




Started	Mon Oct 09 2023 07:13:17 GMT+0000 (Coordinated Universal Time)
Finished	Mon Oct 09 2023 07:13:24 GMT+0000 (Coordinated Universal Time)
Mode	Deep
Client Tool	Mythx-Cli-0.7.3
Main Source File	Contracts/FeeCollector.Sol

DETECTED VULNERABILITIES

 HIGH	 MEDIUM	 LOW
0	0	0

ISSUES

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

```
60 | function safeIncreaseAllowance(IERC20 token, address spender, uint256 value) internal {
61 |     uint256 oldAllowance = token.allowance(address(this), spender);
62 |     _callOptionalReturn(token, abi.encodeWithSelector(token.approve.selector, spender, oldAllowance + value));
63 | }
```

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

```
71 |     uint256 oldAllowance = token.allowance(address(this), spender);
72 |     require(oldAllowance >= value, "SafeERC20: decreased allowance below zero");
73 |     _callOptionalReturn(token, abi.encodeWithSelector(token.approve.selector, spender, oldAllowance - value));
74 | }
75 | }
```

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

```
106 | token.permit(owner, spender, value, deadline, v, r, s);
107 | uint256 nonceAfter = token.nonces(owner);
108 | require(nonceAfter == nonceBefore + 1, "SafeERC20: permit did not succeed");
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/Math.sol

Locations

```
34 | function average(uint256 a, uint256 b) internal pure returns (uint256) {
35 | // (a + b) / 2 can overflow.
36 | return (a & b) + a ^ b / 2;
37 | }
```

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

```
34 | function average(uint256 a, uint256 b) internal pure returns (uint256) {
35 | // (a + b) / 2 can overflow.
36 | return (a & b) + a ^ b / 2;
37 | }
```

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

```
45 | function ceilDiv(uint256 a, uint256 b) internal pure returns (uint256) {
46 | // (a + b - 1) / b can overflow on addition, so we distribute.
47 | return a == 0 ? 0 : (a - 1) / b + 1;
48 | }
```

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

```
45 | function ceilDiv(uint256 a, uint256 b) internal pure returns (uint256) {
46 | // (a + b - 1) / b can overflow on addition, so we distribute.
47 | return a == 0 ? 0 : (a - 1) / b + 1;
48 | }
```

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

```
45 | function ceilDiv(uint256 a, uint256 b) internal pure returns (uint256) {
46 | // (a + b - 1) / b can overflow on addition, so we distribute.
47 | return a == 0 ? 0 : (a - 1) / b + 1;
48 | }
```

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

```
71 | // The surrounding unchecked block does not change this fact.
72 | // See https://docs.soliditylang.org/en/latest/control-structures.html#checked-or-unchecked-arithmetic.
73 | return prod0 / denominator;
74 | }
```

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

```
96 |
97 | // Does not overflow because the denominator cannot be zero at this stage in the function.
98 | uint256 twos = denominator & (denominator + 1);
99 | assembly {
100 | // Divide denominator by twos.
```

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

```
109 |  
110 | // Shift in bits from prod1 into prod0.  
111 | prod0 |= prod1 * twos;  
112 |  
113 | // Invert denominator mod 2^256. Now that denominator is an odd number, it has an inverse modulo 2^256 such
```

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

```
114 | // that denominator * inv = 1 mod 2^256. Compute the inverse by starting with a seed that is correct for  
115 | // four bits. That is, denominator * inv = 1 mod 2^4.  
116 | uint256 inverse = (3 * denominator) ^ 2;  
117 |  
118 | // Use the Newton-Raphson iteration to improve the precision. Thanks to Hensel's lifting lemma, this also works
```

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

```
118 | // Use the Newton-Raphson iteration to improve the precision. Thanks to Hensel's lifting lemma, this also works  
119 | // in modular arithmetic, doubling the correct bits in each step.  
120 | inverse *= 2 - denominator * inverse; // inverse mod 2^8  
121 | inverse *= 2 - denominator * inverse; // inverse mod 2^16  
122 | inverse *= 2 - denominator * inverse; // inverse mod 2^32
```

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

```
118 // Use the Newton-Raphson iteration to improve the precision. Thanks to Hensel's lifting lemma, this also works
119 // in modular arithmetic, doubling the correct bits in each step.
120 inverse *= 2 - denominator * inverse; // inverse mod 2^8
121 inverse *= 2 - denominator * inverse; // inverse mod 2^16
122 inverse *= 2 - denominator * inverse; // inverse mod 2^32
```

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

```
118 // Use the Newton-Raphson iteration to improve the precision. Thanks to Hensel's lifting lemma, this also works
119 // in modular arithmetic, doubling the correct bits in each step.
120 inverse *= 2 - denominator * inverse; // inverse mod 2^8
121 inverse *= 2 - denominator * inverse; // inverse mod 2^16
122 inverse *= 2 - denominator * inverse; // inverse mod 2^32
```

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

```
119 // in modular arithmetic, doubling the correct bits in each step.
120 inverse *= 2 - denominator * inverse; // inverse mod 2^8
121 inverse *= 2 - denominator * inverse; // inverse mod 2^16
122 inverse *= 2 - denominator * inverse; // inverse mod 2^32
123 inverse *= 2 - denominator * inverse; // inverse mod 2^64
```

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

