

REPORT 628D6808666C2300183596D9

| | |
|--------------------|--|
| Created | Tue May 24 2022 23:19:36 GMT+0000 (Coordinated Universal Time) |
| Number of analyses | 1 |
| User | 6197960e3494e9c8c076e89b |

REPORT SUMMARY

| Analyses ID | Main source file | Detected vulnerabilities |
|--|------------------------|--------------------------|
| 66757573-d200-4fcc-bb5b-6cdbe21cbea7 | RewardsDistributor.sol | 0 |

| | |
|------------------|--|
| Started | Tue May 24 2022 23:19:40 GMT+0000 (Coordinated Universal Time) |
| Finished | Tue May 24 2022 23:21:50 GMT+0000 (Coordinated Universal Time) |
| Mode | Deep |
| Client Tool | Remythx |
| Main Source File | RewardsDistributor.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

RewardsDistributor.sol

Locations

```
28 | );  
29 |  
30 | uint constant WEEK = 7 * 86400;  
31 |  
32 | uint public start_time;
```

UNKNOWN Arithmetic operation "*" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
47 |  
48 | constructor(address _voting_escrow) {  
49 |     uint _t = block.timestamp / WEEK * WEEK;  
50 |     start_time = _t;  
51 |     last_token_time = _t;
```

UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
47 |  
48 | constructor(address _voting_escrow) {  
49 |     uint _t = block.timestamp / WEEK * WEEK;  
50 |     start_time = _t;  
51 |     last_token_time = _t;
```

UNKNOWN Arithmetic operation "*" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
59 |  
60 | function timestamp() external view returns (uint) {  
61 |     return block.timestamp / WEEK * WEEK;  
62 | }  
63 |
```

UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
59 |  
60 | function timestamp() external view returns (uint) {  
61 |     return block.timestamp / WEEK * WEEK;  
62 | }  
63 |
```

UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
64 | function _checkpoint_token() internal {
65 |     uint token_balance = IERC20(token).balanceOf(address(this));
66 |     uint to_distribute = token_balance - token_last_balance;
67 |     token_last_balance = token_balance;
68 | }
```

UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
68 |
69 | uint t = last_token_time;
70 | uint since_last = block.timestamp - t;
71 | last_token_time = block.timestamp;
72 | uint this_week = t / WEEK * WEEK;
```

UNKNOWN Arithmetic operation "*" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
70 | uint since_last = block.timestamp - t;
71 | last_token_time = block.timestamp;
72 | uint this_week = t / WEEK * WEEK;
73 | uint next_week = 0;
74 |
```

UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
70 | uint since_last = block.timestamp - t;  
71 | last_token_time = block.timestamp;  
72 | uint this_week = t / WEEK * WEEK;  
73 | uint next_week = 0;  
74 |
```

UNKNOWN Arithmetic operation "++" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
73 | uint next_week = 0;  
74 |  
75 | for (uint i = 0; i < 20; i++) {  
76 |     next_week = this_week + WEEK;  
77 |     if (block.timestamp < next_week) {
```

UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
74 |  
75 | for (uint i = 0; i < 20; i++) {  
76 |     next_week = this_week + WEEK;  
77 |     if (block.timestamp < next_week) {  
78 |         if (since_last == 0 && block.timestamp == t) {
```

UNKNOWN Arithmetic operation "+=" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
77 | if (block.timestamp < next_week) {
78 |   if (since_last == 0 && block.timestamp == t) {
79 |     tokens_per_week[this_week] += to_distribute;
80 |   } else {
81 |     tokens_per_week[this_week] += to_distribute * (block.timestamp - t) / since_last;
```

UNKNOWN Arithmetic operation "+=" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
79 | tokens_per_week[this_week] += to_distribute;
80 | } else {
81 |   tokens_per_week[this_week] += to_distribute * (block.timestamp - t) / since_last;
82 | }
83 | break;
```

UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
79 | tokens_per_week[this_week] += to_distribute;
80 | } else {
81 |   tokens_per_week[this_week] += to_distribute * (block.timestamp - t) / since_last;
82 | }
83 | break;
```

UNKNOWN Arithmetic operation "*" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
79 | tokens_per_week[this_week] += to_distribute;  
80 | } else {  
81 | tokens_per_week[this_week] += to_distribute * (block.timestamp - t) / since_last;  
82 | }  
83 | break;
```

UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
79 | tokens_per_week[this_week] += to_distribute;  
80 | } else {  
81 | tokens_per_week[this_week] += to_distribute * (block.timestamp - t) / since_last;  
82 | }  
83 | break;
```

UNKNOWN Arithmetic operation "+=" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
84 | } else {  
85 | if (since_last == 0 && next_week == t) {  
86 | tokens_per_week[this_week] += to_distribute;  
87 | } else {  
88 | tokens_per_week[this_week] += to_distribute * (next_week - t) / since_last;
```

UNKNOWN Arithmetic operation "+=" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
86 | tokens_per_week[this_week] += to_distribute;  
87 | } else {  
88 | tokens_per_week[this_week] += to_distribute * (next_week - t) / since_last;  
89 | }  
90 | }
```

UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
86 | tokens_per_week[this_week] += to_distribute;  
87 | } else {  
88 | tokens_per_week[this_week] += to_distribute * (next_week - t) / since_last;  
89 | }  
90 | }
```

UNKNOWN Arithmetic operation "*" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
86 | tokens_per_week[this_week] += to_distribute;  
87 | } else {  
88 | tokens_per_week[this_week] += to_distribute * (next_week - t) / since_last;  
89 | }  
90 | }
```


UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
86 | tokens_per_week[this_week] += to_distribute;  
87 | } else {  
88 | tokens_per_week[this_week] += to_distribute * (next_week - t) / since_last;  
89 | }  
90 | }
```

UNKNOWN Arithmetic operation "++" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
103 | uint _min = 0;  
104 | uint _max = IVotingEscrow(ve).epoch();  
105 | for (uint i = 0; i < 128; i++) {  
106 | if (_min >= _max) break;  
107 | uint _mid = (_min + _max + 2) / 2;
```

UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
105 | for (uint i = 0; i < 128; i++) {  
106 | if (_min >= _max) break;  
107 | uint _mid = (_min + _max + 2) / 2;  
108 | IVotingEscrow.Point memory pt = IVotingEscrow(ve).point_history(_mid);  
109 | if (pt.ts <= _timestamp) {
```

UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
105 | for (uint i = 0; i < 128; i++) {  
106 |     if (_min >= _max) break;  
107 |     uint _mid = (_min + _max + 2) / 2;  
108 |     IVotingEscrow.Point memory pt = IVotingEscrow(ve).point_history(_mid);  
109 |     if (pt.ts <= _timestamp) {
```

UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
105 | for (uint i = 0; i < 128; i++) {  
106 |     if (_min >= _max) break;  
107 |     uint _mid = (_min + _max + 2) / 2;  
108 |     IVotingEscrow.Point memory pt = IVotingEscrow(ve).point_history(_mid);  
109 |     if (pt.ts <= _timestamp) {
```

UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
110 | _min = _mid;  
111 | } else {  
112 |     _max = _mid - 1;  
113 | }  
114 | }
```

UNKNOWN Arithmetic operation "++" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
119 | uint _min = 0;  
120 | uint _max = max_user_epoch;  
121 | for (uint i = 0; i < 128; i++) {  
122 | if (_min >= _max) break;  
123 | uint _mid = (_min + _max + 2) / 2;
```

UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
121 | for (uint i = 0; i < 128; i++) {  
122 | if (_min >= _max) break;  
123 | uint _mid = (_min + _max + 2) / 2;  
124 | IVotingEscrow.Point memory pt = IVotingEscrow(ve).user_point_history(tokenId, _mid);  
125 | if (pt.ts <= _timestamp) {
```

UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
121 | for (uint i = 0; i < 128; i++) {  
122 | if (_min >= _max) break;  
123 | uint _mid = (_min + _max + 2) / 2;  
124 | IVotingEscrow.Point memory pt = IVotingEscrow(ve).user_point_history(tokenId, _mid);  
125 | if (pt.ts <= _timestamp) {
```

UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
121 | for (uint i = 0; i < 128; i++) {  
122 |   if (_min >= _max) break;  
123 |   uint _mid = (_min + _max + 2) / 2;  
124 |   IVotingEscrow.Point memory pt = IVotingEscrow(ve).user_point_history(tokenId, _mid);  
125 |   if (pt.ts <= _timestamp) {
```

UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
126 | _min = _mid;  
127 | } else {  
128 |   _max = _mid - 1;  
129 | }  
130 | }
```

UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
137 | uint epoch = _find_timestamp_user_epoch(ve, _tokenId, _timestamp, max_user_epoch);  
138 | IVotingEscrow.Point memory pt = IVotingEscrow(ve).user_point_history(_tokenId, epoch);  
139 | return Math.max(uint(int256(pt.bias - pt.slope * ((int128(int256(_timestamp) - pt.ts)))), 0);  
140 | }  
141 |
```

UNKNOWN Arithmetic operation "*" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
137 | uint epoch = _find_timestamp_user_epoch(ve, _tokenId, _timestamp, max_user_epoch);
138 | IVotingEscrow.Point memory pt = IVotingEscrow(ve).user_point_history(_tokenId, epoch);
139 | return Math.max(uint(int256(pt.bias - pt.slope * (int128(int256(_timestamp) - pt.ts)))), 0);
140 | }
141 |
```

UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
137 | uint epoch = _find_timestamp_user_epoch(ve, _tokenId, _timestamp, max_user_epoch);
138 | IVotingEscrow.Point memory pt = IVotingEscrow(ve).user_point_history(_tokenId, epoch);
139 | return Math.max(uint(int256(pt.bias - pt.slope * (int128(int256(_timestamp) - pt.ts)))), 0);
140 | }
141 |
```

UNKNOWN Arithmetic operation "*" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
143 | address ve = voting_escrow;
144 | uint t = time_cursor;
145 | uint rounded_timestamp = block.timestamp / WEEK * WEEK;
146 | IVotingEscrow(ve).checkpoint();
147 |
```

UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
143 | address ve = voting_escrow;
144 | uint t = time_cursor;
145 | uint rounded_timestamp = block.timestamp / WEEK * WEEK;
146 | IVotingEscrow(ve).checkpoint();
147 |
```

UNKNOWN Arithmetic operation "++" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
146 | IVotingEscrow(ve).checkpoint();
147 |
148 | for (uint i = 0; i < 20; i++) {
149 |     if (t > rounded_timestamp) {
150 |         break;
151 |     }
152 | }
```

UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
154 | int128 dt = 0;
155 | if (t > pt.ts) {
156 |     dt = int128(int256(t - pt.ts));
157 | }
158 | ve_supply[t] = Math.max(uint(int256(pt.bias - pt.slope * dt)), 0);
```

UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
156 | dt = int128(int256(t - pt.ts));
157 | }
158 | ve_supply[t] = Math.max(uint(int256(pt.bias - pt.slope * dt)), 0);
159 | }
160 | t += WEEK;
```

UNKNOWN Arithmetic operation "*" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
156 | dt = int128(int256(t - pt.ts));
157 | }
158 | ve_supply[t] = Math.max(uint(int256(pt.bias - pt.slope * dt)), 0);
159 | }
160 | t += WEEK;
```

UNKNOWN Arithmetic operation "+=" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
158 | ve_supply[t] = Math.max(uint(int256(pt.bias - pt.slope * dt)), 0);
159 | }
160 | t += WEEK;
161 | }
162 | time_cursor = t;
```

