Circle Internet Financial – EVM Bridge
Smart Contract Security Audit

Prepared by: Halborn
Date of Engagement: October 10th, 2022 – November 4th, 2022
Visit: Halborn.com
## EXECUTIVE OVERVIEW

### 1.1 INTRODUCTION

### 1.2 AUDIT SUMMARY

### 1.3 TEST APPROACH & METHODOLOGY

### RISK METHODOLOGY

### 1.4 SCOPE

## ASSESSMENT SUMMARY & FINDINGS OVERVIEW

### FINDINGS & TECH DETAILS

#### 3.1 (HAL-01) INCOMPATIBILITY WITH NON-STANDARD ERC20 TOKENS - **MEDIUM**

<table>
<thead>
<tr>
<th>Description</th>
<th>Code Location</th>
<th>Risk Level</th>
<th>Recommendation</th>
<th>Remediation Plan</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### 3.2 (HAL-02) LACK OF INPUT VALIDATION IN REPLACEDEPOSITFORBURN MAY RESULT IN TOKEN LOSS - **MEDIUM**

<table>
<thead>
<tr>
<th>Description</th>
<th>Proof of Concept</th>
<th>Recommendation</th>
<th>Remediation Plan</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### 3.3 (HAL-03) LACK OF TRANSFEROWNERSHIP PATTERN - **LOW**

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Risk Level</td>
</tr>
<tr>
<td>Recommendation</td>
</tr>
<tr>
<td>Remediation Plan</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>3.4 (HAL-04) REMOVEREMOTE_TOKENMESSENGER EMITS EVENT BASING ON USER INPUT - INFORMATIONAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
</tr>
<tr>
<td>Recommendation</td>
</tr>
<tr>
<td>Remediation Plan</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>3.5 (HAL-05) UPDATEATTESTERMANAGER EMITS EVENT WITH INCORRECT DATA - INFORMATIONAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
</tr>
<tr>
<td>Recommendation</td>
</tr>
<tr>
<td>Remediation Plan</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>3.6 (HAL-06) GAS OVER-CONSUMPTION IN LOOPS - INFORMATIONAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
</tr>
<tr>
<td>Code Location</td>
</tr>
<tr>
<td>Proof of Concept</td>
</tr>
<tr>
<td>Risk Level</td>
</tr>
<tr>
<td>Recommendation</td>
</tr>
<tr>
<td>Remediation Plan</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>3.7 (HAL-07) UNNEEDED INITIALIZATION OF UINT256 VARIABLES TO 0 - INFORMATIONAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
</tr>
<tr>
<td>Code Location</td>
</tr>
<tr>
<td>Risk Level</td>
</tr>
<tr>
<td>Recommendation</td>
</tr>
<tr>
<td>Remediation Plan</td>
</tr>
<tr>
<td>Section</td>
</tr>
<tr>
<td>------------------------------</td>
</tr>
<tr>
<td>4 MANUAL TESTING</td>
</tr>
<tr>
<td>5 AUTOMATED TESTING</td>
</tr>
<tr>
<td>5.1 STATIC ANALYSIS REPORT</td>
</tr>
<tr>
<td>Description</td>
</tr>
<tr>
<td>Slither results</td>
</tr>
<tr>
<td>5.2 AUTOMATED SECURITY SCAN</td>
</tr>
<tr>
<td>Description</td>
</tr>
<tr>
<td>MythX results</td>
</tr>
</tbody>
</table>
## DOCUMENT REVISION HISTORY

<table>
<thead>
<tr>
<th>VERSION</th>
<th>MODIFICATION</th>
<th>DATE</th>
<th>AUTHOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>Document Creation</td>
<td>10/10/2022</td>
<td>Grzegorz Trawinski</td>
</tr>
<tr>
<td>0.2</td>
<td>Document Update</td>
<td>10/31/2022</td>
<td>Grzegorz Trawinski</td>
</tr>
<tr>
<td>0.3</td>
<td>Draft Review</td>
<td>10/31/2022</td>
<td>Kubilay Onur Gungor</td>
</tr>
<tr>
<td>0.4</td>
<td>Draft Review</td>
<td>10/31/2022</td>
<td>Gabi Urrutia</td>
</tr>
<tr>
<td>0.5</td>
<td>Draft Update</td>
<td>11/07/2022</td>
<td>Grzegorz Trawinski</td>
</tr>
<tr>
<td>0.6</td>
<td>Draft Review</td>
<td>11/10/2022</td>
<td>Kubilay Onur Gungor</td>
</tr>
<tr>
<td>0.7</td>
<td>Draft Review</td>
<td>11/10/2022</td>
<td>Gabi Urrutia</td>
</tr>
<tr>
<td>1.0</td>
<td>Remediation Plan</td>
<td>12/05/2022</td>
<td>Grzegorz Trawinski</td>
</tr>
<tr>
<td>1.1</td>
<td>Remediation Plan Review</td>
<td>12/05/2022</td>
<td>Roberto Reigada</td>
</tr>
<tr>
<td>1.2</td>
<td>Remediation Plan Review</td>
<td>12/05/2022</td>
<td>Piotr Cielas</td>
</tr>
<tr>
<td>1.3</td>
<td>Remediation Plan Review</td>
<td>12/05/2022</td>
<td>Gabi Urrutia</td>
</tr>
</tbody>
</table>
# CONTACTS

<table>
<thead>
<tr>
<th>CONTACT</th>
<th>COMPANY</th>
<th>EMAIL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rob Behnke</td>
<td>Halborn</td>
<td><a href="mailto:Rob.Behnke@halborn.com">Rob.Behnke@halborn.com</a></td>
</tr>
<tr>
<td>Steven Walbroehl</td>
<td>Halborn</td>
<td><a href="mailto:Steven.Walbroehl@halborn.com">Steven.Walbroehl@halborn.com</a></td>
</tr>
<tr>
<td>Gabi Urrutia</td>
<td>Halborn</td>
<td><a href="mailto:Gabi.Urrutia@halborn.com">Gabi.Urrutia@halborn.com</a></td>
</tr>
<tr>
<td>Grzegorz Trawinski</td>
<td>Halborn</td>
<td><a href="mailto:Grzegorz.Trawinski@halborn.com">Grzegorz.Trawinski@halborn.com</a></td>
</tr>
<tr>
<td>Kubilay Onur Gungor</td>
<td>Halborn</td>
<td><a href="mailto:Kubilay.Gungor@halborn.com">Kubilay.Gungor@halborn.com</a></td>
</tr>
<tr>
<td>Roberto Reigada</td>
<td>Halborn</td>
<td><a href="mailto:Roberto.Reigada@halborn.com">Roberto.Reigada@halborn.com</a></td>
</tr>
<tr>
<td>Piotr Cielas</td>
<td>Halborn</td>
<td><a href="mailto:Piotr.Cielas@halborn.com">Piotr.Cielas@halborn.com</a></td>
</tr>
</tbody>
</table>
EXECUTIVE OVERVIEW
1.1 INTRODUCTION

Circle is a global financial technology company, the creators of USDC and Euro Coin.

Circle Internet Financial engaged Halborn to conduct a security audit on their smart contracts beginning on October 10th, 2022 and ending on November 4th, 2022. The security assessment was scoped to the smart contracts provided to the Halborn team.

1.2 AUDIT SUMMARY

The team at Halborn was provided two weeks for the engagement and assigned a full-time security engineer to audit the security of the smart contract. The security engineer is a blockchain and smart-contract security expert with advanced penetration testing, smart-contract hacking, and deep knowledge of multiple blockchain protocols.

The purpose of this audit is to:

- Ensure that smart contract functions operate as intended
- Identify potential security issues with the smart contracts

In summary, Halborn identified some security risks that were mostly addressed by the Circle Internet Financial team.

1.3 TEST APPROACH & METHODOLOGY

Halborn performed a combination of manual and automated security testing to balance efficiency, timeliness, practicality, and accuracy in regard to the scope of this audit. While manual testing is recommended to uncover flaws in logic, process, and implementation; automated testing techniques help enhance coverage of the bridge code and can quickly identify items
that do not follow security best practices. The following phases and associated tools were used throughout the term of the audit:

- Research into architecture and purpose
- Smart contract manual code review and walkthrough
- Graphing out functionality and contract logic/connectivity/functions. (solgraph)
- Manual assessment of use and safety for the critical Solidity variables and functions in scope to identify any arithmetic related vulnerability classes
- Manual testing by custom scripts
- Scanning of solidity files for vulnerabilities, security hotspots or bugs. (MythX)
- Static Analysis of security for scoped contract, and imported functions. (Slither)
- Testnet deployment (Brownie, Remix IDE, Visual Studio Code)

**RISK METHODOLOGY:**

Vulnerabilities or issues observed by Halborn are ranked based on the risk assessment methodology by measuring the **LIKELIHOOD** of a security incident and the **IMPACT** should an incident occur. This framework works for communicating the characteristics and impacts of technology vulnerabilities. The quantitative model ensures repeatable and accurate measurement while enabling users to see the underlying vulnerability characteristics that were used to generate the Risk scores. For every vulnerability, a risk level will be calculated on a scale of 5 to 1 with 5 being the highest likelihood or impact.