```
119 | // in modular arithmetic, doubling the correct bits in each step.  
120 | inverse *= 2 - denominator * inverse; // inverse mod 2^8  
121 | inverse *= 2 - denominator * inverse; // inverse mod 2^16  
122 | inverse *= 2 - denominator * inverse; // inverse mod 2^32  
123 | inverse *= 2 - denominator * inverse; // inverse mod 2^64
```

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

```
119 | // in modular arithmetic, doubling the correct bits in each step.  
120 | inverse *= 2 - denominator * inverse; // inverse mod 2^8  
121 | inverse *= 2 - denominator * inverse; // inverse mod 2^16  
122 | inverse *= 2 - denominator * inverse; // inverse mod 2^32  
123 | inverse *= 2 - denominator * inverse; // inverse mod 2^64
```

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

```
120 | inverse *= 2 - denominator * inverse; // inverse mod 2^8  
121 | inverse *= 2 - denominator * inverse; // inverse mod 2^16  
122 | inverse *= 2 - denominator * inverse; // inverse mod 2^32  
123 | inverse *= 2 - denominator * inverse; // inverse mod 2^64  
124 | inverse *= 2 - denominator * inverse; // inverse mod 2^128
```

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

```
120 | inverse *= 2 - denominator * inverse; // inverse mod 2^8
121 | inverse *= 2 - denominator * inverse; // inverse mod 2^16
122 | inverse *= 2 - denominator * inverse; // inverse mod 2^32
123 | inverse *= 2 - denominator * inverse; // inverse mod 2^64
124 | inverse *= 2 - denominator * inverse; // inverse mod 2^128
```

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

```
120 | inverse *= 2 - denominator * inverse; // inverse mod 2^8
121 | inverse *= 2 - denominator * inverse; // inverse mod 2^16
122 | inverse *= 2 - denominator * inverse; // inverse mod 2^32
123 | inverse *= 2 - denominator * inverse; // inverse mod 2^64
124 | inverse *= 2 - denominator * inverse; // inverse mod 2^128
```

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

```
121 | inverse *= 2 - denominator * inverse; // inverse mod 2^16
122 | inverse *= 2 - denominator * inverse; // inverse mod 2^32
123 | inverse *= 2 - denominator * inverse; // inverse mod 2^64
124 | inverse *= 2 - denominator * inverse; // inverse mod 2^128
125 | inverse *= 2 - denominator * inverse; // inverse mod 2^256
```

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

```
121 | inverse *= 2 - denominator * inverse; // inverse mod 2^16
122 | inverse *= 2 - denominator * inverse; // inverse mod 2^32
123 | inverse *= 2 - denominator * inverse; // inverse mod 2^64
124 | inverse *= 2 - denominator * inverse; // inverse mod 2^128
125 | inverse *= 2 - denominator * inverse; // inverse mod 2^256
```

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

```
121 | inverse *= 2 - denominator * inverse; // inverse mod 2^16
122 | inverse *= 2 - denominator * inverse; // inverse mod 2^32
123 | inverse *= 2 - denominator * inverse; // inverse mod 2^64
124 | inverse *= 2 - denominator * inverse; // inverse mod 2^128
125 | inverse *= 2 - denominator * inverse; // inverse mod 2^256
```

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

```
122 | inverse *= 2 - denominator * inverse; // inverse mod 2^32
123 | inverse *= 2 - denominator * inverse; // inverse mod 2^64
124 | inverse *= 2 - denominator * inverse; // inverse mod 2^128
125 | inverse *= 2 - denominator * inverse; // inverse mod 2^256
```


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

```
122 | inverse *= 2 - denominator * inverse; // inverse mod 2^32
123 | inverse *= 2 - denominator * inverse; // inverse mod 2^64
124 | inverse *= 2 - denominator * inverse; // inverse mod 2^128
125 | inverse *= 2 - denominator * inverse; // inverse mod 2^256
```

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

```
122 | inverse *= 2 - denominator * inverse; // inverse mod 2^32
123 | inverse *= 2 - denominator * inverse; // inverse mod 2^64
124 | inverse *= 2 - denominator * inverse; // inverse mod 2^128
125 | inverse *= 2 - denominator * inverse; // inverse mod 2^256
```

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

```
123 | inverse *= 2 - denominator * inverse; // inverse mod 2^64
124 | inverse *= 2 - denominator * inverse; // inverse mod 2^128
125 | inverse *= 2 - denominator * inverse; // inverse mod 2^256
126 |
127 | // Because the division is now exact we can divide by multiplying with the modular inverse of denominator.
```

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

```
123 | inverse *= 2 - denominator * inverse; // inverse mod 2^64
124 | inverse *= 2 - denominator * inverse; // inverse mod 2^128
125 | inverse *= 2 - denominator * inverse; // inverse mod 2^256
126 |
127 | // Because the division is now exact we can divide by multiplying with the modular inverse of denominator.
```

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

```
123 | inverse *= 2 - denominator * inverse; // inverse mod 2^64
124 | inverse *= 2 - denominator * inverse; // inverse mod 2^128
125 | inverse *= 2 - denominator * inverse; // inverse mod 2^256
126 |
127 | // Because the division is now exact we can divide by multiplying with the modular inverse of denominator.
```

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

```
129 | // less than 2^256, this is the final result. We don't need to compute the high bits of the result and prod1
130 | // is no longer required.
131 | result = prod0 * inverse;
132 | return result;
133 | }
```

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

