MythX

REPORT 62501D6725DC4F0019997A2A

Created	Fri Apr 08 2022 11:32:55 GMT+0000 (Coordinated Universal Time)
Number of analyses	1
User	6135edf7a6e184c5d2c6ee1e

REPORT SUMMARY

Analyses ID	Main source file	Detected vulnerabilities
b314e2f4-7b76-4cf5-8087-f2726c5aa872	/contracts/vault.sol	5

Analysis b314e2f4-7b76-4cf5-8087-f2726c5aa872 MythX

Started	Fri Apr 08 2022 11:33:03 GMT+0000 (Coordinated Universal Time)
Finished	Fri Apr 08 2022 11:33:12 GMT+0000 (Coordinated Universal Time)
Mode	Deep
Client Tool	Mythx-Vscode-Extension
Main Source File	/Contracts/Vault.Sol

DETECTED VULNERABILITIES

HIGH	MEDIUM	LOW
0	0	5

ISSUES

UNKNOWN Arithmetic operation "**" discovered

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

SWC-101

Source file
/contracts/vault.sol
Locations

```
30 /*////////////////////////////////////////////////////////////////
31 IMMUTABLES
32 ////////////////////////////////////////////////////////////////*/
33 /// @notice The underlying token the vault accepts.
34 address public immutable override wantToken;
```

UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file
/contracts/vault.sol
Locations

```
98 }
99
100 /// @notice Initiates a withdrawal of vault tokens to the user.
101 /// @param sharesIn The amount of vault tokens to withdraw.
102 /// @param receiver The address to receive the vault tokens.
```

UNKNOWN Arithmetic operation "*" discovered

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

SWC-101

Source file

/contracts/vault.sol

Locations

```
98  }
99
100 /// @notice Initiates a withdrawal of vault tokens to the user.
101 /// @param sharesIn The amount of vault tokens to withdraw.
102 /// @param receiver The address to receive the vault tokens.
```

UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

/contracts/vault.sol

Locations

```
121 }
122
123 /// @notice Calculates the total amount of underlying tokens the vault holds.
124 /// @return The total amount of underlying tokens the vault holds.
125 function totalVaultFunds() public view returns (uint256) {
126 return IERC20(wantToken).balanceOf(address(this)) + totalExecutorFunds();
```

UNKNOWN Arithmetic operation "*" discovered

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

SWC-101

Source file

/contracts/vault.sol

Locations

```
121 }
122
123 /// @notice Calculates the total amount of underlying tokens the vault holds.
124 /// @return The total amount of underlying tokens the vault holds.
125 function totalVaultFunds() public view returns (uint256) {
```

UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

/contracts/vault.sol

Locations

```
129 /*////////////////////////////////////////////////////////////////
130 EXECUTOR DEPOSIT/WITHDRAWAL LOGIC
131 ////////////////////////////////////////////////////////////////////*/
132
133 /// @notice list of trade executors connected to vault.
134 AddrArrayLib.Addresses tradeExecutorsList;
```

UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

/contracts/vault.sol

Locations

```
197 event FeesCollected(uint256 collectedFees);
198
199 //// @notice Calculates and collects the fees from the vault.
200 /// @dev This function sends all the accured fees to governance.
201 /// checks the yield made since previous harvest and
```

UNKNOWN Arithmetic operation "*" discovered

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

SWC-101

Source file

/contracts/vault.sol

Locations

```
214 }
215
216 /*////////////////////////////////////////////////////////////////////
217 EXECUTOR ADDITION/REMOVAL LOGIC
218 ////////////////////////////////////////////////////////////////////*/
```

UNKNOWN Arithmetic operation "-" discovered

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

SWC-101

Source file

/contracts/vault.sol

Locations

```
214 }
215
216 /*////////////////////////////////////////////////////////
217 EXECUTOR ADDITION/REMOVAL LOGIC
218 //////////////////////////////////////////////////////////*/
```