**RISK SCALE - LIKELIHOOD**

5 - Almost certain an incident will occur.
4 - High probability of an incident occurring.
3 - Potential of a security incident in the long term.
2 - Low probability of an incident occurring.
1 - Very unlikely issue will cause an incident.

**RISK SCALE - IMPACT**
5 - May cause devastating and unrecoverable impact or loss.
4 - May cause a significant level of impact or loss.
3 - May cause a partial impact or loss to many.
2 - May cause temporary impact or loss.
1 - May cause minimal or un-noticeable impact.

The risk level is then calculated using a sum of these two values, creating a value of 10 to 1 with 10 being the highest level of security risk.

<table>
<thead>
<tr>
<th>CRITICAL</th>
<th>HIGH</th>
<th>MEDIUM</th>
<th>LOW</th>
<th>INFORMATIONAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 - CRITICAL</td>
<td>9 - 8 - HIGH</td>
<td>7 - 6 - MEDIUM</td>
<td>5 - 4 - LOW</td>
<td>3 - 1 - VERY LOW AND INFORMATIONAL</td>
</tr>
</tbody>
</table>
1.4 SCOPE

IN-SCOPE:
The security assessment was scoped to the following evm-bridge-contracts:

- MessageTransmitter.sol
- TokenMessenger.sol
- TokenMinter.sol
- roles/TokenController.sol
- roles/Rescuable.sol
- roles/Pausable.sol
- roles/Ownable.sol
- roles/Attestable.sol
- messages/Message.sol
- messages/BurnMessage.sol

Commit ID: 7092d95eb35a49e404af349fc4ee5735a630e04c

Additionally, Circle Internet Financial team requested to include third-party library TypedMemView.sol into the scope of the assessment.

Commit ID: 3071bb11a8f87dface65846f3f12bba2ddf16add8

OUT-OF-SCOPE:
Other smart contracts in the repository, external libraries and economical attacks.
### 2. ASSESSMENT SUMMARY & FINDINGS

#### OVERVIEW

<table>
<thead>
<tr>
<th>CRITICAL</th>
<th>HIGH</th>
<th>MEDIUM</th>
<th>LOW</th>
<th>INFORMATIONAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>4</td>
</tr>
</tbody>
</table>

#### LIKELIHOOD

- (HAL-01) (HAL-02)
- (HAL-04) (HAL-05)
- (HAL-03)
- (HAL-06) (HAL-07)
<table>
<thead>
<tr>
<th>SECURITY ANALYSIS</th>
<th>RISK LEVEL</th>
<th>REMEDIATION DATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>HAL-01 - INCOMPATIBILITY WITH NON-STANDARD ERC20 TOKENS</td>
<td>Medium</td>
<td>RISK ACCEPTED</td>
</tr>
<tr>
<td>HAL-02 - LACK OF INPUT VALIDATION IN REPLACEDEPOSITFORBURN MAY RESULT IN TOKEN LOSS</td>
<td>Medium</td>
<td>SOLVED - 12/05/2022</td>
</tr>
<tr>
<td>HAL-03 - LACK OF TRANSFER-OWNERSHIP PATTERN</td>
<td>Low</td>
<td>SOLVED - 12/05/2022</td>
</tr>
<tr>
<td>HAL-04 - REMOVEREMOTETOKENMESSENGER EMITS EVENT BASING ON USER INPUT</td>
<td>Informational</td>
<td>SOLVED - 12/05/2022</td>
</tr>
<tr>
<td>HAL-05 - UPDATEATTESTERMANAGER EMITS EVENT WITH INCORRECT DATA</td>
<td>Informational</td>
<td>SOLVED - 12/05/2022</td>
</tr>
<tr>
<td>HAL-06 - GAS OVER-CONSUMPTION IN LOOPS</td>
<td>Informational</td>
<td>SOLVED - 12/05/2022</td>
</tr>
<tr>
<td>HAL-07 - UNNEEDED INITIALIZATION OF UINT256 VARIABLES TO 0</td>
<td>Informational</td>
<td>SOLVED - 12/05/2022</td>
</tr>
</tbody>
</table>
FINDINGS & TECH DETAILS
3.1 (HAL-01) INCOMPATIBILITY WITH NON-STANDARD ERC20 TOKENS - MEDIUM

Description:

Some tokens (such as USDT) do not properly implement the EIP20 standard and their `transfer/transferFrom` functions return `void`, instead of a boolean. Calling these functions with the correct EIP20 function signatures will always revert as it does in the `_depositForBurn()` function in the TokenMessenger contract.

Tokens that do not correctly implement the latest EIP20 spec, such as USDT, will not be able to be used in the smart contract as they revert the transaction due to missing return value.

It is recommended using the SafeERC20 versions of OpenZeppelin with the `safeTransfer` and `safeTransferFrom` functions that handle return value check as well as non-standard compliant tokens.

Code Location:

```
function _depositForBurn(
  uint256 _amount,
  uint32 _destinationDomain,
  bytes32 _mintRecipient,
  address _burnToken,
  bytes32 _destinationCaller
) internal returns (uint64 nonce) {
  require(_amount > 0, "Amount must be nonzero");
  require(_mintRecipient != bytes32(0), "Mint recipient must be nonzero");

  bytes32 _destinationTokenMessenger =
    _getRemoteTokenMessenger(_destinationDomain,

Listing 1: TokenMessenger.sol (Lines 434-441)
```
ITokenMinter _localMinter = _getLocalMinter();
IMintBurnToken _mintBurnToken = IMintBurnToken(_burnToken)

require(
    _mintBurnToken.transferFrom(
        msg.sender,
        address(_localMinter),
        _amount
    ),
    "Transfer operation failed"
);

_localMinter.burn(_burnToken, _amount);

// Format message body
bytes memory _burnMessage = BurnMessage._formatMessage(
    messageBodyVersion,
    Message.addressToBytes32(_burnToken),
    _mintRecipient,
    _amount,
    Message.addressToBytes32(msg.sender)
);

uint64 _nonceReserved = _sendDepositForBurnMessage(
    _destinationDomain,
    _destinationTokenMessenger,
    _destinationCaller,
    _burnMessage
);

emit DepositForBurn(
    _nonceReserved,
    _burnToken,
    _amount,
    msg.sender,
    _mintRecipient,
    _destinationDomain,
    _destinationTokenMessenger,
    _destinationCaller
);

return _nonceReserved;
Risk Level:

Likelihood - 1
Impact - 5

Recommendation:

It is recommended to use SafeERC20: safeTransfer() and safeTransferFrom().

Remediation Plan:

RISK ACCEPTED: The Circle team is aware of the finding, but it is not expected to support any tokens with solution's legacy implementation of transfer/transferFrom.
3.2 (HAL-02) LACK OF INPUT VALIDATION IN REPLACEDEPOSITFORBURN MAY RESULT IN TOKEN LOSS - MEDIUM

Description:

The `replaceDepositForBurn()` function of the `TokenMessenger.sol` contract does not check the `newMintRecipient` parameter address zero. Instead, the `_depositForBurn()` internal function performs such a check. Additionally, the `replaceDepositForBurn()` function accepts a second address for the `newDestinationCaller` parameter, which can be set to the address zero, and the solution is capable of handling such a situation (empty `destinationCaller` means any address can call the `receiveMessage` function).

On the other hand, it is not possible to update `newDestinationCaller` without updating the `newMintRecipient`. Lack of validation increase the risk that the user may unintentionally and accidentally provide a zero address for the `newMintRecipient` parameter. As a result, the user would not receive tokens transferred between chains.