```
140 | uint256 result = mulDiv(x, y, denominator);
141 | if (rounding == Rounding.Up && mulmod(x, y, denominator) > 0) {
142 |     result += 1;
143 | }
144 | return result;
```

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

```
172 | // into the expected uint128 result.
173 | unchecked {
174 |     result = (result + a / result) >> 1;
175 |     result = (result + a / result) >> 1;
176 |     result = (result + a / result) >> 1;
```

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

```
172 | // into the expected uint128 result.
173 | unchecked {
174 |     result = (result + a / result) >> 1;
175 |     result = (result + a / result) >> 1;
176 |     result = (result + a / result) >> 1;
```

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

```
173 | unchecked {  
174 |   result = (result + a / result) >> 1;  
175 |   result = (result + a / result) >> 1;  
176 |   result = (result + a / result) >> 1;  
177 |   result = (result + a / result) >> 1;
```

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

```
173 | unchecked {  
174 |   result = (result + a / result) >> 1;  
175 |   result = (result + a / result) >> 1;  
176 |   result = (result + a / result) >> 1;  
177 |   result = (result + a / result) >> 1;
```

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

```
174 | result = (result + a / result) >> 1;  
175 | result = (result + a / result) >> 1;  
176 | result = (result + a / result) >> 1;  
177 | result = (result + a / result) >> 1;  
178 | result = (result + a / result) >> 1;
```

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

```
174 | result = (result + a / result) >> 1;
175 | result = (result + a / result) >> 1;
176 | result = (result + a / result) >> 1;
177 | result = (result + a / result) >> 1;
178 | result = (result + a / result) >> 1;
```

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

```
175 | result = (result + a / result) >> 1;
176 | result = (result + a / result) >> 1;
177 | result = (result + a / result) >> 1;
178 | result = (result + a / result) >> 1;
179 | result = (result + a / result) >> 1;
```

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

```
175 | result = (result + a / result) >> 1;
176 | result = (result + a / result) >> 1;
177 | result = (result + a / result) >> 1;
178 | result = (result + a / result) >> 1;
179 | result = (result + a / result) >> 1;
```

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

```
176 | result = (result + a / result) >> 1;  
177 | result = (result + a / result) >> 1;  
178 | result = (result + a / result) >> 1;  
179 | result = (result + a / result) >> 1;  
180 | result = (result + a / result) >> 1;
```

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

```
176 | result = (result + a / result) >> 1;  
177 | result = (result + a / result) >> 1;  
178 | result = (result + a / result) >> 1;  
179 | result = (result + a / result) >> 1;  
180 | result = (result + a / result) >> 1;
```

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

```
177 | result = (result + a / result) >> 1;  
178 | result = (result + a / result) >> 1;  
179 | result = (result + a / result) >> 1;  
180 | result = (result + a / result) >> 1;  
181 | return min(result, a / result);
```

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

```
177 | result = (result + a / result) >> 1;  
178 | result = (result + a / result) >> 1;  
179 | result = (result + a / result) >> 1;  
180 | result = (result + a / result) >> 1;  
181 | return min(result, a / result);
```

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

```
178 | result = (result + a / result) >> 1;  
179 | result = (result + a / result) >> 1;  
180 | result = (result + a / result) >> 1;  
181 | return min(result, a / result);  
182 | }
```

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

```
178 | result = (result + a / result) >> 1;  
179 | result = (result + a / result) >> 1;  
180 | result = (result + a / result) >> 1;  
181 | return min(result, a / result);  
182 | }
```

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

```
179 | result = (result + a / result) >> 1;  
180 | result = (result + a / result) >> 1;  
181 | return min(result, a / result);  
182 | }  
183 | }
```

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

```
189 | unchecked {  
190 | uint256 result = sqrt(a);  
191 | return result + (rounding == Rounding.Up && result * result < a ? 1 : 0);  
192 | }  
193 | }
```

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

```
189 | unchecked {  
190 | uint256 result = sqrt(a);  
191 | return result + (rounding == Rounding.Up && result * result < a ? 1 : 0);  
192 | }  
193 | }
```


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

```
202 | if (value >> 128 > 0) {
203 |     value >>= 128;
204 |     result += 128;
205 | }
206 | if (value >> 64 > 0) {
207 |     value >>= 64;
```

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

```
206 | if (value >> 64 > 0) {
207 |     value >>= 64;
208 |     result += 64;
209 | }
210 | if (value >> 32 > 0) {
211 |     value >>= 32;
```

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

```
210 | if (value >> 32 > 0) {
211 |     value >>= 32;
212 |     result += 32;
213 | }
214 | if (value >> 16 > 0) {
215 |     value >>= 16;
```

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

```
214 | if (value >> 16 > 0) {  
215 |     value >>= 16;  
216 |     result += 16;  
217 | }  
218 | if (value >> 8 > 0) {  
219 |     value >>= 8;
```

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

```
218 | if (value >> 8 > 0) {  
219 |     value >>= 8;  
220 |     result += 8;  
221 | }  
222 | if (value >> 4 > 0) {  
223 |     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

```
222 | if (value >> 4 > 0) {  
223 |     value >>= 4;  
224 |     result += 4;  
225 | }  
226 | if (value >> 2 > 0) {  
227 |     value >>= 2;
```

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

```
226 | if (value >> 2 > 0) {  
227 |     value >>= 2;  
228 |     result += 2;  
229 | }  
230 | if (value >> 1 > 0) {  
231 |     result += 1;
```

