MythX

Analysis 34fa1563-0dd9-428f-9249-a5ac1abf14f6

MythX

Started	Mon Aug 23 2021 14:33:11 GMT+0000 (Coordinated Universal Time)
Finished	Mon Aug 23 2021 15:18:34 GMT+0000 (Coordinated Universal Time)
Mode	Deep
Client Tool	Remythx
Main Source File	XTT-TokenTimeLock.Sol

DETECTED VULNERABILITIES

HIGH	MEDIUM	LOW
0	0	0

ISSUES

UNKNOWN Arithmetic operation "+" discovered

SWC-101

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

Source file

IERC20.sol

Locations

```
* Note that `value` may be zero.
*/
event Transfer(address indexed from, address indexed to, uint256 value);

/**
```

UNKNOWN Arithmetic operation "+" discovered

SWC-101

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

Source file

SafeMath.sol

Locations

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

UNKNOWN

SWC-101

Arithmetic operation "-" discovered

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

Source file

SafeMath.sol

Locations

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

```

UNKNOWN

SWC-101

Arithmetic operation "*" discovered

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

Source file

SafeMath.sol

Locations

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

UNKNOWN

SWC-101

Arithmetic operation "/" discovered

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

Source file

SafeMath.sol

Locations

```
if (a == 0) return (true, 0);
uint256 c = a * b;
if (c / a != b) return (false, 0);
return (true, c);
}
```

UNKNOWN

SWC-101

Arithmetic operation "/" discovered

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

Source file

`SafeMath.sol`

Locations

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

```

UNKNOWN

SWC-101

Arithmetic operation "%" discovered

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

Source file

`SafeMath.sol`

Locations

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

```

UNKNOWN

SWC-101

Arithmetic operation "+" discovered

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

Source file

`SafeMath.sol`

Locations

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

UNKNOWN

SWC-101

Arithmetic operation "-" discovered

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

Source file

SafeMath.sol

Locations

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

```

UNKNOWN

SWC-101

Arithmetic operation "*" discovered

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

Source file

SafeMath.sol

Locations

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

UNKNOWN

SWC-101

Arithmetic operation "/" discovered

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

Source file

SafeMath.sol

Locations

```
if (a == 0) return 0;
uint256 c = a * b;
require(c / a == b, "SafeMath: multiplication overflow");
return c;
}
```

UNKNOWN

SWC-101

Arithmetic operation "/" discovered

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

Source file

SafeMath.sol

Locations

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

```

UNKNOWN

SWC-101

Arithmetic operation "%" discovered

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

Source file

SafeMath.sol

Locations

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

```

UNKNOWN

SWC-101

Arithmetic operation "-" discovered

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

Source file

SafeMath.sol

Locations

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

```

UNKNOWN

SWC-101

Arithmetic operation "/" discovered

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

Source file

`SafeMath.sol`

Locations

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

```

UNKNOWN

SWC-101

Arithmetic operation "%" discovered

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

Source file

`SafeMath.sol`

Locations

```
function mod(uint256 a, uint256 b, string memory errorMessage) internal pure returns (uint256) {
require(b > 0, errorMessage);
return a % b;
}
}
```