Listing 2: TokenMessenger.sol (Line 251)

```solidity
function replaceDepositForBurn(
    bytes memory originalMessage,
    bytes calldata originalAttestation,
    bytes32 newDestinationCaller,
    bytes32 newMintRecipient
) external {
    bytes29 _originalMsg = originalMessage.ref(0);
    bytes29 _originalMsgBody = _originalMsg._messageBody();
    bytes32 _originalMsgSender = _originalMsgBody.
    _getMessageSender();
    // _originalMsgSender must match msg.sender of original message
    require(
        msg.sender == Message.bytes32ToAddress(
            _originalMsgSender),
        "Invalid sender for message"
    );
```
bytes32 _burnToken = _originalMsgBody._getBurnToken();
uint256 _amount = _originalMsgBody._getAmount();

bytes memory _newMessageBody = BurnMessage._formatMessage(
    messageBodyVersion,
    _burnToken,
    newMintRecipient,
    _amount,
    _originalMsgSender
);

localMessageTransmitter.replaceMessage(
    originalMessage,
    originalAttestation,
    _newMessageBody,
    newDestinationCaller
);

emit DepositForBurn(
    _originalMsg._nonce(),
    Message.bytes32ToAddress(_burnToken),
    _amount,
    msg.sender,
    newMintRecipient,
    _originalMsg._destinationDomain(),
    _originalMsg._recipient(),
    newDestinationCaller
);

Listing 3: TokenMessenger.sol (Line 426)

function _depositForBurn(
    uint256 _amount,
    uint32 _destinationDomain,
    bytes32 _mintRecipient,
    address _burnToken,
    bytes32 _destinationCaller
) internal returns (uint64 nonce) {
    require(_amount > 0, "Amount must be nonzero");
    require(_mintRecipient != bytes32(0), "Mint recipient must be nonzero");
bytes32 _destinationTokenMessenger = _getRemoteTokenMessenger(_destinationDomain);

ITokenMinter _localMinter = _getLocalMinter();
IMintBurnToken _mintBurnToken = IMintBurnToken(_burnToken);

require(_mintBurnToken.transferFrom(msg.sender, address(_localMinter), _amount), "Transfer operation failed");

_localMinter.burn(_burnToken, _amount);

// Format message body
bytes memory _burnMessage = BurnMessage._formatMessage(
    messageBodyVersion,
    Message.addressToBytes32(_burnToken),
    _mintRecipient,
    _amount,
    Message.addressToBytes32(msg.sender)
);

uint64 _nonceReserved = _sendDepositForBurnMessage(
    _destinationDomain,
    _destinationTokenMessenger,
    _destinationCaller,
    _burnMessage
);

emit DepositForBurn(
    _nonceReserved,
    _burnToken,
    _amount,
    msg.sender,
    _mintRecipient,
    _destinationDomain,
    _destinationTokenMessenger,
    _destinationCaller
);
Proof of Concept:

1. All necessary contracts are deployed: MessageTransmitter, TokenMessenger, TokenMinter, and MockMintBurnToken for the source and destination.
2. Configure all contracts, set `burnLimitPerTransaction` to $10^{16}$. Link token pairs between source and destination.
3. As Source User 4 `depositForBurn` $10^{16}$ of tokens for Destination User 6.
4. As Source User 4 again `depositForBurn` $10^{16}$ of tokens for Destination User 6.
5. As Source User 4 calls `replaceDepositForBurn` for the message from step 4 with the destination caller set to Destination User 7. Set the mint recipient as zero address.
6. As Destination User 6 `receiveMessage` from the step 3.
7. As Destination User 7 `receiveMessage` from the step 5.
8. Observe the users’ balances. Note that Destination User 7 did not receive a cross-chains transfer.

Recommendation:

It is recommended to add a validation check for the `newMintRecipient` parameter against the zero address value to remove the risk related to human errors.
Remediation Plan:

**SOLVED:** The Circle team solved this issue in commit f2cc3448aaa827a029825a2f47256f86615f9744: the `newMintRecipient` address is now checked against the zero-byte value.
3.3 (HAL-03) LACK OF TRANSFER OWNERSHIP PATTERN - LOW

Description:

The transfer of current ownership for the TokenMinter.sol, TokenMessenger.sol, and MessageTransmitter.sol contracts implies that the current owner calls the transferOwnership() function from the Ownable contract:

Listing 4: Ownable.sol

```solidity
function transferOwnership(address newOwner) external onlyOwner {
    require(
        newOwner != address(0),
        "Ownable: new owner is the zero address"
    );
    emit OwnershipTransferred(_owner, newOwner);
    setOwner(newOwner);
}
```

Suppose the nominated EOA account is invalid. In that case, the owner can accidentally transfer ownership to an uncontrolled account, losing access to all functions with the onlyOwner modifier.

The same issue is identified in the Attestable.sol contract.

Listing 5: Attestable.sol

```solidity
function updateAttesterManager(address newAttesterManager) external onlyAttesterManager {
    require(
        newAttesterManager != address(0),
        "Invalid attester manager address"
    );
    _setAttesterManager(newAttesterManager);
    emit AttesterManagerUpdated(newAttesterManager, newAttesterManager);
```
Risk Level:

Likelihood - 2
Impact - 2

Recommendation:

It is recommended to implement a zero address check in the function and a two-step process where the owner nominates an account. The nominated account needs to call an acceptOwnership() function to transfer ownership to be fully successful. This ensures that the nominated EOA account is valid and active.

Remediation Plan:

SOLVED: The Circle team solved this issue in commit f2cc3448aaa827a029825a2f47256f86615f9744: the Ownable2Step contract is now used across the solution.
3.4 (HAL-04) REMOVE REMOTE TOKEN MESSENGER EMITS EVENT BASING ON USER INPUT - INFORMATIONAL

Description:

The `removeRemoteTokenMessenger()` function of the `TokenMessenger.sol` contract emits the `RemoteTokenMessengerRemoved` event based on the user input, while the `tokenMessenger` value could be obtained from the `remoteTokenMessengers` collection. In rare cases, the present implementation may result in emitting events with inaccurate data.

Listing 6: TokenMessenger.sol
```solidity
/**
 * @notice Emitted when a remote TokenMessenger is removed
 * @param domain remote domain
 * @param tokenMessenger TokenMessenger on remote domain
 */

event RemoteTokenMessengerRemoved(
  uint32 indexed domain,
  bytes32 indexed tokenMessenger
);
```

Listing 7: TokenMessenger.sol (Lines 364,375)
```solidity
function removeRemoteTokenMessenger(uint32 domain, bytes32 tokenMessenger)
  external onlyOwner
{
  // No TokenMessenger set for given remote domain.
  require(
    remoteTokenMessengers[domain] != bytes32(0),
    "No TokenMessenger set"
  );
```
Recommendation:

It is recommended to emit the `RemoteTokenMessengerRemoved` event based on the value obtained from the contract data rather than user input.

Remediation Plan:

**SOLVED:** The Circle team solved this issue in commit `f2cc3448aaa029825a2f47256f86615f9744`: the `RemoteTokenMessengerRemoved` event is now based on the value obtained from the contract’s data.
3.5 (HAL-05) UPDATEATTESTERMANAGER EMITS EVENT WITH INCORRECT DATA - INFORMATIONAL

Description:

The `updateAttesterManager()` function from the `Attestable.sol` contract emits the `AttesterManagerUpdated` event using the `newAttesterManager` input parameter twice, instead of the `_attesterManager` parameter for `previousAttesterManager`.

**Listing 8: Attestable.sol**

```solidity
/**
 * @dev Emitted when attester manager address is updated
 * \* @param previousAttesterManager representing the address of
 * \* the previous attester manager
 * \* @param newAttesterManager representing the address of the
 * \* new attester manager
 * */

event AttesterManagerUpdated(
    address indexed previousAttesterManager,
    address indexed newAttesterManager
);
```

**Listing 9: Attestable.sol (Lines 118,127)**

```solidity
function updateAttesterManager(address newAttesterManager) external
onlyAttesterManager
{
    require(
        newAttesterManager != address(0),
        "Invalid attester manager address"
    );
    _setAttesterManager(newAttesterManager);
    emit AttesterManagerUpdated(newAttesterManager,
        newAttesterManager);
}
```
Recommendation:

It is recommended to emit the `AttesterManagerUpdated` with previous and new address values.

Remediation Plan:

**SOLVED:** The Circle team solved this issue in commit `f2cc3448aaa827a029825a2f47256f86615f9744`: the `AttesterManagerUpdated` event is emitted with previous and new address values.
3.6 (HAL-06) GAS OVER-CONSUMPTION IN LOOPS - INFORMATIONAL

Description:

In all the loops, the counter variable is incremented using \texttt{i++}. It is known that, in loops, using \texttt{++i} costs less gas per iteration than \texttt{i++}.

Code Location:

\texttt{Attestable.sol}
- Line 233: \texttt{for (uint256 i = 0; i < signatureThreshold; i++){}

Proof of Concept:

For example, based in the following test contract:

\begin{Verbatim}
Listing 10: Test.sol
1 // SPDX-License-Identifier: MIT
2 pragma solidity ^0.8.9;
3
4 contract test {
5     function postiincrement(uint256 iterations) public {
6         for (uint256 i = 0; i < iterations; i++) {
7             \}
8     }
9     function preiincrement(uint256 iterations) public {
10         for (uint256 i = 0; i < iterations; ++i) {
11             \}
12     }
13 }
\end{Verbatim}

Differences in the gas costs:
Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

It is recommended to use `++i` instead of `i++` to increment the value of an `uint` variable inside a loop to save some gas. This is not applicable outside of loops.

Remediation Plan:

SOLVED: The Circle team solved this issue in commit `f2cc3448aa827a029825a2f47256f86615f9744`: the solution now uses `++i` to increment the value of a `uint` variable inside a loop.
3.7 (HAL-07) UNNEEDED INITIALIZATION OF UINT256 VARIABLES TO 0 - INFORMATIONAL

Description:

As \( i \) is an \( \text{uint256} \), it is already initialized to 0. \( \text{uint256} \ i = 0 \) reassigns the 0 to \( i \) which wastes gas.