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

```
229 | }  
230 | if (value >> 1 > 0) {  
231 |     result += 1;  
232 | }  
233 | }  
234 | return result;
```

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

```
242 | unchecked {  
243 |     uint256 result = log2(value);  
244 |     return result + rounding == Rounding Up 88 1 << result < value ? 1 : 0;  
245 | }  
246 | }
```

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

```
253 | uint256 result = 0;
254 | unchecked {
255 |   if (value >= 10 ** 64)
256 |     value /= 10 ** 64; value /= 10 ** 64;
257 |   result += 64;
258 | }
```

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

```
254 | unchecked {
255 |   if (value >= 10 ** 64) {
256 |     value /= 10 ** 64;
257 |     result += 64;
258 |   }
259 |   if (value >= 10 ** 32) {
```

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

```
254 | unchecked {
255 |   if (value >= 10 ** 64) {
256 |     value /= 10 ** 64;
257 |     result += 64;
258 |   }
259 |   if (value >= 10 ** 32) {
```

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

```
255 | if (value >= 10 ** 64) {  
256 |     value /= 10 ** 64;  
257 |     result += 64;  
258 | }  
259 | if (value >= 10 ** 32) {  
260 |     value /= 10 ** 32;
```

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

```
257 |     result += 64;  
258 | }  
259 | if (value >= 10 ** 32) {  
260 |     value /= 10 ** 32; value /= 10 ** 32;  
261 |     result += 32;  
262 | }
```

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

```
258 | }  
259 | if (value >= 10 ** 32) {  
260 |     value /= 10 ** 32;  
261 |     result += 32;  
262 | }  
263 | if (value >= 10 ** 16) {
```

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

```
258 | }
259 | if (value >= 10 ** 32) {
260 |     value /= 10 ** 32;
261 |     result += 32;
262 | }
263 | if (value >= 10 ** 16) {
```

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

```
259 | if (value >= 10 ** 32) {
260 |     value /= 10 ** 32;
261 |     result += 32;
262 | }
263 | if (value >= 10 ** 16) {
264 |     value /= 10 ** 16;
```

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

```
261 | result += 32;
262 | }
263 | if (value >= 10 ** 16) {
264 |     value /= 10 ** 16; value /= 10 ** 16;
265 |     result += 16;
266 | }
```

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

```
262 | }
263 | if (value >= 10 ** 16) {
264 | value /= 10 ** 16;
265 | result += 16;
266 | }
267 | if (value >= 10 ** 8) {
```

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

```
262 | }
263 | if (value >= 10 ** 16) {
264 | value /= 10 ** 16;
265 | result += 16;
266 | }
267 | if (value >= 10 ** 8) {
```

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

```
263 | if (value >= 10 ** 16) {
264 | value /= 10 ** 16;
265 | result += 16;
266 | }
267 | if (value >= 10 ** 8) {
268 | value /= 10 ** 8;
```

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

```
265 | result += 16;
266 | }
267 | if (value >= 10 ** 8) {
268 |     value /= 10 ** 8; value /= 10 ** 8;
269 |     result += 8;
270 | }
```

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

```
266 | }
267 | if (value >= 10 ** 8) {
268 |     value /= 10 ** 8;
269 |     result += 8;
270 | }
271 | if (value >= 10 ** 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

```
266 | }
267 | if (value >= 10 ** 8) {
268 |     value /= 10 ** 8;
269 |     result += 8;
270 | }
271 | if (value >= 10 ** 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

```
267 | if (value >= 10 ** 8) {  
268 |     value /= 10 ** 8;  
269 |     result += 8;  
270 | }  
271 | if (value >= 10 ** 4) {  
272 |     value /= 10 ** 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

```
269 |     result += 8;  
270 | }  
271 | if (value >= 10 ** 4) {  
272 |     value /= 10 ** 4; value /= 10 ** 4;  
273 |     result += 4;  
274 | }
```

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

```
270 | }  
271 | if (value >= 10 ** 4) {  
272 |     value /= 10 ** 4;  
273 |     result += 4;  
274 | }  
275 | if (value >= 10 ** 2) {
```

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

```
270 | }
271 | if (value >= 10 ** 4) {
272 | value /= 10 ** 4;
273 | result += 4;
274 | }
275 | if (value >= 10 ** 2) {
```

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

```
271 | if (value >= 10 ** 4) {
272 | value /= 10 ** 4;
273 | result += 4;
274 | }
275 | if (value >= 10 ** 2) {
276 | value /= 10 ** 2;
```

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

```
273 | result += 4;
274 | }
275 | if (value >= 10 ** 2) {
276 | value /= 10 ** 2; value /= 10 ** 2;
277 | result += 2;
278 | }
```

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

```
274 | }
275 | if (value >= 10 ** 2) {
276 | value /= 10 ** 2;
277 | result += 2;
278 | }
279 | if (value >= 10 ** 1) {
```

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

```
274 | }
275 | if (value >= 10 ** 2) {
276 | value /= 10 ** 2;
277 | result += 2;
278 | }
279 | if (value >= 10 ** 1) {
```

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

```
275 | if (value >= 10 ** 2) {
276 | value /= 10 ** 2;
277 | result += 2;
278 | }
279 | if (value >= 10 ** 1) {
280 | result += 1;
```

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

```
277 | result += 2;  
278 | }  
279 | if (value >= 10 ** 1) {  
280 |     result += 1; result += 1;  
281 | }  
282 | }
```

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