UNKNOWN Arithmetic operation "*" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
187 | IVotingEscrow.Point memory user_point = IVotingEscrow(ve).user_point_history(_tokenId, user_epoch);
188 |
189 | if (week_cursor == 0) week_cursor = user_point.ts + WEEK - 1 / WEEK * WEEK;
190 | if (week_cursor >= last_token_time) return 0;
191 | if (week_cursor < _start_time) week_cursor = _start_time;
```

UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
187 | IVotingEscrow.Point memory user_point = IVotingEscrow(ve).user_point_history(_tokenId, user_epoch);
188 |
189 | if (week_cursor == 0) week_cursor = user_point.ts + WEEK - 1 / WEEK * WEEK;
190 | if (week_cursor >= last_token_time) return 0;
191 | if (week_cursor < _start_time) week_cursor = _start_time;
```

UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
187 | IVotingEscrow.Point memory user_point = IVotingEscrow(ve).user_point_history(_tokenId, user_epoch);
188 |
189 | if (week_cursor == 0) week_cursor = (user_point.ts + WEEK - 1) / WEEK * WEEK;
190 | if (week_cursor >= last_token_time) return 0;
191 | if (week_cursor < _start_time) week_cursor = _start_time;
```


UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
187 | IVotingEscrow.Point memory user_point = IVotingEscrow(ve).user_point_history(_tokenId, user_epoch);
188 |
189 | if (week_cursor == 0) week_cursor = (user_point.ts + WEEK - 1) / WEEK * WEEK;
190 | if (week_cursor >= last_token_time) return 0;
191 | if (week_cursor < _start_time) week_cursor = _start_time;
```

UNKNOWN Arithmetic operation "++" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
193 | IVotingEscrow.Point memory old_user_point;
194 |
195 | for (uint i = 0; i < 50; i++) {
196 |     if (week_cursor >= _last_token_time) break;
197 | }
```

UNKNOWN Arithmetic operation "+=" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
197 |
198 | if (week_cursor >= user_point.ts && user_epoch <= max_user_epoch) {
199 |     user_epoch += 1;
200 |     old_user_point = user_point;
201 |     if (user_epoch > max_user_epoch) {
```

UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
205 | }  
206 | } else {  
207 | int128 dt = int128(int256(week_cursor - old_user_point.ts));  
208 | uint balance_of = Math.max(uint(int256(old_user_point.bias - dt * old_user_point.slope)), 0);  
209 | if (balance_of == 0 && user_epoch > max_user_epoch) break;
```

UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
206 | } else {  
207 | int128 dt = int128(int256(week_cursor - old_user_point.ts));  
208 | uint balance_of = Math.max(uint(int256(old_user_point.bias - dt * old_user_point.slope)), 0);  
209 | if (balance_of == 0 && user_epoch > max_user_epoch) break;  
210 | if (balance_of != 0) {
```

UNKNOWN Arithmetic operation "*" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
206 | } else {  
207 | int128 dt = int128(int256(week_cursor - old_user_point.ts));  
208 | uint balance_of = Math.max(uint(int256(old_user_point.bias - dt * old_user_point.slope)), 0);  
209 | if (balance_of == 0 && user_epoch > max_user_epoch) break;  
210 | if (balance_of != 0) {
```

UNKNOWN Arithmetic operation "+=" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
209 | if (balance_of == 0 && user_epoch > max_user_epoch) break;
210 | if (balance_of != 0) {
211 |   to_distribute += balance_of * tokens_per_week week_cursor / ve_supply week_cursor;
212 | }
213 | week_cursor += WEEK;
```

UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
209 | if (balance_of == 0 && user_epoch > max_user_epoch) break;
210 | if (balance_of != 0) {
211 |   to_distribute += balance_of * tokens_per_week week_cursor / ve_supply week_cursor;
212 | }
213 | week_cursor += WEEK;
```

UNKNOWN Arithmetic operation "*" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
209 | if (balance_of == 0 && user_epoch > max_user_epoch) break;
210 | if (balance_of != 0) {
211 |   to_distribute += balance_of * tokens_per_week week_cursor / ve_supply[week_cursor];
212 | }
213 | week_cursor += WEEK;
```

UNKNOWN Arithmetic operation "+=" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
211 | to_distribute += balance_of * tokens_per_week[week_cursor] / ve_supply[week_cursor];
212 | }
213 | week_cursor += WEEK;
214 | }
215 | }
```

UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
215 | }
216 |
217 | user_epoch = Math.min(max_user_epoch, user_epoch - 1);
218 | user_epoch_of[_tokenId] = user_epoch;
219 | time_cursor_of[_tokenId] = week_cursor;
```