Code Location:

\texttt{Attestable.sol}
- Line 233: \( \text{for} \ (\text{uint256} \ i = 0; \ i < \text{signatureThreshold}; \ i++)\{

Risk Level:

\textbf{Likelihood} - 1
\textbf{Impact} - 1

Recommendation:

It is recommended to not initialize \( \text{uint256} \) variables to 0 to save some gas. For example, use instead:
\texttt{for (\text{uint256} \ i; \ i < \text{proposal.targets.length}; \ ++\ i)}.

Remediation Plan:

\textbf{SOLVED:} The \texttt{Circle team} solved this issue in commit \texttt{f2cc3448aaa827a029825a2f47256f86615f9744}: the solution now does not initialize a \texttt{uint} variable to 0 value.
MANUAL TESTING
Halborn performed several manual tests in the MessageTransmitter.sol, TokenMessenger.sol, TokenMinter.sol, TokenController.sol, Attestable.sol contracts:

```solidity
[+] Deployment
message: MessageTransmitter.deploy fron: owner
Transaction sent: 0x6c672a9d26ed6e3a3c187bf3f90c13e6156f5a977fb9b910ffdec8e2379ce
Gas price: 0.6 gwei Gas limit: 12000000Nonce: 253
MessageContract constructor confirmed Block: 150 Gas used: 10628 (0.800)
MessageTransmitter deployed at: 0x123890def236c57f123efc5b69a0d4a89b894a8

Calling -> sourceMessageTransmitter = MessageTransmitter.deploy(sourceDomain, ZERO_ADDRESS, maxMessageBodySize, version, from: owner)
Transaction sent: 0x6c672a9d26ed6e3a3c187bf3f90c13e6156f5a977fb9b910ffdec8e2379ce
Gas price: 0.6 gwei Gas limit: 12000000Nonce: 253
MessageTransmitter constructor confirmed Block: 154 Gas used: 334437 (26.124)
MessageTransmitter deployed at: 0x123890def236c57f123efc5b69a0d4a89b894a8

Calling -> destinationMessageTransmitter = MessageTransmitter.deploy(sourceDomain, ZERO_ADDRESS, maxMessageBodySize, version, from: owner)
Transaction sent: 0x6c672a9d26ed6e3a3c187bf3f90c13e6156f5a977fb9b910ffdec8e2379ce
Gas price: 0.6 gwei Gas limit: 12000000Nonce: 253
MessageTransmitter constructor confirmed Block: 154 Gas used: 334437 (26.124)
MessageTransmitter deployed at: 0x123890def236c57f123efc5b69a0d4a89b894a8

sourceTokenMessenger = TokenMessenger.deploy(sourceMessageTransmitter, messageBodyVersion, from: owner)
Transaction sent: 0x6c672a9d26ed6e3a3c187bf3f90c13e6156f5a977fb9b910ffdec8e2379ce
Gas price: 0.6 gwei Gas limit: 12000000Nonce: 253
TokenMessenger constructor confirmed Block: 154 Gas used: 224626 (1.673)
TokenMessenger deployed at: 0x123890def236c57f123efc5b69a0d4a89b894a8

destinationTokenMessenger = TokenMessenger.deploy(destinationMessageTransmitter, messageBodyVersion, from: owner)
Transaction sent: 0x6c672a9d26ed6e3a3c187bf3f90c13e6156f5a977fb9b910ffdec8e2379ce
Gas price: 0.6 gwei Gas limit: 12000000Nonce: 253
TokenMessenger constructor confirmed Block: 154 Gas used: 224626 (1.673)
TokenMessenger deployed at: 0x123890def236c57f123efc5b69a0d4a89b894a8

sourceTokenMinter = TokenMinter.deployAnyAddress, from: owner)
Transaction sent: 0x6c672a9d26ed6e3a3c187bf3f90c13e6156f5a977fb9b910ffdec8e2379ce
Gas price: 0.6 gwei Gas limit: 12000000Nonce: 253
TokenMinter constructor confirmed Block: 154 Gas used: 144679 (12.80)
TokenMinter deployed at: 0x123890def236c57f123efc5b69a0d4a89b894a8

destinationTokenMinter = TokenMinter.deployAnyAddress, from: owner)
Transaction sent: 0x6c672a9d26ed6e3a3c187bf3f90c13e6156f5a977fb9b910ffdec8e2379ce
Gas price: 0.6 gwei Gas limit: 12000000Nonce: 253
TokenMinter constructor confirmed Block: 154 Gas used: 144679 (12.80)
TokenMinter deployed at: 0x123890def236c57f123efc5b69a0d4a89b894a8

sourceTokenMinter.addLocalTokenMessenger(sourceTokenMessenger, from: owner)
Transaction sent: 0x6c672a9d26ed6e3a3c187bf3f90c13e6156f5a977fb9b910ffdec8e2379ce
Gas price: 0.0 gwei Gas limit: 12000000Nonce: 274
TokenMinter.addLocalTokenMessenger confirmed Block: 470 Gas used: 45465 (0.380)

destinationTokenMinter.addLocalTokenMessenger(destinationTokenMessenger, from: owner)
Transaction sent: 0x6c672a9d26ed6e3a3c187bf3f90c13e6156f5a977fb9b910ffdec8e2379ce
Gas price: 0.0 gwei Gas limit: 12000000Nonce: 275
TokenMinter.addLocalTokenMessenger confirmed Block: 471 Gas used: 45465 (0.380)

sourceTokenMinter.addLocalMinter(sourceTokenMinter, from: owner)
Transaction sent: 0x6c672a9d26ed6e3a3c187bf3f90c13e6156f5a977fb9b910ffdec8e2379ce
Gas price: 0.0 gwei Gas limit: 12000000Nonce: 276
TokenMinter.addLocalMinter confirmed Block: 472 Gas used: 45424 (0.380)

destinationTokenMinter.addLocalMinter(destinationTokenMinter, from: owner)
Transaction sent: 0x6c672a9d26ed6e3a3c187bf3f90c13e6156f5a977fb9b910ffdec8e2379ce
Gas price: 0.0 gwei Gas limit: 12000000Nonce: 277
TokenMinter.addLocalMinter confirmed Block: 473 Gas used: 45424 (0.380)

sourceMockMintBurnToken = MockMintBurnToken.deploy('from: owner)
Transaction sent: 0x6c672a9d26ed6e3a3c187bf3f90c13e6156f5a977fb9b910ffdec8e2379ce
Gas price: 0.0 gwei Gas limit: 12000000Nonce: 278
MockMintBurnToken constructor confirmed Block: 474 Gas used: 541799 (4.56)
MockMintBurnToken deployed at: 0x6c672a9d26ed6e3a3c187bf3f90c13e6156f5a977fb9b910ffdec8e2379ce

destinationMockMintBurnToken = MockMintBurnToken.deploy('from: owner)
Transaction sent: 0x6c672a9d26ed6e3a3c187bf3f90c13e6156f5a977fb9b910ffdec8e2379ce
Gas price: 0.0 gwei Gas limit: 12000000Nonce: 279
MockMintBurnToken constructor confirmed Block: 475 Gas used: 541799 (4.56)
MockMintBurnToken deployed at: 0x6c672a9d26ed6e3a3c187bf3f90c13e6156f5a977fb9b910ffdec8e2379ce
```
--- receive Message ---

destinationMessageTransmitter.receiveMessage(messageToSign, signed_message, signature, {from: owner})
Transaction sent: 0xfbd83b419d8a2c64f3a39e7b28e48d27d323435761022027c7713bb0b0
Gas price: 0.0 gwei Gas limit: 12000000 None: 286
MessageTransmitter.receiveMessage confirmed Block: 492 Gas used: 184859 (0.87%)

sourceMockMintBurnToken.balanceOf(sourceUser1) 990,000,000,000,000,000
sourceMockMintBurnToken.balanceOf(sourceUser5) 1,000,000,000,000,000,000
destinationMockMintBurnToken.balanceOf(destinationUser5) 1,000,000,000,000,000,000
sourceUser4.address 0x6ca0b3e2e6a6e47f24319d56bae174ae8
sourceUser5.address 0xb7c4a97f28e4e3eb3a3c28c8d6c498f615
destinationUser6.address 0x0446e84a626f76e4a532670a8e670a59654738
destinationUser7.address 0x3882b2b6b34d4c91ca1786df593f8c446d48

--- replay receive Message ---

destinationMessageTransmitter.receiveMessage(messageToSign, signed_message, signature, {from: owner})
Transaction sent: 0x29e5120a3ad9b6578d6bfdf37feec488e6e37c043b48082f57b51c2419f69a
Gas price: 0.0 gwei Gas limit: 12000000 None: 287
MessageTransmitter.receiveMessage confirmed (None already used) Block: 493 Gas used: 39539 (0.33%)