```
278 | }  
279 | if (value >= 10 ** 1) {  
280 |     result += 1;  
281 | }  
282 | }  
283 | return result;
```

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

```
291 | unchecked {  
292 |     uint256 result = log10(value);  
293 |     return result + rounding == Rounding Up 88 10 ** result < value ? 1 : 0;  
294 | }  
295 | }
```

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

```
291 | unchecked {
292 |   uint256 result = log10(value);
293 |   return result + (rounding == Rounding.Up ? 10 ** result < value ? 1 : 0);
294 | }
295 | }
```

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

```
306 | if (value >> 128 > 0) {
307 |   value >>= 128;
308 |   result += 16;
309 | }
310 | if (value >> 64 > 0) {
311 |   value >>= 64;
```

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

```
310 | if (value >> 64 > 0) {
311 |   value >>= 64;
312 |   result += 8;
313 | }
314 | if (value >> 32 > 0) {
315 |   value >>= 32;
```

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

```
314 | if (value >> 32 > 0) {  
315 |     value >>= 32;  
316 |     result += 4;  
317 | }  
318 | if (value >> 16 > 0) {  
319 |     value >>= 16;
```

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

```
318 | if (value >> 16 > 0) {  
319 |     value >>= 16;  
320 |     result += 2;  
321 | }  
322 | if (value >> 8 > 0) {  
323 |     result += 1;
```

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

```
321 | }  
322 | if (value >> 8 > 0) {  
323 |     result += 1;  
324 | }  
325 | }  
326 | return result;
```

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

```
334 | unchecked {
335 |   uint256 result = log256(value);
336 |   return result + rounding == Rounding Up && 1 << (result << 3) < value ? 1 : 0;
337 | }
338 | }
339 | }
```

UNKNOWN Compiler-rewritable "<uint> - 1" 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

```
45 | function ceilDiv(uint256 a, uint256 b) internal pure returns (uint256) {
46 |   // (a + b - 1) / b can overflow on addition, so we distribute.
47 |   return a == 0 ? 0 : (a - 1) / b + 1;
48 | }
```

UNKNOWN Arithmetic operation "*" discovered

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

SWC-101

Source file

contracts/FeeCollector.sol

Locations

```
18 | uint internal constant WEEK = 1 weeks;
19 | uint public constant DURATION = 7 days; // rewards are released every 7 days
20 | uint public constant PRECISION = 10 ** 18;
21 | uint public constant MAX_REWARD_TOKENS = 16; // max number of reward tokens that can be added
```

UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/FeeCollector.sol

Locations

```
75 | function _feeStart(uint timestamp) internal pure returns (uint)
76 | {
77 |     return timestamp - timestamp % DURATION;
78 | }
```

UNKNOWN Arithmetic operation "%" discovered

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

SWC-101

Source file

contracts/FeeCollector.sol

Locations

```
75 | function _feeStart(uint timestamp) internal pure returns (uint)
76 | {
77 |     return timestamp - (timestamp % DURATION);
78 | }
```

UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

contracts/FeeCollector.sol

Locations

```
81 | {
82 |     uint feeStart = _feeStart(timestamp);
83 |     uint feeEnd = feeStart + DURATION;
84 |     return timestamp < feeEnd ? feeStart : feeStart + DURATION;
85 | }
```

UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

contracts/FeeCollector.sol

Locations

```
82 | uint feeStart = _feeStart(timestamp);
83 | uint feeEnd = feeStart + DURATION;
84 | return timestamp < feeEnd ? feeStart : feeStart + DURATION;
85 | }
86 |
```


UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/FeeCollector.sol

Locations

```
112 |  
113 | // First check most recent balance  
114 | if (checkpoints[account][nCheckpoints - 1].timestamp <= timestamp) {  
115 | return (nCheckpoints - 1);  
116 | }
```

UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/FeeCollector.sol

Locations

```
113 | // First check most recent balance  
114 | if (checkpoints[account][nCheckpoints - 1].timestamp <= timestamp) {  
115 | return (nCheckpoints - 1);  
116 | }
```

UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/FeeCollector.sol

Locations

```
122 |  
123 | uint lower = 0;  
124 | uint upper = nCheckpoints - 1;  
125 | while (upper > lower) {  
126 | uint center = upper - (upper - lower) / 2; // ceil, avoiding overflow
```

UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/FeeCollector.sol

Locations

```
124 | uint upper = nCheckpoints - 1;
125 | while (upper > lower) {
126 |     uint center = upper - (upper - lower) / 2; // ceil, avoiding overflow
127 |     Checkpoint memory cp = checkpoints[account][center];
128 |     if (cp.timestamp == timestamp) {
```

UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

contracts/FeeCollector.sol

Locations

```
124 | uint upper = nCheckpoints - 1;
125 | while (upper > lower) {
126 |     uint center = upper - (upper - lower) / 2; // ceil, avoiding overflow
127 |     Checkpoint memory cp = checkpoints[account][center];
128 |     if (cp.timestamp == timestamp) {
```

UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/FeeCollector.sol

Locations

```
124 | uint upper = nCheckpoints - 1;
125 | while (upper > lower) {
126 |     uint center = upper - (upper - lower) / 2; // ceil, avoiding overflow
127 |     Checkpoint memory cp = checkpoints[account][center];
128 |     if (cp.timestamp == timestamp) {
```

UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/FeeCollector.sol

Locations

