MythX

Analysis 0d25c5fa-d202-41a5-baf6-d3052249a5e8 MythX

Started	Mon Oct 09 2023 07:18:18 GMT+0000 (Coordinated Universal Time)
Finished	Mon Oct 09 2023 08:03:36 GMT+0000 (Coordinated Universal Time)
Mode	Deep
Client Tool	Mythx-Cli-0.7.3
Main Source File	Contracts/Handlers/Erc20TransferHandler.Sol

DETECTED VULNERABILITIES

HIGH	MEDIUM	LOW
0	0	0

ISSUES

UNKNOWN SWC-101 Arithmetic operation "+" discovered

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

Source file

node_modules/@openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol

Locations

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

UNKNOWN SWC-101 Arithmetic operation "-" discovered

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

Source file

node_modules/@openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol

Locations

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

UNKNOWN

SWC-101

Arithmetic operation "+" discovered

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

Source file

`node_modules/@openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol`

Locations

```
token.permit(owner, spender, value, deadline, v, r, s);
uint256 nonceAfter = token.nonces(owner);
require(nonceAfter == nonceBefore + 1, "SafeERC20: permit did not succeed");
}
```

UNKNOWN

SWC-101

Arithmetic operation "/" discovered

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

Source file

`contracts/handlers/Erc20TransferHandler.sol`

Locations

```

if(feeOnTransferPercentageRate[tokenSymbol] > 0) {
uint256 transferFee = (feeOnTransferPercentageRate[tokenSymbol] * amount) / 1000;
require(amountToSend > transferFee, 'Insufficient amount to send : Percent Rate');
amountToSend = amountToSend - transferFee;
```

UNKNOWN

SWC-101

Arithmetic operation "*" discovered

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

Source file

`contracts/handlers/Erc20TransferHandler.sol`

Locations

```

if(feeOnTransferPercentageRate[tokenSymbol] > 0) {
uint256 transferFee = (feeOnTransferPercentageRate[tokenSymbol] * amount) / 1000;
require(amountToSend > transferFee, 'Insufficient amount to send : Percent Rate');
amountToSend = amountToSend - transferFee;
```

UNKNOWN SWC-101

Arithmetic operation "-" discovered

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

Source file

`contracts/handlers/Erc20TransferHandler.sol`

Locations

```
uint256 transferFee = (feeOnTransferPercentageRate[tokenSymbol] * amount) / 1000;
require(amountToSend > transferFee, 'Insufficient amount to send : Percent Rate');
amountToSend = amountToSend - transferFee;
}
```

UNKNOWN SWC-101

Arithmetic operation "-" discovered

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

Source file

`contracts/handlers/Erc20TransferHandler.sol`

Locations

```
if(feeOnTransferFlatRate[tokenSymbol] > 0) {
require(amountToSend > feeOnTransferFlatRate[tokenSymbol], 'Insufficient amount to send : Flat Rate');
amountToSend = amountToSend - feeOnTransferFlatRate[tokenSymbol];
}
```

UNKNOWN SWC-101

Arithmetic operation "-" discovered

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

Source file

`contracts/handlers/Erc20TransferHandler.sol`

Locations

```
}

return abi.encode(tokenSymbol, amountToSend, amount - amountToSend);
}
```

UNKNOWN SWC-101

Arithmetic operation "/" discovered

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

Source file

`contracts/handlers/Erc20TransferHandler.sol`

Locations

```
uint256 txFee = 0;
if(chargeTxFee) {
txFee = (txFeeRate * amount) / 1000;
uint256 bribeFee = (bribeFeeRate * txFee) / 1000;
uint256 lpFee = txFee - bribeFee;
```

UNKNOWN

SWC-101

Arithmetic operation "*" discovered

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

Source file

`contracts/handlers/Erc20TransferHandler.sol`

Locations

```
uint256 txFee = 0;
if(chargeTxFee) {
txFee = (txFeeRate * amount) / 1000;
uint256 bribeFee = (bribeFeeRate * txFee) / 1000;
uint256 lpFee = txFee - bribeFee;
```

UNKNOWN

SWC-101

Arithmetic operation "/" discovered

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

Source file

`contracts/handlers/Erc20TransferHandler.sol`

Locations

```
if(chargeTxFee) {
txFee = (txFeeRate * amount) / 1000;
uint256 bribeFee = (bribeFeeRate * txFee) / 1000;
uint256 lpFee = txFee - bribeFee;
```