--- depositForBurnWithCaller ---

depositForBurnWithCaller.depositForBurnMessage(burnLimitPerTransaction, {from: sourceUser5})
Transaction sent: 0x4375b5777ba594f91a5c6f004015b76a1960049579e7f5144a72
Gas price: 0.0 gwei Gas limit: 12000000 None: 288
MockCoinBurnToken.approve confirmed Block: 493 Gas used: 1111 (0.003%)

destinationMockMintBurnToken.balanceOf(sourceUser5) 990,000,000,000,000,000
destinationMockMintBurnToken.balanceOf(destinationUser2) 1,010,000,000,000,000,000
sourceUser5.address 0x6ca0b3e2e6a6e47f24319d56bae174ae8
sourceUser6.address 0x6ca0b3e2e6a6e47f24319d56bae174ae8
destinationUser1.address 0x0446e84a626f76e4a532670a8e670a59654738
destinationUser2.address 0x3882b2b6b34d4c91ca1786df593f8c446d48

--- receive Message ---

destinationMessageTransmitter.receiveMessage(messageToSign, signed_message, signature, {from: owner})
Transaction sent: 0x83cb87a873b5d9918c54b2e6dca397e126b3a26934952934713e8ae4d4f6998
Gas price: 0.0 gwei Gas limit: 12000000 None: 16
MessageTransmitter.receiveMessage confirmed (invalid caller for message) Block: 496 Gas used: 39248 (0.33%)

sourceMockMintBurnToken.balanceOf(sourceUser4) 990,000,000,000,000,000
sourceMockMintBurnToken.balanceOf(sourceUser5) 1,010,000,000,000,000,000
destinationMockMintBurnToken.balanceOf(destinationUser1) 1,010,000,000,000,000,000
sourceUser4.address 0x6ca0b3e2e6a6e47f24319d56bae174ae8
sourceUser5.address 0xb7c4a97f28e4e3eb3a3c28c8d6c498f615
destinationUser1.address 0x0446e84a626f76e4a532670a8e670a59654738
destinationUser2.address 0x3882b2b6b34d4c91ca1786df593f8c446d48

--- depositForBurn (and replace DepositForBurn later) ---

sourceMockMintBurnToken.approve(sourceTokenMessage, burnLimitPerTransaction + 1, {from: sourceUser4})
Transaction sent: 0x83cb87a873b5d9918c54b2e6dca397e126b3a26934952934713e8ae4d4f6998
Gas price: 0.0 gwei Gas limit: 12000000 None: 199
MockCoinBurnToken.approve confirmed Block: 498 Gas used: 29282 (0.24%)

sourceTokenMessage.depositForBurn(burnLimitPerTransaction, destinationDomain, destinationUserBytes, MockCoinBurnToken, {from: sourceUser4})
Transaction sent: 0x83cb87a873b5d9918c54b2e6dca397e126b3a26934952934713e8ae4d4f6998
Gas price: 0.0 gwei Gas limit: 12000000 None: 118
TokenMessage.depositForBurn confirmed Block: 499 Gas used: 6842 (0.074%)

sourceMockMintBurnToken.balanceOf(sourceUser4) 990,000,000,000,000,000
sourceMockMintBurnToken.balanceOf(sourceUser5) 1,010,000,000,000,000,000
destinationMockMintBurnToken.balanceOf(destinationUser1) 1,010,000,000,000,000,000
sourceUser4.address 0x6ca0b3e2e6a6e47f24319d56bae174ae8
sourceUser5.address 0xb7c4a97f28e4e3eb3a3c28c8d6c498f615
destinationUser1.address 0x0446e84a626f76e4a532670a8e670a59654738
destinationUser2.address 0x3882b2b6b34d4c91ca1786df593f8c446d48

--- MANUAL TESTING ---

34
sourceMockKillMintBurnToken.balanceOf(sourceUser4) 980,000,000,000,000,000,000
sourceMockKillMintBurnToken.balanceOf(sourceUser5) 990,000,000,000,000,000,000

destinationMockKillMintBurnToken.balanceOf(destinationUser6) 1,010,000,000,000,000,000,000
destinationMockKillMintBurnToken.balanceOf(destinationUser7) 1,020,000,000,000,000,000,000

sourceUser4.address 0x4d0ca532e643e4f7d103149d50b4210e17a4e84
sourceUser5.address 0x807c47a89f720fe4e9b8343c286fc886f43191b
destinationUser6.address 0x844ec86426f876647a5362706a0457e5a965473b
destinationUser7.address 0x23b82b6bc340d4c91ca4a78edf6593c5c4e4dd4b

--- set up attesters ---

sourceMessageTransmitter.isEnabledAttester(attester1) True
destinationMessageTransmitter.isEnabledAttester(attester1) True
sourceMessageTransmitter.isEnabledAttester(attester2) False
destinationMessageTransmitter.isEnabledAttester(attester2) False
sourceMessageTransmitter.isEnabledAttester(attester3) False
destinationMessageTransmitter.isEnabledAttester(attester3) False

sourceMessageTransmitter.setEnabledAttester(attester1, 'from: owner')
destinationMessageTransmitter.setEnabledAttester(attester1, 'from: owner')

sourceMessageTransmitter.setEnabledAttester(attester2, 'from: owner')
destinationMessageTransmitter.setEnabledAttester(attester2, 'from: owner')

sourceMessageTransmitter.setEnabledAttester(attester3, 'from: owner')
destinationMessageTransmitter.setEnabledAttester(attester3, 'from: owner')

sourceMessageTransmitter.setEnabledAttester(attester2, 'from: owner')
destinationMessageTransmitter.setEnabledAttester(attester2, 'from: owner')

sourceMessageTransmitter.setEnabledAttester(attester3, 'from: owner')
destinationMessageTransmitter.setEnabledAttester(attester3, 'from: owner')

sourceMessageTransmitter.setEnabledAttester(attester2, 'from: owner')
destinationMessageTransmitter.setEnabledAttester(attester2, 'from: owner')

sourceMessageTransmitter.setEnabledAttester(Attester already enabled) Block: 540
destinationMessageTransmitter.setEnabledAttester(Attester already enabled) Block: 541

sourceMessageTransmitter.setEnabledAttester(Attester already enabled) Block: 540
destinationMessageTransmitter.setEnabledAttester(Attester already enabled) Block: 541

sourceMessageTransmitter.setEnabledAttester(Attester already enabled) Block: 540
destinationMessageTransmitter.setEnabledAttester(Attester already enabled) Block: 541

sourceMessageTransmitter.setEnabledAttester(Attester already enabled) Block: 540
destinationMessageTransmitter.setEnabledAttester(Attester already enabled) Block: 541

sourceMessageTransmitter.setEnabledAttester(Attester already enabled) Block: 540
destinationMessageTransmitter.setEnabledAttester(Attester already enabled) Block: 541
The manual tests were focused on testing the main functions of these contracts:
- `addLocalTokenMessenger()`
- `addLocalMinter()`
- `setMaxBurnAmountPerTransaction()`
- `linkTokenPair()`
- `unlinkTokenPair()`
- `addRemoteTokenMessenger()`
- `depositForBurn()`
- `receiveMessage()`
- `depositForBurnWithCaller()`
- `replaceMessage()`
- `enableAttester()`
- `isEnabledAttester()`
- `setSignatureThreshold()`

Apart from one medium finding, no significant issues were found during the manual tests.
AUTOMATED TESTING
5.1 STATIC ANALYSIS REPORT

Description:

Halborn used automated testing techniques to enhance the coverage of certain areas of the scoped contracts. Among the tools used was Slither, a Solidity static analysis framework. After Halborn verified all the contracts in the repository and was able to compile them correctly into their ABI and binary formats, Slither was run on the all-scoped contracts. This tool can statically verify mathematical relationships between Solidity variables to detect invalid or inconsistent usage of the contracts’ APIs across the entire code-base.