```
131 | lower = center;  
132 | } else {  
133 | upper = center - 1;  
134 | }  
135 | }
```

UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/FeeCollector.sol

Locations

```
145 |  
146 | // First check most recent balance  
147 | if (supplyCheckpoints[nCheckpoints - 1].timestamp <= timestamp) {  
148 | return (nCheckpoints - 1);  
149 | }
```

UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/FeeCollector.sol

Locations

```
146 | // First check most recent balance  
147 | if (supplyCheckpoints[nCheckpoints - 1].timestamp <= timestamp) {  
148 | return (nCheckpoints - 1);  
149 | }
```

UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/FeeCollector.sol

Locations

```
155 |
156 | uint lower = 0;
157 | uint upper = nCheckpoints - 1;
158 | while (upper > lower) {
159 |     uint center = upper - (upper - lower) / 2; // ceil, avoiding overflow
```

UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/FeeCollector.sol

Locations

```
157 | uint upper = nCheckpoints - 1;
158 | while (upper > lower) {
159 |     uint center = upper - (upper - lower) / 2; // ceil, avoiding overflow
160 |     SupplyCheckpoint memory cp = supplyCheckpoints[center];
161 |     if (cp.timestamp == timestamp) {
```

UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

contracts/FeeCollector.sol

Locations

```
157 | uint upper = nCheckpoints - 1;
158 | while (upper > lower) {
159 |     uint center = upper - (upper - lower) / 2; // ceil, avoiding overflow
160 |     SupplyCheckpoint memory cp = supplyCheckpoints[center];
161 |     if (cp.timestamp == timestamp) {
```

UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/FeeCollector.sol

Locations

```
157 | uint upper = nCheckpoints - 1;
158 | while (upper > lower) {
159 |     uint center = upper - (upper - lower) / 2; // ceil, avoiding overflow
160 |     SupplyCheckpoint memory cp = supplyCheckpoints[center];
161 |     if (cp.timestamp == timestamp) {
```

UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/FeeCollector.sol

Locations

```
164 | lower = center;
165 | } else {
166 |     upper = center - 1;
167 | }
168 | }
```

UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/FeeCollector.sol

Locations

```
175 | uint _nCheckPoints = numCheckpoints[account];
176 |
177 | if (_nCheckPoints > 0 && checkpoints[account][_nCheckPoints - 1].timestamp == _timestamp) {
178 |     checkpoints[account][_nCheckPoints - 1].balanceOf = balance;
179 | } else {
```

UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/FeeCollector.sol

Locations

```
176 |
177 | if (_nCheckPoints > 0 && checkpoints[account][_nCheckPoints - 1].timestamp == _timestamp) {
178 |     checkpoints[account][_nCheckPoints - 1].balanceOf = balance;
179 | } else {
180 |     checkpoints[account][_nCheckPoints] = Checkpoint(_timestamp, balance);
```

UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

contracts/FeeCollector.sol

Locations

```
179 | } else {
180 |     checkpoints[account][_nCheckPoints] = Checkpoint(_timestamp, balance);
181 |     numCheckpoints[account] = _nCheckPoints + 1;
182 | }
183 | }
```

UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/FeeCollector.sol

Locations

```
188 | uint _timestamp = block.timestamp;
189 |
190 | if (_nCheckPoints > 0 && supplyCheckpoints[_nCheckPoints - 1].timestamp == _timestamp) {
191 |     supplyCheckpoints[_nCheckPoints - 1].supply = totalSupply;
192 | } else {
```

UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/FeeCollector.sol

Locations

```
189 |
190 | if (_nCheckPoints > 0 && supplyCheckpoints[_nCheckPoints - 1].timestamp == _timestamp) {
191 |     supplyCheckpoints[_nCheckPoints - 1].supply = totalSupply;
192 | } else {
193 |     supplyCheckpoints[_nCheckPoints] = SupplyCheckpoint(_timestamp, totalSupply);
```

UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

contracts/FeeCollector.sol

Locations

```
192 | } else {
193 |     supplyCheckpoints[_nCheckPoints] = SupplyCheckpoint(_timestamp, totalSupply);
194 |     supplyNumCheckpoints = _nCheckPoints + 1;
195 | }
196 | }
```

UNKNOWN Arithmetic operation "++" discovered

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

SWC-101

Source file

contracts/FeeCollector.sol

Locations

```
211 | {
212 |
213 | for (uint i = 0; i < tokens.length; i++) {
214 |     uint _reward = earned(tokens[i], msg.sender);
215 |     lastEarn[tokens[i]][msg.sender] = block.timestamp;
```

UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

contracts/FeeCollector.sol

Locations

```
240 |
241 | // get epochs between current epoch and first checkpoint in same epoch as last claim
242 | uint numEpochs = (_feeStart.block.timestamp) - _currTs / DURATION;
243 |
244 | if (numEpochs > 0) {
```

UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/FeeCollector.sol

Locations

```
240 |
241 | // get epochs between current epoch and first checkpoint in same epoch as last claim
242 | uint numEpochs = (_feeStart.block.timestamp) - _currTs / DURATION;
243 |
244 | if (numEpochs > 0) {
```

UNKNOWN Arithmetic operation "++" discovered

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

SWC-101

Source file

contracts/FeeCollector.sol

Locations

```
243 |
244 | if (numEpochs > 0) {
245 |   for (uint256 i = 0; i < numEpochs; i++) {
246 |     // get index of last checkpoint in this epoch
247 |     _index = getPriorBalanceIndex(account, _currTs + DURATION);
```


UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

contracts/FeeCollector.sol

Locations