UNKNOWN Arithmetic operation "*" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
244 | IVotingEscrow.Point memory user_point = IVotingEscrow(ve).user_point_history(_tokenId, user_epoch);
245 |
246 | if (week_cursor == 0) week_cursor = (user_point.ts + WEEK - 1) / WEEK * WEEK;
247 | if (week_cursor >= last_token_time) return 0;
248 | if (week_cursor < _start_time) week_cursor = _start_time;
```

UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
244 | IVotingEscrow.Point memory user_point = IVotingEscrow(ve).user_point_history(_tokenId, user_epoch);
245 |
246 | if (week_cursor == 0) week_cursor = (user_point.ts + WEEK - 1) / WEEK * WEEK;
247 | if (week_cursor >= last_token_time) return 0;
248 | if (week_cursor < _start_time) week_cursor = _start_time;
```

UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
244 | IVotingEscrow.Point memory user_point = IVotingEscrow(ve).user_point_history(_tokenId, user_epoch);
245 |
246 | if (week_cursor == 0) week_cursor = (user_point.ts + WEEK - 1) / WEEK * WEEK;
247 | if (week_cursor >= last_token_time) return 0;
248 | if (week_cursor < _start_time) week_cursor = _start_time;
```

UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
244 | IVotingEscrow.Point memory user_point = IVotingEscrow(ve).user_point_history(_tokenId, user_epoch);
245 |
246 | if (week_cursor == 0) week_cursor = (user_point.ts + WEEK - 1) / WEEK * WEEK;
247 | if (week_cursor >= last_token_time) return 0;
248 | if (week_cursor < _start_time) week_cursor = _start_time;
```

UNKNOWN Arithmetic operation "++" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
250 | IVotingEscrow.Point memory old_user_point;
251 |
252 | for (uint i = 0; i < 50; i++) {
253 |     if (week_cursor >= _last_token_time) break;
254 | }
```

UNKNOWN Arithmetic operation "+=" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
254 |
255 | if (week_cursor >= user_point.ts && user_epoch <= max_user_epoch) {
256 |     user_epoch += 1;
257 |     old_user_point = user_point;
258 |     if (user_epoch > max_user_epoch) {
```

UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
262 | }
263 | } else {
264 |     int128 dt = int128(int256(week_cursor - old_user_point.ts));
265 |     uint balance_of = Math.max(uint(int256(old_user_point.bias - dt * old_user_point.slope)), 0);
266 |     if (balance_of == 0 && user_epoch > max_user_epoch) break;
```

UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
263 | } else {  
264 |     int128 dt = int128(int256(week_cursor - old_user_point.ts));  
265 |     uint balance_of = Math.max(uint(int256(old_user_point.bias - dt * old_user_point.slope)), 0);  
266 |     if (balance_of == 0 && user_epoch > max_user_epoch) break;  
267 |     if (balance_of != 0) {
```

UNKNOWN Arithmetic operation "*" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
263 | } else {  
264 |     int128 dt = int128(int256(week_cursor - old_user_point.ts));  
265 |     uint balance_of = Math.max(uint(int256(old_user_point.bias - dt * old_user_point.slope)), 0);  
266 |     if (balance_of == 0 && user_epoch > max_user_epoch) break;  
267 |     if (balance_of != 0) {
```

UNKNOWN Arithmetic operation "+=" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
266 | if (balance_of == 0 && user_epoch > max_user_epoch) break;  
267 | if (balance_of != 0) {  
268 |     to_distribute += balance_of * tokens_per_week week_cursor / ve_supply week_cursor;  
269 | }  
270 | week_cursor += WEEK;
```

UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
266 | if (balance_of == 0 && user_epoch > max_user_epoch) break;
267 | if (balance_of != 0) {
268 |   to_distribute += balance_of * tokens_per_week[week_cursor] / ve_supply[week_cursor];
269 | }
270 | week_cursor += WEEK;
```

UNKNOWN Arithmetic operation "*" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
266 | if (balance_of == 0 && user_epoch > max_user_epoch) break;
267 | if (balance_of != 0) {
268 |   to_distribute += balance_of * tokens_per_week[week_cursor] / ve_supply[week_cursor];
269 | }
270 | week_cursor += WEEK;
```

UNKNOWN Arithmetic operation "+=" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
268 | to_distribute += balance_of * tokens_per_week[week_cursor] / ve_supply[week_cursor];
269 | }
270 | week_cursor += WEEK;
271 | }
272 | }
```


UNKNOWN Arithmetic operation "*" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
276 |  
277 | function claimable(uint _tokenId) external view returns (uint) {  
278 |     uint _last_token_time = last_token_time / WEEK * WEEK;  
279 |     return _claimable(_tokenId, voting_escrow, _last_token_time);  
280 | }
```

UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
276 |  
277 | function claimable(uint _tokenId) external view returns (uint) {  
278 |     uint _last_token_time = last_token_time / WEEK * WEEK;  
279 |     return _claimable(_tokenId, voting_escrow, _last_token_time);  
280 | }
```

UNKNOWN Arithmetic operation "*" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
283 | if (block.timestamp >= time_cursor) _checkpoint_total_supply();  
284 | uint _last_token_time = last_token_time;  
285 | _last_token_time = _last_token_time / WEEK * WEEK;  
286 | uint amount = _claim(_tokenId, voting_escrow, _last_token_time);  
287 | if (amount != 0) {
```

UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
283 | if (block.timestamp >= time_cursor) _checkpoint_total_supply();
284 | uint _last_token_time = last_token_time;
285 | _last_token_time = _last_token_time / WEEK * WEEK;
286 | uint amount = _claim(_tokenId, voting_escrow, _last_token_time);
287 | if (amount != 0) {
```

UNKNOWN Arithmetic operation "-=" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
287 | if (amount != 0) {
288 |     IVotingEscrow(voting_escrow).deposit_for(_tokenId, amount);
289 |     token_last_balance -= amount;
290 | }
291 | return amount;
```

UNKNOWN Arithmetic operation "*" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
295 | if (block.timestamp >= time_cursor) _checkpoint_total_supply();
296 | uint _last_token_time = last_token_time;
297 | _last_token_time = _last_token_time / WEEK * WEEK;
298 | address _voting_escrow = voting_escrow;
299 | uint total = 0;
```

UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
295 | if (block.timestamp >= time_cursor) _checkpoint_total_supply();
296 | uint _last_token_time = last_token_time;
297 | _last_token_time = _last_token_time / WEEK * WEEK;
298 | address _voting_escrow = voting_escrow;
299 | uint total = 0;
```

UNKNOWN Arithmetic operation "++" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
299 | uint total = 0;
300 |
301 | for (uint i = 0; i < _tokenIds.length; i++) {
302 |     uint _tokenId = _tokenIds[i];
303 |     if (_tokenId == 0) break;
```

UNKNOWN Arithmetic operation "+=" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
305 | if (amount != 0) {
306 |     IVotingEscrow(_voting_escrow).deposit_for(_tokenId, amount);
307 |     total += amount;
308 | }
309 | }
```

UNKNOWN Arithmetic operation "-=" discovered

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
309 | }  
310 | if (total != 0) {  
311 | token_last_balance -= total;  
312 | }  
313 |
```

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

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
110 | _min = _mid;  
111 | } else {  
112 | _max = _mid - 1;  
113 | }  
114 | }
```

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

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
126 | _min = _mid;  
127 | } else {  
128 | _max = _mid - 1;  
129 | }  
130 | }
```

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

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
187 | IVotingEscrow.Point memory user_point = IVotingEscrow(ve).user_point_history(_tokenId, user_epoch);
188 |
189 | if (week_cursor == 0) week_cursor = (user_point.ts + WEEK - 1) / WEEK * WEEK;
190 | if (week_cursor >= last_token_time) return 0;
191 | if (week_cursor < _start_time) week_cursor = _start_time;
```