UNKNOWN

SWC-101

Arithmetic operation "*" discovered

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

Source file

`contracts/handlers/Erc20TransferHandler.sol`

Locations

```
if(chargeTxFee) {
txFee = (txFeeRate * amount) / 1000;
uint256 bribeFee = (bribeFeeRate * txFee) / 1000;
uint256 lpFee = txFee - bribeFee;
```

UNKNOWN

SWC-101

Arithmetic operation "-" discovered

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

Source file

contracts/handlers/Erc20TransferHandler.sol

Locations

```
txFee = (txFeeRate * amount) / 1000;
uint256 bribeFee = (bribeFeeRate * txFee) / 1000;
uint256 lpFee = txFee - bribeFee;

if(bribeFee > 0) {
```

UNKNOWN

SWC-101

Arithmetic operation "-" discovered

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

Source file

contracts/handlers/Erc20TransferHandler.sol

Locations

```
}

token.safeTransfer(target, amount - txFee);
}
```

UNKNOWN

SWC-101

Arithmetic operation "/" discovered

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

Source file

contracts/handlers/Erc20TransferHandler.sol

Locations

```

if(feeOnTransferPercentageRate[tokenSymbol] > 0) {
uint256 transferFee = (feeOnTransferPercentageRate[tokenSymbol] * (amount + feeOnTransferAmount)) / 1000;
require(feeOnTransfer >= transferFee, 'Insufficient amount to send : Percent Rate');
feeOnTransfer = feeOnTransfer - transferFee;
```

UNKNOWN

SWC-101

Arithmetic operation "*" discovered

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

Source file

contracts/handlers/Erc20TransferHandler.sol

Locations

```

if(feeOnTransferPercentageRate[tokenSymbol] > 0) {
uint256 transferFee = (feeOnTransferPercentageRate[tokenSymbol] * (amount + feeOnTransferAmount)) / 1000;
require(feeOnTransfer >= transferFee, 'Insufficient amount to send : Percent Rate');
feeOnTransfer = feeOnTransfer - transferFee;
```

UNKNOWN

SWC-101

Arithmetic operation "+" discovered

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

Source file

contracts/handlers/Erc20TransferHandler.sol

Locations

```

if(feeOnTransferPercentageRate[tokenSymbol] > 0) {
uint256 transferFee = (feeOnTransferPercentageRate[tokenSymbol] * (amount + feeOnTransferAmount)) / 1000;
require(feeOnTransfer >= transferFee, 'Insufficient amount to send : Percent Rate');
feeOnTransfer = feeOnTransfer - transferFee;
```

UNKNOWN

SWC-101

Arithmetic operation "-" discovered

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

Source file

contracts/handlers/Erc20TransferHandler.sol

Locations

```
uint256 transferFee = (feeOnTransferPercentageRate[tokenSymbol] * (amount + feeOnTransferAmount)) / 1000;
require(feeOnTransfer >= transferFee, 'Insufficient amount to send : Percent Rate');
feeOnTransfer = feeOnTransfer - transferFee;
}
```

UNKNOWN

SWC-101

Arithmetic operation "-" discovered

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

Source file

contracts/handlers/Erc20TransferHandler.sol

Locations

```
if(feeOnTransferFlatRate[tokenSymbol] > 0) {
require(feeOnTransfer >= feeOnTransferFlatRate[tokenSymbol], 'Insufficient amount to send : Flat Rate');
feeOnTransfer = feeOnTransfer - feeOnTransferFlatRate[tokenSymbol];
}
```

UNKNOWN

SWC-101

Arithmetic operation "+" discovered

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

Source file

contracts/handlers/Erc20TransferHandler.sol

Locations

```
require(feeOnTransfer == 0, 'Invalid fee on transfer amount');

uint256 amountToSend = amount + feeOnTransferAmount;

IERC20 token = IERC20(tokenAddress);
```

UNKNOWN

SWC-101

Arithmetic operation "-" discovered

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

Source file

contracts/handlers/Erc20TransferHandler.sol

Locations

```
token.safeTransferFrom(target, address(this), amountToSend);
uint256 postTransferBalance = token.balanceOf(address(this));
uint256 diffTransferBalance = postTransferBalance - preTransferBalance;
require(diffTransferBalance >= amount, 'Invalid transfer amount');
}
```

UNKNOWN

SWC-101

Arithmetic operation "-" discovered

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

Source file

`contracts/handlers/Erc20TransferHandler.sol`

Locations

```
token.safeTransferFrom(msg.sender, address(this), amount);
uint256 postDepositBalance = token.balanceOf(address(this));
uint256 diffDepositBalance = postDepositBalance - preDepositBalance;

IFeeCollector(feeCollectors[tokenSymbol]).deposit(msg.sender, diffDepositBalance);
```