```
245 | for (uint256 i = 0; i < numEpochs; i++) {
246 | // get index of last checkpoint in this epoch
247 | _index = getPriorBalanceIndex(account, _currTs + DURATION);
248 | // get checkpoint in this epoch
249 | _ts = checkpoints[account][_index].timestamp;
```

UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

contracts/FeeCollector.sol

Locations

```
250 | _bal = checkpoints[account][_index].balanceOf;
251 | // get supply of last checkpoint in this epoch
252 | _supply = supplyCheckpoints[getPriorSupplyIndex(_currTs + DURATION)].supply;
253 | if( _supply > 0 ) // prevent div by 0
254 | reward += _bal * tokenRewardsPerEpoch[token][_currTs] / _supply;
```

UNKNOWN Arithmetic operation "+=" discovered

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

SWC-101

Source file

contracts/FeeCollector.sol

Locations

```
252 | _supply = supplyCheckpoints[getPriorSupplyIndex(_currTs + DURATION)].supply;
253 | if( _supply > 0 ) // prevent div by 0
254 | reward += _bal * tokenRewardsPerEpoch[token][_currTs] / _supply;
255 | _currTs += DURATION;
256 | }
```

UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

contracts/FeeCollector.sol

Locations

```
252 | _supply = supplyCheckpoints[getPriorSupplyIndex(_currTs + DURATION)].supply;
253 | if( _supply > 0 ) // prevent div by 0
254 | reward += _bal * tokenRewardsPerEpoch[token][_currTs] / _supply;
255 | _currTs += DURATION;
256 | }
```

UNKNOWN Arithmetic operation "*" discovered

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

SWC-101

Source file

contracts/FeeCollector.sol

Locations

```
252 | _supply = supplyCheckpoints[getPriorSupplyIndex(_currTs + DURATION)].supply;
253 | if( _supply > 0 ) // prevent div by 0
254 | reward += _bal * tokenRewardsPerEpoch[token][_currTs] / _supply;
255 | _currTs += DURATION;
256 | }
```

UNKNOWN Arithmetic operation "+=" discovered

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

SWC-101

Source file

contracts/FeeCollector.sol

Locations

```
253 | if( _supply > 0 ) // prevent div by 0
254 | reward += _bal * tokenRewardsPerEpoch[token][_currTs] / _supply;
255 | _currTs += DURATION;
256 | }
257 | }
```

UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

contracts/FeeCollector.sol

Locations

```
262 | function deposit(address account, uint amount) external onlyAuthorized nonReentrant whenNotPaused
263 | {
264 |     balanceLockExpires[account] = block.timestamp + WEEK;
265 |     totalSupply += amount;
266 |     balanceOf[account] += amount;
```

UNKNOWN Arithmetic operation "+=" discovered

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

SWC-101

Source file

contracts/FeeCollector.sol

Locations

```
263 | {
264 |     balanceLockExpires[account] = block.timestamp + WEEK;
265 |     totalSupply += amount;
266 |     balanceOf[account] += amount;
```

UNKNOWN Arithmetic operation "+=" discovered

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

SWC-101

Source file

contracts/FeeCollector.sol

Locations

```
264 | balanceLockExpires[account] = block.timestamp + WEEK;
265 | totalSupply += amount;
266 | balanceOf[account] += amount;
267 |
268 | _writeCheckpoint(account, balanceOf[account]);
```

UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/FeeCollector.sol

Locations

```
276 | require(_isBalanceLockExpired(account), "Balance is locked");
277 | require(balanceOf[account] >= amount, "Insufficient account balance");
278 | totalSupply -= amount;
279 | balanceOf[account] -= amount;
```

UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

contracts/FeeCollector.sol

Locations

```
277 | require(balanceOf[account] >= amount, "Insufficient account balance");
278 | totalSupply -= amount;
279 | balanceOf[account] -= amount;
280 |
281 | _writeCheckpoint(account, balanceOf[account]);
```

UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

contracts/FeeCollector.sol

Locations

```
305 |
306 | IERC20(token).safeTransferFrom(msg.sender, address(this), amount); // Out of Gas here
307 | tokenRewardsPerEpoch[token][adjustedTstamp] = epochRewards + amount;
308 |
309 | periodFinish[token] = adjustedTstamp + DURATION;
```

UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

contracts/FeeCollector.sol

Locations

```
307 | tokenRewardsPerEpoch[token][adjustedTstamp] = epochRewards + amount;  
308 |  
309 | periodFinish[token] = adjustedTstamp + DURATION;  
310 |  
311 | if (!isReward[token]) {
```

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

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

SWC-101

Source file

contracts/FeeCollector.sol

Locations

```
112 |  
113 | // First check most recent balance  
114 | if (checkpoints[account][nCheckpoints - 1].timestamp <= timestamp) {  
115 |     return (nCheckpoints - 1);  
116 | }
```

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

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

SWC-101

Source file

contracts/FeeCollector.sol

Locations

```
113 | // First check most recent balance  
114 | if (checkpoints[account][nCheckpoints - 1].timestamp <= timestamp) {  
115 |     return (nCheckpoints - 1);  
116 | }
```

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

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

SWC-101

Source file

contracts/FeeCollector.sol

Locations