Slither results:

BurnMessage.sol

Message.sol

MessageTransmitter.sol

Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#dead-code
Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#solidity-90
Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#solidity-80
Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#solidity-70
Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#solidity-60
Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#solidity-50
Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#solidity-20
Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#solidity-10
Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#solidity-0
Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#solidity-90
Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#solidity-80
Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#solidity-70
Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#solidity-60
Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#solidity-50
Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#solidity-20
Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#solidity-10
Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#solidity-0
Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#solidity-90
Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#solidity-80
Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#solidity-70
Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#solidity-60
Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#solidity-50
Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#solidity-20
Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#solidity-10
Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#solidity-0
Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#solidity-90
Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#solidity-80
Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#solidity-70
Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#solidity-60
Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#solidity-50
Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#solidity-20
Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#solidity-10
Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#solidity-0
Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#solidity-90
Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#solidity-80
Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#solidity-70
Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#solidity-60
Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#solidity-50
Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#solidity-20
Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#solidity-10
Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#solidity-0
Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#solidity-90
Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#solidity-80
Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#solidity-70
Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#solidity-60
Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#solidity-50
Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#solidity-20
Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#solidity-10
Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#solidity-0
Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#solidity-90
Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#solidity-80
Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#solidity-70
Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#solidity-60
Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#solidity-50
Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#solidity-20
Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#solidity-10
Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#solidity-0
Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#solidity-90
Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#solidity-80
Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#solidity-70
Reference: https://github.com/cryptic.sl
Pausable.sol

Parameter Pausable.updatepaused(address).newPause (src/roles/Pausable.sol#AB) is not in mixedCase

TokenController.sol

TokenController.getLocalToken(uint32,bytes32) (src/roles/TokenController.sol#136-166) is never used and should be removed
TokenController.setTokenController(address) (src/roles/TokenController.sol#118-126) is never used and should be removed
Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#dead-code

TokenMessenger.sol

Reentrancy in TokenMessenger.depositBurn(uint32,uint32,bytes32,address,bytes32) (src/TokenMessenger.sol#418-472):

- external calls:
  - require(bool,string).mintBurnToken.transferFrom(msg.sender,address,(localMinter).amount).Transfer operation failed (src/TokenMessenger.sol#441)
  - localMinter.burnBurnToken(amount) (src/TokenMessenger.sol#442)
  - localMinter.onMessageReceivedDomain(destinationTokenMessenger,destinationCaller,burnMessage) (src/TokenMessenger.sol#449-450)

- localMessageTransmitter.sendMessageWithCaller(destinationDomain,destinationTokenMessenger,burnMessage) (src/TokenMessenger.sol#49)

- event emitted after the call(s):
  - DepositForBurn_committed,burnToken.amount, msg.sender, localMinter.destinationDomain, destinationTokenMessenger.destinationCaller (src/TokenMessenger.sol#446-449)

Reentrancy in TokenMessenger.mintAndWithdraw(address,uint32,bytes32,address,uint32) (src/TokenMessenger.sol#527-531):

- external calls:
  - mintToken = interactTokenTransferDomain.burnToken(amount) (src/TokenMessenger.sol#529-530)

- event emitted after the call(s):
  - mintAndWithdraw(amount, mintToken) (src/TokenMessenger.sol#532)

Reentrancy in TokenMessenger.replaceDepositedBurn(bytes,bytes32,bytes) (src/TokenMessenger.sol#264-270):

- external calls:

- event emitted after the call(s):
  - DepositForBurn_originalMessage, message.sender, originalRecipient, originalRecipient.messageMessage, messageMessage.destinationDomain, originalRecipient.messageRecipient, newDestTokenMessenger.destinationCaller (src/TokenMessenger.sol#290-299)

Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities

Message.delegationCaller(bytes32) (src/messages/Message.sol#123-129) is never used and should be removed
Message.receiptRecipient(bytes32,bytes32,bytes) (src/messages/Message.sol#46-86) is never used and should be removed
Message.receiptRecipient(bytes32) (src/messages/Message.sol#142-166) is never used and should be removed
Message.sender(bytes32) (src/messages/Message.sol#113-115) is never used and should be removed
Message.version(bytes32) (src/messages/Message.sol#88-95) is never used and should be removed
Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#dead-code

Parameter Message.bytes32Address(bytes32).buf (src/messages/Message.sol#169) is not in mixedCase
TokenMinter.sol

TokenMinter.constructor(address _tokenController) (src-TokenMinter.sol#L41) shadows
- TokenController.constructor(address _tokenController) (src-TokenController.sol#L40) (state variable)

Reentrancy in TokenMinter._sendBurnForMessage(uint64, uint32, bytes32, address, bytes32) (src-TokenMinter.sol#L418-417):
- localMessage = _sendDepositForBurnMessage(_destinationDomain, _destinationTokenMessage, _destinationCaller, _burnMessage) (src-TokenMinter.sol#L441)
- localMessage = _sendDepositForBurnMessage(_destinationDomain, _destinationTokenMessage, _destinationCaller, _burnMessage) (src-TokenMinter.sol#L449)
- localMessage = _sendDepositForBurnMessage(_destinationDomain, _destinationTokenMessage, _destinationCaller, _burnMessage) (src-TokenMinter.sol#L449)

event emitted after the call(s):
- DepositForBurn(_nonReserved, _burnToken, _amount, _msg.sender, _mintRecipient, _destinationDomain, _destinationTokenMessage, _destinationCaller) (src-TokenMinter.sol#L446)

Reentrancy in TokenMinter._mintAndWithdraw(address,uint32,bytes32,address,uint256) (src-TokenMinter.sol#L517-539):
- _mintToken = _mintToken(_initialMessageDomain, _burnToken, _initialRecipient, _amount) (src-TokenMinter.sol#L525-538)

event emitted after the call(s):
- MintAndWithdraw(_recipient, _amount, _mintToken) (src-TokenMinter.sol#L532)

Reentrancy in TokenMinter.replaceDepositForBurn(bytes,bytes,bytes32) (src-TokenMinter.sol#L247-299):
- event emitted after the call(s):
- DepositForBurn(_originalMessage, _originalRecipient, _newMessageBody, _newDestinationCaller) (src-TokenMinter.sol#L272-279)

Reference: https://github.com/ArtyomS/solidity-vulnerabilities-3

Parameter Message bytes32[address(bytes32)].buf (src-TokenMinter.sol#L120) is not in titleCase
Parameter Message bytes32[address(bytes32)].buf (src-TokenMinter.sol#L148) is not in titleCase

Reference: https://github.com/ArtyomS/solidity-vulnerabilities-3
TypedMemView.sol

```
TypedMemView.leftMask(uint8) (TypedMemView.sol#167–176) uses assembly
  - INLINE ASM (TypedMemView.sol#169–175)
TypedMemView.isValid(bytes29) (TypedMemView.sol#210–217) uses assembly
  - INLINE ASM (TypedMemView.sol#213–216)
TypedMemView.castTo(bytes29,uint48) (TypedMemView.sol#270–278) uses assembly
  - INLINE ASM (TypedMemView.sol#272–277)
TypedMemView.unsafeBuildUnchecked(uint256,uint256,uint256) (TypedMemView.sol#290–297) uses assembly
  - INLINE ASM (TypedMemView.sol#291–296)
TypedMemView.build(uint256,uint256,uint256) (TypedMemView.sol#309–312) uses assembly
  - INLINE ASM (TypedMemView.sol#311–316)
TypedMemView.ref(bytes,uint48) (TypedMemView.sol#331–341) uses assembly
  - INLINE ASM (TypedMemView.sol#335–338)
TypedMemView.typeOf(bytes29) (TypedMemView.sol#348–354) uses assembly
  - INLINE ASM (TypedMemView.sol#349–353)
TypedMemView.loc(bytes29) (TypedMemView.sol#371–378) uses assembly
  - INLINE ASM (TypedMemView.sol#373–377)
TypedMemView.len(bytes29) (TypedMemView.sol#403–409) uses assembly
  - INLINE ASM (TypedMemView.sol#405–408)
TypedMemView.index(bytes29,uint256,uint8) (TypedMemView.sol#585–519) uses assembly
  - INLINE ASM (TypedMemView.sol#585–518)
TypedMemView.keccak(bytes29) (TypedMemView.sol#568–567) uses assembly
  - INLINE ASM (TypedMemView.sol#563–566)
TypedMemView.sha256(bytes29) (TypedMemView.sol#575–584) uses assembly
  - INLINE ASM (TypedMemView.sol#578–583)
TypedMemView.hash160(bytes29) (TypedMemView.sol#591–601) uses assembly
  - INLINE ASM (TypedMemView.sol#594–600)
TypedMemView.hash256(bytes29) (TypedMemView.sol#608–618) uses assembly
  - INLINE ASM (TypedMemView.sol#611–617)
TypedMemView.unsafeCopyTo(bytes29,uint256) (TypedMemView.sol#673–694) uses assembly
  - INLINE ASM (TypedMemView.sol#680–691)
TypedMemView.clone(bytes29) (TypedMemView.sol#703–717) uses assembly
  - INLINE ASM (TypedMemView.sol#706–710)
  - INLINE ASM (TypedMemView.sol#712–716)
TypedMemView.unsafeJoin(bytes291[],uint256) (TypedMemView.sol#729–746) uses assembly
  - INLINE ASM (TypedMemView.sol#730–737)
TypedMemView.joinKeccak(bytes29[]) (TypedMemView.sol#753–766) uses assembly
  - INLINE ASM (TypedMemView.sol#755–758)
TypedMemView.joinSha256(bytes29[]) (TypedMemView.sol#767–774) uses assembly
  - INLINE ASM (TypedMemView.sol#769–772)
TypedMemView.join(bytes29[]) (TypedMemView.sol#781–800) uses assembly
  - INLINE ASM (TypedMemView.sol#783–786)
  - INLINE ASM (TypedMemView.sol#792–799)
```

Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#assembly-usage

Different versions of Solidity are used:
- Version used: ['>=0.5.10', '>=0.5.10<0.8.0']
  - >=0.5.10 (SafeMath.sol#2)
  - >=0.5.10<0.8.0 (TypedMemView.sol#2)

Reference: https://github.com/cryptic/slither/wiki/Detector-Documentation#different-pragmas-directives-are-used