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

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
215 | }
216 |
217 | user_epoch = Math.min(max_user_epoch, user_epoch - 1);
218 | user_epoch_of[_tokenId] = user_epoch;
219 | time_cursor_of[_tokenId] = week_cursor;
```

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

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

SWC-101

Source file

RewardsDistributor.sol

Locations

```
244 | IVotingEscrow.Point memory user_point = IVotingEscrow(ve).user_point_history(_tokenId, user_epoch);
245 |
246 | if (week_cursor == 0) week_cursor = (user_point.ts + WEEK - 1) / WEEK * WEEK;
247 | if (week_cursor >= last_token_time) return 0;
248 | if (week_cursor < _start_time) week_cursor = _start_time;
```

UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

libraries/Math.sol

Locations

```
11 | if (y > 3) {  
12 |   z = y;  
13 |   uint x = y / 2 + 1;  
14 |   while (x < z) {  
15 |     z = x;
```

UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

libraries/Math.sol

Locations

```
11 | if (y > 3) {  
12 |   z = y;  
13 |   uint x = y / 2 + 1;  
14 |   while (x < z) {  
15 |     z = x;
```

UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

libraries/Math.sol

Locations

```
14 | while (x < z) {  
15 |   z = x;  
16 |   x = y / x + x / 2;  
17 | }  
18 | else if (y != 0) {
```

UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file
libraries/Math.sol
Locations

```
14 | while (x < z) {  
15 |     z = x;  
16 |     x = (y / x + x) / 2;  
17 | }  
18 | } else if (y != 0) {
```

UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file
libraries/Math.sol
Locations

```
14 | while (x < z) {  
15 |     z = x;  
16 |     x = (y / x + x) / 2;  
17 | }  
18 | } else if (y != 0) {
```

UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file
libraries/Math.sol
Locations

```
24 | for (uint256 y = 1 << 255; y > 0; y >= 3) {  
25 |     x <<= 1;  
26 |     uint256 z = 5 * x * (x + 1) + 1;  
27 |     if (n / y >= z) {  
28 |         n -= y * z;
```

UNKNOWN Arithmetic operation "*" discovered

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

SWC-101

Source file
libraries/Math.sol
Locations

```
24 | for (uint256 y = 1 << 255; y > 0; y >= 3) {  
25 |     x <= 1;  
26 |     uint256 z = 5 + x * (x + 1) + 1;  
27 |     if (n / y >= z) {  
28 |         n -= y * z;
```

UNKNOWN Arithmetic operation "*" discovered

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

SWC-101

Source file
libraries/Math.sol
Locations

```
24 | for (uint256 y = 1 << 255; y > 0; y >= 3) {  
25 |     x <= 1;  
26 |     uint256 z = 5 + x * (x + 1) + 1;  
27 |     if (n / y >= z) {  
28 |         n -= y * z;
```

UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file
libraries/Math.sol
Locations

```
24 | for (uint256 y = 1 << 255; y > 0; y >= 3) {  
25 |     x <= 1;  
26 |     uint256 z = 3 + x * (x + 1) + 1;  
27 |     if (n / y >= z) {  
28 |         n -= y * z;
```


UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

libraries/Math.sol

Locations

```
25 | x <= 1;
26 | uint256 z = 3 * x * (x + 1) + 1;
27 | if (n / y >= z) {
28 |     n -= y * z;
29 |     x += 1;
```

UNKNOWN Arithmetic operation "-=" discovered

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

SWC-101

Source file

libraries/Math.sol

Locations

```
26 | uint256 z = 3 * x * (x + 1) + 1;
27 | if (n / y >= z) {
28 |     n -= y + z;
29 |     x += 1;
30 | }
```

UNKNOWN Arithmetic operation "*" discovered

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

SWC-101

Source file

libraries/Math.sol

Locations

```
26 | uint256 z = 3 * x * (x + 1) + 1;
27 | if (n / y >= z) {
28 |     n -= y * z;
29 |     x += 1;
30 | }
```

UNKNOWN Arithmetic operation "+=" discovered

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

SWC-101

Source file

libraries/Math.sol

Locations