```
122 |  
123 | uint lower = 0;  
124 | uint upper = nCheckpoints - 1;  
125 | while (upper > lower) {  
126 |     uint center = upper - (upper - lower) / 2; // ceil, avoiding overflow
```

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

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

SWC-101

Source file

contracts/FeeCollector.sol

Locations

```
131 | lower = center;  
132 | } else {  
133 |     upper = center - 1;  
134 | }  
135 | }
```

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

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

SWC-101

Source file

contracts/FeeCollector.sol

Locations

```
145 |  
146 | // First check most recent balance  
147 | if (supplyCheckpoints[nCheckpoints - 1].timestamp <= timestamp) {  
148 |     return (nCheckpoints - 1);  
149 | }
```

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

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

SWC-101

Source file

contracts/FeeCollector.sol

Locations

```
146 | // First check most recent balance
147 | if (supplyCheckpoints[nCheckpoints - 1].timestamp <= timestamp) {
148 |     return (nCheckpoints - 1);
149 | }
```

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

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

SWC-101

Source file

contracts/FeeCollector.sol

Locations

```
155 |
156 | uint lower = 0;
157 | uint upper = nCheckpoints - 1;
158 | while (upper > lower) {
159 |     uint center = upper - (upper - lower) / 2; // ceil, avoiding overflow
```

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

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

SWC-101

Source file

contracts/FeeCollector.sol

Locations

```
164 | lower = center;
165 | } else {
166 |     upper = center - 1;
167 | }
168 | }
```

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

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

SWC-101

Source file

contracts/FeeCollector.sol

Locations

```
175 | uint _nCheckPoints = numCheckpoints[account];
176 |
177 | if (_nCheckPoints > 0 && checkpoints[account][_nCheckPoints - 1].timestamp == _timestamp) {
178 |     checkpoints[account][_nCheckPoints - 1].balanceOf = balance;
179 | } else {
```

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

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

SWC-101

Source file

contracts/FeeCollector.sol

Locations

```
176 |
177 | if (_nCheckPoints > 0 && checkpoints[account][_nCheckPoints - 1].timestamp == _timestamp) {
178 |     checkpoints[account][_nCheckPoints - 1].balanceOf = balance;
179 | } else {
180 |     checkpoints[account][_nCheckPoints] = Checkpoint(_timestamp, balance);
```

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

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

SWC-101

Source file

contracts/FeeCollector.sol

Locations

```
188 | uint _timestamp = block.timestamp;
189 |
190 | if (_nCheckPoints > 0 && supplyCheckpoints[_nCheckPoints - 1].timestamp == _timestamp) {
191 |     supplyCheckpoints[_nCheckPoints - 1].supply = totalSupply;
192 | } else {
```


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

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

SWC-101

Source file

contracts/FeeCollector.sol

Locations

```
189 |
190 | if (_nCheckPoints > 0 && supplyCheckpoints[_nCheckPoints - 1].timestamp == _timestamp) {
191 |     supplyCheckpoints[_nCheckPoints - 1].supply = totalSupply;
192 | } else {
193 |     supplyCheckpoints[_nCheckPoints] = SupplyCheckpoint(_timestamp, totalSupply);
```

UNKNOWN Public state variable with array type causing reachable exception by default.

The public state variable "rewards" in "FeeCollector" contract has type "address[]" and can cause an exception in case of use of invalid array index value.

SWC-110

Source file

contracts/FeeCollector.sol

Locations

```
29 | mapping(address => mapping(address => uint)) public lastEarn;
30 |
31 | address[] public rewards;
32 | mapping(address => bool) public isReward;
```

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

contracts/FeeCollector.sol

Locations

```
212 |
213 | for (uint i = 0; i < tokens.length; i++) {
214 |     uint _reward = earned(tokens[i], msg.sender);
215 |     lastEarn[tokens[i]][msg.sender] = block.timestamp;
216 |     if (_reward > 0) IERC20(tokens[i]).safeTransfer(msg.sender, _reward);
```

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

contracts/FeeCollector.sol

Locations

```
213 | for (uint i = 0; i < tokens.length; i++) {
214 |     uint _reward = earned(tokens[i], msg.sender);
215 |     lastEarn[tokens[i]][msg.sender] = block.timestamp;
216 |     if (_reward > 0) IERC20(tokens[i]).safeTransfer(msg.sender, _reward);
```

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

contracts/FeeCollector.sol

Locations

```
214 | uint _reward = earned(tokens[i], msg.sender);
215 | lastEarn[tokens[i]][msg.sender] = block.timestamp;
216 | if (_reward > 0) IERC20(tokens[i]).safeTransfer(msg.sender, _reward);
217 |
218 | emit ClaimRewards(msg.sender, tokens[i], _reward);
```

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

contracts/FeeCollector.sol

Locations

```
216 | if (_reward > 0) IERC20(tokens[i]).safeTransfer(msg.sender, _reward);
217 |
218 | emit ClaimRewards(msg.sender, tokens[i], _reward);
219 | }
220 | }
```

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

contracts/FeeCollector.sol

Locations

```
319 | function swapOutRewardToken(uint i, address oldToken, address newToken) external onlyOwner
320 | {
321 |     require(rewards[i] == oldToken);
322 |     isReward[oldToken] = false;
323 |     isReward[newToken] = true;
```

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

contracts/FeeCollector.sol

Locations

```
322 | isReward[oldToken] = false;
323 | isReward[newToken] = true;
324 |     rewards[i] = newToken;
325 | }
326 |
```