UNKNOWN Arithmetic operation "/" discovered

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

SWC-101

Source file

/contracts/vault.sol

Locations

```
215
216 /*//////////////////////////////////////////////////////////
217 EXECUTOR ADDITION/REMOVAL LOGIC
218 ////////////////////////////////////////////////////////////*/
219 /// @notice Emitted when executor is added to vault.
220 /// @param executor The address of added executor.
```

UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

/contracts/vault.sol

Locations

```
262 }
263
264 /// @notice gives the number of trade executors.
265 /// @return The number of trade executors.
266 function totalExecutors() public view returns (uint256) {
267 return tradeExecutorsList.size();
```

UNKNOWN Arithmetic operation "++" discovered

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

SWC-101

Source file

/contracts/vault.sol

Locations

```
284 .totalFunds();
285 require(block.number <= blockUpdated + BLOCK_LIMIT, "FUNDS_NOT_UPDATED");
286 totalFunds += executorFunds;
287 }
288 return totalFunds;
```

UNKNOWN Arithmetic operation "+" discovered

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

SWC-101

Source file

/contracts/vault.sol

Locations

```
291 /*////////////////////////////////////////////////////////////////
292 GOVERNANCE ACTIONS
293 ////////////////////////////////////////////////////////////////*/
294
295 /// @notice Emitted when a batcher is updated.
```

UNKNOWN Arithmetic operation "+=" discovered

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

SWC-101

Source file

/contracts/vault.sol

Locations

```
293 ////////////////////////////////////////////////////////////////*/
294
295 /// @notice Emitted when a batcher is updated.
296 /// @param oldBatcher The address of the current batcher.
297 /// @param newBatcher The address of new batcher.
```

LOW

SWC-103

A floating pragma is set.

The current pragma Solidity directive is ""^0.8.0"". It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

Source file

/contracts/vault.sol

Locations

```
1  /// SPDX-License-Identifier: GPL-3.0-or-later
2  pragma solidity ^0.8.0;
3
4  import "@openzeppelin/contracts/token/ERC20/ERC20.sol";
```

LOW

SWC-108

State variable visibility is not set.

It is best practice to set the visibility of state variables explicitly. The default visibility for "pendingGovernance" is internal. Other possible visibility settings are public and private.

Source file

/contracts/vault.sol

Locations

```
45 /// @notice Governance address to add/remove executors.
46 address public override governance;
47 address pendingGovernance;
48
49 /// @notice Creates a new Vault that accepts a specific underlying token.
```

LOW

SWC-108

State variable visibility is not set.

It is best practice to set the visibility of state variables explicitly. The default visibility for "tradeExecutorsList" is internal. Other possible visibility settings are public and private.

Source file

/contracts/vault.sol

Locations

```
132
133 /// @notice list of trade executors connected to vault.
134 AddrArrayLib.Addresses tradeExecutorsList;
135
136 /// @notice Emitted after the vault deposits into a executor contract.
```

LOW

SWC-120

Potential use of "block.number" as source of randonmness.

The environment variable "block.number" looks like it might be used as a source of randomness. Note that the values of variables like coinbase, gaslimit, block number and timestamp are predictable and can be manipulated by a malicious miner. Also keep in mind that attackers know hashes of earlier blocks. Don't use any of those environment variables as sources of randomness and be aware that use of these variables introduces a certain level of trust into miners.

Source file

/contracts/vault.sol

Locations

```
262 }
263
264 /// @notice gives the number of trade executors.
265 /// @return The number of trade executors.
266 function totalExecutors() public view returns (uint256) {
```

LOW

SWC-120

Potential use of "block.number" as source of randonmness.

The environment variable "block.number" looks like it might be used as a source of randomness. Note that the values of variables like coinbase, gaslimit, block number and timestamp are predictable and can be manipulated by a malicious miner. Also keep in mind that attackers know hashes of earlier blocks. Don't use any of those environment variables as sources of randomness and be aware that use of these variables introduces a certain level of trust into miners.

Source file

/contracts/vault.sol

Locations

```
291 /*////////////////////////////////////////////////////////////////
292 GOVERNANCE ACTIONS
293 //////////////////////////////////////////////////////////////*/
294
295 /// @notice Emitted when a batcher is updated.
```