 10;
27 | }
28 | bytes memory buffer = new bytes(digits);
```

## UNKNOWN Arithmetic operation "-=" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/Strings.sol

Locations

```
28 | bytes memory buffer = new bytes(digits);
29 | while (value != 0) {
30 |     digits -= 1;
31 |     buffer[digits] = bytes1(uint8(48 + uint256(value % 10)));
32 |     value /= 10;
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/Strings.sol

Locations

```
29 | while (value != 0) {
30 |     digits -= 1;
31 |     buffer[digits] = bytes1(uint8(48 + uint256(value % 10)));
32 |     value /= 10;
33 | }
```

## UNKNOWN Arithmetic operation "%" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/Strings.sol

Locations

```
29 | while (value != 0) {
30 |     digits -= 1;
31 |     buffer[digits] = bytes1(uint8(48 + uint256(value % 10)));
32 |     value /= 10;
33 | }
```

## UNKNOWN Arithmetic operation "/"= discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/Strings.sol

Locations

```
30 | digits -= 1;
31 | buffer[digits] = bytes1(uint8(48 + uint256(value % 10)));
32 | value /= 10;
33 | }
34 | return string(buffer);
```

## UNKNOWN Arithmetic operation "++" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/Strings.sol

Locations

```
45 | uint256 length = 0;
46 | while (temp != 0) {
47 |     length++;
48 |     temp >>= 8;
49 | }
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/Strings.sol

Locations

```
55 | */
56 | function toHexString(uint256 value, uint256 length) internal pure returns (string memory) {
57 |     bytes memory buffer = new bytes(2 * length + 2);
58 |     buffer[0] = "0";
59 |     buffer[1] = "x";
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/Strings.sol

Locations

```
55 | */
56 | function toHexString(uint256 value, uint256 length) internal pure returns (string memory) {
57 |     bytes memory buffer = new bytes(2 * length + 2);
58 |     buffer[0] = "0";
59 |     buffer[1] = "x";
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/Strings.sol

Locations

```
58 |     buffer[0] = "0";
59 |     buffer[1] = "x";
60 |     for (uint256 i = 2 * length + 1; i > 1; --i) {
61 |         buffer[i] = _HEX_SYMBOLS[value & 0xf];
62 |         value >>= 4;
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/Strings.sol

Locations

```
58 |     buffer[0] = "0";
59 |     buffer[1] = "x";
60 |     for (uint256 i = 2 * length + 1; i > 1; --i) {
61 |         buffer[i] = _HEX_SYMBOLS[value & 0xf];
62 |         value >>= 4;
```

## UNKNOWN Arithmetic operation "--" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/Strings.sol

Locations

```
58 | buffer[0] = "0";
59 | buffer[1] = "x";
60 | for (uint256 i = 2 * length + 1; i > 1; --i) {
61 |     buffer[i] = _HEX_SYMBOLS[value & 0xf];
62 |     value >>= 4;
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/math/Math.sol

Locations

```
28 | function average(uint256 a, uint256 b) internal pure returns (uint256) {
29 |     // (a + b) / 2 can overflow.
30 |     return (a & b) + (a ^ b) / 2;
31 | }
```

## UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/math/Math.sol

Locations

```
28 | function average(uint256 a, uint256 b) internal pure returns (uint256) {
29 |     // (a + b) / 2 can overflow.
30 |     return (a & b) + (a ^ b) / 2;
31 | }
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/math/Math.sol

Locations

```
39 | function ceilDiv(uint256 a, uint256 b) internal pure returns (uint256) {
40 |     // (a + b - 1) / b can overflow on addition, so we distribute.
41 |     return a / b + (a % b == 0 ? 0 : 1);
42 | }
43 | }
```

## UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/math/Math.sol

Locations

```
39 | function ceilDiv(uint256 a, uint256 b) internal pure returns (uint256) {
40 | // (a + b - 1) / b can overflow on addition, so we distribute.
41 | return a / b + (a % b == 0 ? 0 : 1);
42 | }
43 | }
```

## UNKNOWN Arithmetic operation "%" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/math/Math.sol

Locations

```
39 | function ceilDiv(uint256 a, uint256 b) internal pure returns (uint256) {
40 | // (a + b - 1) / b can overflow on addition, so we distribute.
41 | return a / b + (a % b == 0 ? 0 : 1);
42 | }
43 | }
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/math/SafeMath.sol

Locations

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

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/math/SafeMath.sol

Locations

```
36 | unchecked {  
37 |   if (b > a) return (false, 0);  
38 |   return (true, a - b);  
39 | }  
40 | }
```

## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/math/SafeMath.sol

Locations

```
51 | // See: https://github.com/OpenZeppelin/openzeppelin-contracts/pull/522  
52 | if (a == 0) return (true, 0);  
53 | uint256 c = a * b;  
54 | if (c / a != b) return (false, 0);  
55 | return (true, c);
```

## UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/math/SafeMath.sol

Locations