SafeMath.add(uint256,uint256) (SafeMath.sol#73–77) is never used and should be removed
SafeMath.div(uint256,uint256) (SafeMath.sol#65–68) is never used and should be removed
SafeMath.mul(uint256,uint256) (SafeMath.sol#39–48) is never used and should be removed
SafeMath.sub(uint256,uint256) (SafeMath.sol#46–48) is never used and should be removed

Pragma version>=0.5.10 (SafeMath.sol#2) allows old versions
Pragma version>=0.5.10<0.8.0 (TypedMemView.sol#2) is too complex

Parameter SafeMath.mul(uint256,uint256)._a (SafeMath.sol#39) is not in mixedCase
Parameter SafeMath.mul(uint256,uint256)._b (SafeMath.sol#39) is not in mixedCase
Parameter SafeMath.div(uint256,uint256)._a (SafeMath.sol#55) is not in mixedCase
Parameter SafeMath.div(uint256,uint256)._b (SafeMath.sol#55) is not in mixedCase
Parameter SafeMath.sub(uint256,uint256)._a (SafeMath.sol#55) is not in mixedCase
Parameter SafeMath.sub(uint256,uint256)._b (SafeMath.sol#55) is not in mixedCase
Parameter SafeMath.add(uint256,uint256)._a (SafeMath.sol#73) is not in mixedCase
Parameter SafeMath.add(uint256,uint256)._b (SafeMath.sol#73) is not in mixedCase
• Majority of identified issues are related to third-party libraries.
• Reentrancy issues are false positives.
• Several informational issues related to solidity naming convention were identified.
• Attestable.sol, Ownable.sol, Rescuable.sol yielded no result.
• No major issues were found by Slither.
5.2 AUTOMATED SECURITY SCAN

Description:

Halborn used automated security scanners to assist with detection of well-known security issues, and to identify low-hanging fruits on the targets for this engagement. Among the tools used was MythX, a security analysis service for Ethereum smart contracts. MythX performed a scan on all the contracts and sent the compiled results to the analyzers to locate any vulnerabilities.

MythX results:

Attestable.sol

<table>
<thead>
<tr>
<th>Line</th>
<th>SWC Title</th>
<th>Severity</th>
<th>Short Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>69</td>
<td>SWC-101</td>
<td>High</td>
<td>The arithmetic operator can overflow.</td>
</tr>
<tr>
<td>81</td>
<td>SWC-101</td>
<td>Unknown</td>
<td>Compiler-rewritable &quot;&lt;int&gt; - 1&quot; discovered</td>
</tr>
<tr>
<td>81</td>
<td>SWC-101</td>
<td>Unknown</td>
<td>Arithmetic operation &quot;-&quot; discovered</td>
</tr>
<tr>
<td>82</td>
<td>SWC-101</td>
<td>Unknown</td>
<td>Arithmetic operation &quot;-&quot; discovered</td>
</tr>
<tr>
<td>82</td>
<td>SWC-101</td>
<td>Unknown</td>
<td>Compiler-rewritable &quot;&lt;int&gt; - 1&quot; discovered</td>
</tr>
<tr>
<td>87</td>
<td>SWC-110</td>
<td>Unknown</td>
<td>Out of bounds array access</td>
</tr>
<tr>
<td>90</td>
<td>SWC-110</td>
<td>Unknown</td>
<td>Out of bounds array access</td>
</tr>
<tr>
<td>92</td>
<td>SWC-101</td>
<td>Unknown</td>
<td>Arithmetic operation &quot;-&quot; discovered</td>
</tr>
<tr>
<td>132</td>
<td>SWC-110</td>
<td>Unknown</td>
<td>Out of bounds array access</td>
</tr>
</tbody>
</table>

Attestable.sol

<table>
<thead>
<tr>
<th>Line</th>
<th>SWC Title</th>
<th>Severity</th>
<th>Short Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>SWC-100</td>
<td>Low</td>
<td>A floating pragma is set.</td>
</tr>
<tr>
<td>236</td>
<td>SWC-101</td>
<td>Unknown</td>
<td>Arithmetic operation &quot;*&quot; discovered</td>
</tr>
<tr>
<td>233</td>
<td>SWC-101</td>
<td>Unknown</td>
<td>Arithmetic operation &quot;++&quot; discovered</td>
</tr>
<tr>
<td>234</td>
<td>SWC-101</td>
<td>Unknown</td>
<td>Arithmetic operation &quot;*&quot; discovered</td>
</tr>
<tr>
<td>234</td>
<td>SWC-101</td>
<td>Unknown</td>
<td>Arithmetic operation &quot;*&quot; discovered</td>
</tr>
</tbody>
</table>
BurnMessage.sol

Report for src/messages/BurnMessage.sol
https://dashboard.mythx.io/#/console/analyses/37898046-1c42-43c1-a3aa-6644c8fd7f3e

<table>
<thead>
<tr>
<th>Line</th>
<th>SWC Title</th>
<th>Severity</th>
<th>Short Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>(SWC-103) FloatingPragma</td>
<td>Low</td>
<td>A floating pragma is set.</td>
</tr>
</tbody>
</table>

Message.sol

Report for src/messages/Message.sol
https://dashboard.mythx.io/#/console/analyses/ca3cf88f-4e96-491b-b7d5-d18449ed720a

<table>
<thead>
<tr>
<th>Line</th>
<th>SWC Title</th>
<th>Severity</th>
<th>Short Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>(SWC-103) FloatingPragma</td>
<td>Low</td>
<td>A floating pragma is set.</td>
</tr>
</tbody>
</table>

MessageTransmitter.sol

Report for lib/memview-sol/contracts/SafeMath.sol
https://dashboard.mythx.io/#/console/analyses/575381a3-7a04-4094-b080-d84ee4f55a56

<table>
<thead>
<tr>
<th>Line</th>
<th>SWC Title</th>
<th>Severity</th>
<th>Short Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>47</td>
<td>(SWC-101) IntegerOverflowAndUnderflow</td>
<td>Unknown</td>
<td>Arithmetic operation &quot;*&quot; discovered</td>
</tr>
<tr>
<td>48</td>
<td>(SWC-101) IntegerOverflowAndUnderflow</td>
<td>Unknown</td>
<td>Arithmetic operation &quot;/&quot; discovered</td>
</tr>
<tr>
<td>59</td>
<td>(SWC-101) IntegerOverflowAndUnderflow</td>
<td>Unknown</td>
<td>Arithmetic operation &quot;/&quot; discovered</td>
</tr>
<tr>
<td>67</td>
<td>(SWC-101) IntegerOverflowAndUnderflow</td>
<td>Unknown</td>
<td>Arithmetic operation &quot;&gt;&quot; discovered</td>
</tr>
<tr>
<td>74</td>
<td>(SWC-101) IntegerOverflowAndUnderflow</td>
<td>Unknown</td>
<td>Arithmetic operation &quot;+&quot; discovered</td>
</tr>
</tbody>
</table>
### Report for lib/memview-sol/contracts/TypedMemView.sol

<table>
<thead>
<tr>
<th>Line</th>
<th>SWC Title</th>
<th>Severity</th>
<th>Short Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>119</td>
<td>(SWC-101) Integer Overflow and Underflow</td>
<td>Unknown</td>
<td>Arithmetic operation &quot;-=&quot; discovered</td>
</tr>
<tr>
<td>120</td>
<td>(SWC-101) Integer Overflow and Underflow</td>
<td>Unknown</td>
<td>Arithmetic operation &quot;+&quot; discovered</td>
</tr>
<tr>
<td>128</td>
<td>(SWC-101) Integer Overflow and Underflow</td>
<td>Unknown</td>
<td>Arithmetic operation &quot;-=&quot; discovered</td>
</tr>
<tr>
<td>129</td>
<td>(SWC-101) Integer Overflow and Underflow</td>
<td>Unknown</td>
<td>Arithmetic operation &quot;+&quot; discovered</td>
</tr>
<tr>
<td>363</td>
<td>(SWC-101) Integer Overflow and Underflow</td>
<td>Unknown</td>
<td>Arithmetic operation &quot;+&quot; discovered</td>
</tr>
<tr>
<td>386</td>
<td>(SWC-101) Integer Overflow and Underflow</td>
<td>Unknown</td>
<td>Arithmetic operation &quot;+&quot; discovered</td>
</tr>
<tr>
<td>395</td>
<td>(SWC-101) Integer Overflow and Underflow</td>
<td>Unknown</td>
<td>Arithmetic operation &quot;-=&quot; discovered</td>
</tr>
<tr>
<td>417</td>
<td>(SWC-101) Integer Overflow and Underflow</td>
<td>Unknown</td>
<td>Arithmetic operation &quot;+&quot; discovered</td>
</tr>
<tr>
<td>512</td>
<td>(SWC-101) Integer Overflow and Underflow</td>
<td>Unknown</td>
<td>Arithmetic operation &quot;+&quot; discovered</td>
</tr>
<tr>
<td>530</td>
<td>(SWC-101) Integer Overflow and Underflow</td>
<td>Unknown</td>
<td>Arithmetic operation &quot;+&quot; discovered</td>
</tr>
<tr>
<td>711</td>
<td>(SWC-101) Integer Overflow and Underflow</td>
<td>Unknown</td>
<td>Arithmetic operation &quot;+&quot; discovered</td>
</tr>
<tr>
<td>740</td>
<td>(SWC-101) Integer Overflow and Underflow</td>
<td>Unknown</td>
<td>Arithmetic operation &quot;*&quot; discovered</td>
</tr>
<tr>
<td>741</td>
<td>(SWC-119) Assert Violation</td>
<td>Unknown</td>
<td>Out of bounds array access</td>
</tr>
<tr>
<td>742</td>
<td>(SWC-101) Integer Overflow and Underflow</td>
<td>Unknown</td>
<td>Arithmetic operation &quot;+&quot; discovered</td>
</tr>
<tr>
<td>743</td>
<td>(SWC-101) Integer Overflow and Underflow</td>
<td>Unknown</td>
<td>Arithmetic operation &quot;+&quot; discovered</td>
</tr>
<tr>
<td>788</td>
<td>(SWC-101) Integer Overflow and Underflow</td>
<td>Unknown</td>
<td>Arithmetic operation &quot;+&quot; discovered</td>
</tr>
</tbody>
</table>