```
27 | if (n / y >= z) {  
28 |     n -= y * z;  
29 |     x += 1;  
30 | }  
31 | }
```

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

The public state variable "tokens_per_week" in "RewardsDistributor" contract has type "uint256[1000000000000000]" and can cause an exception in case of use of invalid array index value.

SWC-110

Source file

RewardsDistributor.sol

Locations

```
36 |  
37 | uint public last_token_time;  
38 | uint[1000000000000000] public tokens_per_week;  
39 |  
40 | address public voting_escrow;
```

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

The public state variable "ve_supply" in "RewardsDistributor" contract has type "uint256[1000000000000000]" and can cause an exception in case of use of invalid array index value.

SWC-110

Source file

RewardsDistributor.sol

Locations

```
42 | uint public token_last_balance;  
43 |  
44 | uint[1000000000000000] public ve_supply;  
45 |  
46 | address public depositor;
```

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

RewardsDistributor.sol

Locations

```
77 | if (block.timestamp < next_week) {  
78 |   if (since_last == 0 && block.timestamp == t) {  
79 |     tokens_per_week[this_week] += to_distribute;  
80 |   } else {  
81 |     tokens_per_week[this_week] += to_distribute * (block.timestamp - t) / since_last;
```

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

RewardsDistributor.sol

Locations

```
79 | tokens_per_week[this_week] += to_distribute;  
80 | } else {  
81 |   tokens_per_week[this_week] += to_distribute * (block.timestamp - t) / since_last;  
82 | }  
83 | break;
```

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

RewardsDistributor.sol

Locations

```
84 | } else {  
85 |   if (since_last == 0 && next_week == t) {  
86 |     tokens_per_week[this_week] += to_distribute;  
87 |   } else {  
88 |     tokens_per_week[this_week] += to_distribute * (next_week - t) / since_last;
```

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

RewardsDistributor.sol

Locations

```
86 | tokens_per_week[this_week] += to_distribute;
87 | } else {
88 |     tokens_per_week[this_week] += to_distribute * (next_week - t) / since_last;
89 | }
90 | }
```

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

RewardsDistributor.sol

Locations

```
156 | dt = int128(int256(t - pt.ts));
157 | }
158 | ve_supply[t] = Math.max(uint(int256(pt.bias - pt.slope * dt)), 0);
159 | }
160 | t += WEEK;
```

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

RewardsDistributor.sol

Locations

```
209 | if (balance_of == 0 && user_epoch > max_user_epoch) break;
210 | if (balance_of != 0) {
211 |     to_distribute += balance_of * tokens_per_week[week_cursor] / ve_supply[week_cursor];
212 | }
213 | week_cursor += WEEK;
```

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

RewardsDistributor.sol

Locations

```
209 | if (balance_of == 0 && user_epoch > max_user_epoch) break;
210 | if (balance_of != 0) {
211 |   to_distribute += balance_of * tokens_per_week[week_cursor] / ve_supply[week_cursor];
212 | }
213 | week_cursor += WEEK;
```

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

RewardsDistributor.sol

Locations

```
266 | if (balance_of == 0 && user_epoch > max_user_epoch) break;
267 | if (balance_of != 0) {
268 |   to_distribute += balance_of * tokens_per_week[week_cursor] / ve_supply[week_cursor];
269 | }
270 | week_cursor += WEEK;
```

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

RewardsDistributor.sol

Locations

```
266 | if (balance_of == 0 && user_epoch > max_user_epoch) break;
267 | if (balance_of != 0) {
268 |   to_distribute += balance_of * tokens_per_week[week_cursor] / ve_supply[week_cursor];
269 | }
270 | week_cursor += WEEK;
```

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

RewardsDistributor.sol

Locations

```
300 |
301 | for (uint i = 0; i < _tokenIds.length; i++) {
302 |     uint _tokenId = _tokenIds[i];
303 |     if (_tokenId == 0) break;
304 |     uint amount = _claim(_tokenId, _voting_escrow, _last_token_time);
```