```
52 | if (a == 0) return (true, 0);  
53 | uint256 c = a * b;  
54 | if (c / a != b) return (false, 0);  
55 | return (true, c);  
56 | }
```

## UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/math/SafeMath.sol

Locations

```
65 | unchecked {  
66 |   if (b == 0) return (false, 0);  
67 |   return (true, a / b);  
68 | }  
69 | }
```

## UNKNOWN Arithmetic operation "%" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/math/SafeMath.sol

Locations

```
77 | unchecked {  
78 |   if (b == 0) return (false, 0);  
79 |   return (true, a % b);  
80 | }  
81 | }
```

## UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/math/SafeMath.sol

Locations

```
92 | */  
93 | function add(uint256 a, uint256 b) internal pure returns (uint256) {  
94 |   return a + b;  
95 | }
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/math/SafeMath.sol

Locations

```
106 | */  
107 | function sub(uint256 a, uint256 b) internal pure returns (uint256) {  
108 |   return a - b;  
109 | }
```



## UNKNOWN Arithmetic operation "\*" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/math/SafeMath.sol

Locations

```
120 | */
121 | function mul(uint256 a, uint256 b) internal pure returns (uint256) {
122 |     return a * b;
123 | }
```

## UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/math/SafeMath.sol

Locations

```
134 | */
135 | function div(uint256 a, uint256 b) internal pure returns (uint256) {
136 |     return a / b;
137 | }
```

## UNKNOWN Arithmetic operation "%" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/math/SafeMath.sol

Locations

```
150 | */
151 | function mod(uint256 a, uint256 b) internal pure returns (uint256) {
152 |     return a % b;
153 | }
```

## UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/math/SafeMath.sol

Locations

```
173 | unchecked {
174 |     require(b <= a, errorMessage);
175 |     return a - b;
176 | }
177 | }
```

## UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/math/SafeMath.sol

Locations

```
196 | unchecked {  
197 |   require(b > 0, errorMessage);  
198 |   return a / b;  
199 | }  
200 |
```

## UNKNOWN Arithmetic operation "%" discovered

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

SWC-101

Source file

node\_modules/@openzeppelin/contracts/utils/math/SafeMath.sol

Locations

```
222 | unchecked {  
223 |   require(b > 0, errorMessage);  
224 |   return a % b;  
225 | }  
226 |
```

LOW

A floating pragma is set.

The current pragma Solidity directive is ""^0.8.4"". 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

contracts/peg/NonCustodialPSM.sol

Locations

```
1 | // SPDX-License-Identifier: GPL-3.0-or-later  
2 | pragma solidity ^0.8.4;  
3 |  
4 | import {Decimal} from "../external/Decimal.sol";
```

## UNKNOWN Out of bounds array access

The index access expression can cause an exception in case of use of invalid array index value.

SWC-110

Source file

node\_modules/@openzeppelin/contracts/utils/Strings.sol

Locations

```
29 | while (value != 0) {  
30 |     digits -= 1;  
31 |     buffer[digits] = bytes1(uint8(48 + uint256(value % 10)));  
32 |     value /= 10;  
33 | }
```

## UNKNOWN Out of bounds array access

The index access expression can cause an exception in case of use of invalid array index value.

SWC-110

Source file

node\_modules/@openzeppelin/contracts/utils/Strings.sol

Locations

```
56 | function toHexString(uint256 value, uint256 length) internal pure returns (string memory) {  
57 |     bytes memory buffer = new bytes(2 * length + 2);  
58 |     buffer[0] = "0";  
59 |     buffer[1] = "x";  
60 |     for (uint256 i = 2 * length + 1; i > 1; --i) {
```

## UNKNOWN Out of bounds array access

The index access expression can cause an exception in case of use of invalid array index value.

SWC-110

Source file

node\_modules/@openzeppelin/contracts/utils/Strings.sol

Locations

```
57 | bytes memory buffer = new bytes(2 * length + 2);  
58 | buffer[0] = "0";  
59 | buffer[1] = "x";  
60 | for (uint256 i = 2 * length + 1; i > 1; --i) {  
61 |     buffer[i] = _HEX_SYMBOLS[value & 0xf];
```

## UNKNOWN Out of bounds array access

The index access expression can cause an exception in case of use of invalid array index value.

SWC-110

Source file

node\_modules/@openzeppelin/contracts/utils/Strings.sol

Locations

```
59 | buffer[1] = "x";
60 | for (uint256 i = 2 * length + 1; i > 1; --i) {
61 |     buffer[i] = _HEX_SYMBOLS[value & 0xf];
62 |     value >>= 4;
63 | }
```

## UNKNOWN Out of bounds array access

The index access expression can cause an exception in case of use of invalid array index value.

SWC-110

Source file

node\_modules/@openzeppelin/contracts/utils/Strings.sol

Locations

```
59 | buffer[1] = "x";
60 | for (uint256 i = 2 * length + 1; i > 1; --i) {
61 |     buffer[i] = _HEX_SYMBOLS[value & 0xf];
62 |     value >>= 4;
63 | }
```