### Report for lib/openzeppelin-contracts/contracts/math/SafeMath.sol

<table>
<thead>
<tr>
<th>Line</th>
<th>SWC Title</th>
<th>Severity</th>
<th>Short Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>(SWC-101) Integer Overflow and Underflow</td>
<td>Unknown</td>
<td>Arithmetic operation &quot;+&quot; discovered</td>
</tr>
<tr>
<td>37</td>
<td>(SWC-101) Integer Overflow and Underflow</td>
<td>Unknown</td>
<td>Arithmetic operation &quot;+&quot; discovered</td>
</tr>
<tr>
<td>50</td>
<td>(SWC-101) Integer Overflow and Underflow</td>
<td>Unknown</td>
<td>Arithmetic operation &quot;+&quot; discovered</td>
</tr>
<tr>
<td>51</td>
<td>(SWC-101) Integer Overflow and Underflow</td>
<td>Unknown</td>
<td>Arithmetic operation &quot;+&quot; discovered</td>
</tr>
<tr>
<td>62</td>
<td>(SWC-101) Integer Overflow and Underflow</td>
<td>Unknown</td>
<td>Arithmetic operation &quot;+&quot; discovered</td>
</tr>
<tr>
<td>72</td>
<td>(SWC-101) Integer Overflow and Underflow</td>
<td>Unknown</td>
<td>Arithmetic operation &quot;+&quot; discovered</td>
</tr>
<tr>
<td>86</td>
<td>(SWC-101) Integer Overflow and Underflow</td>
<td>Unknown</td>
<td>Arithmetic operation &quot;+&quot; discovered</td>
</tr>
<tr>
<td>183</td>
<td>(SWC-101) Integer Overflow and Underflow</td>
<td>Unknown</td>
<td>Arithmetic operation &quot;+&quot; discovered</td>
</tr>
<tr>
<td>118</td>
<td>(SWC-101) Integer Overflow and Underflow</td>
<td>Unknown</td>
<td>Arithmetic operation &quot;+&quot; discovered</td>
</tr>
<tr>
<td>119</td>
<td>(SWC-101) Integer Overflow and Underflow</td>
<td>Unknown</td>
<td>Arithmetic operation &quot;+&quot; discovered</td>
</tr>
<tr>
<td>137</td>
<td>(SWC-101) Integer Overflow and Underflow</td>
<td>Unknown</td>
<td>Arithmetic operation &quot;+&quot; discovered</td>
</tr>
<tr>
<td>154</td>
<td>(SWC-101) Integer Overflow and Underflow</td>
<td>Unknown</td>
<td>Arithmetic operation &quot;+&quot; discovered</td>
</tr>
<tr>
<td>172</td>
<td>(SWC-101) Integer Overflow and Underflow</td>
<td>Unknown</td>
<td>Arithmetic operation &quot;+&quot; discovered</td>
</tr>
<tr>
<td>192</td>
<td>(SWC-101) Integer Overflow and Underflow</td>
<td>Unknown</td>
<td>Arithmetic operation &quot;+&quot; discovered</td>
</tr>
<tr>
<td>212</td>
<td>(SWC-101) Integer Overflow and Underflow</td>
<td>Unknown</td>
<td>Arithmetic operation &quot;+&quot; discovered</td>
</tr>
</tbody>
</table>
### Rescuable.sol

Report for lib/openzeppelin-contracts/contracts/utils/EnumerableSet.sol
https://dashboard.mythx.io/#/console/analyses/b75381a3-7a84-4e94-b08d-d84ee4b65a56

<table>
<thead>
<tr>
<th>Line</th>
<th>SWC Title</th>
<th>Severity</th>
<th>Short Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>81</td>
<td>(SWC-181) Integer Overflow and Underflow</td>
<td>Unknown</td>
<td>Compiler-rewritable &quot;&lt;uint&gt; - 1&quot; discovered</td>
</tr>
<tr>
<td>81</td>
<td>(SWC-181) Integer Overflow and Underflow</td>
<td>Unknown</td>
<td>Arithmetic operation &quot;-&quot; discovered</td>
</tr>
<tr>
<td>82</td>
<td>(SWC-181) Integer Overflow and Underflow</td>
<td>Unknown</td>
<td>Arithmetic operation &quot;-&quot; discovered</td>
</tr>
<tr>
<td>82</td>
<td>(SWC-181) Integer Overflow and Underflow</td>
<td>Unknown</td>
<td>Compiler-rewritable &quot;&lt;uint&gt; - 1&quot; discovered</td>
</tr>
<tr>
<td>87</td>
<td>(SWC-110) Assert Violation</td>
<td>Unknown</td>
<td>Out of bounds array access</td>
</tr>
<tr>
<td>90</td>
<td>(SWC-110) Assert Violation</td>
<td>Unknown</td>
<td>Out of bounds array access</td>
</tr>
<tr>
<td>92</td>
<td>(SWC-181) Integer Overflow and Underflow</td>
<td>Unknown</td>
<td>Arithmetic operation &quot;+&quot; discovered</td>
</tr>
<tr>
<td>132</td>
<td>(SWC-110) Assert Violation</td>
<td>Unknown</td>
<td>Out of bounds array access</td>
</tr>
</tbody>
</table>

Report for src/MessageTransmitter.sol
https://dashboard.mythx.io/#/console/analyses/b75381a3-7a84-4e94-b08d-d84ee4b65a56

<table>
<thead>
<tr>
<th>Line</th>
<th>SWC Title</th>
<th>Severity</th>
<th>Short Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>(SWC-183) Floating Pragma</td>
<td>Low</td>
<td>A floating pragma is set.</td>
</tr>
<tr>
<td>387</td>
<td>(SWC-181) Integer Overflow and Underflow</td>
<td>Unknown</td>
<td>Arithmetic operation &quot;+&quot; discovered</td>
</tr>
</tbody>
</table>

Report for src/messages/Message.sol
https://dashboard.mythx.io/#/console/analyses/b75381a3-7a84-4e94-b08d-d84ee4b65a56

<table>
<thead>
<tr>
<th>Line</th>
<th>SWC Title</th>
<th>Severity</th>
<th>Short Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>136</td>
<td>(SWC-181) Integer Overflow and Underflow</td>
<td>Unknown</td>
<td>Arithmetic operation &quot;-&quot; discovered</td>
</tr>
</tbody>
</table>

Report for src/roles/Attestable.sol
https://dashboard.mythx.io/#/console/analyses/b75381a3-7a84-4e94-b08d-d84ee4b65a56

<table>
<thead>
<tr>
<th>Line</th>
<th>SWC Title</th>
<th>Severity</th>
<th>Short Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>226</td>
<td>(SWC-181) Integer Overflow and Underflow</td>
<td>Unknown</td>
<td>Arithmetic operation &quot;+&quot; discovered</td>
</tr>
<tr>
<td>233</td>
<td>(SWC-181) Integer Overflow and Underflow</td>
<td>Unknown</td>
<td>Arithmetic operation &quot;++&quot; discovered</td>
</tr>
<tr>
<td>234</td>
<td>(SWC-181) Integer Overflow and Underflow</td>
<td>Unknown</td>
<td>Arithmetic operation &quot;+&quot; discovered</td>
</tr>
<tr>
<td>234</td>
<td>(SWC-181) Integer Overflow and Underflow</td>
<td>Unknown</td>
<td>Arithmetic operation &quot;+&quot; discovered</td>
</tr>
</tbody>
</table>

### Rescuable.sol

Report for lib/openzeppelin-contracts/contracts/utils/Address.sol
https://dashboard.mythx.io/#/console/analyses/26f83c04-acb3-4736-b8b6-c6fb383468dd

<table>
<thead>
<tr>
<th>Line</th>
<th>SWC Title</th>
<th>Severity</th>
<th>Short Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>119</td>
<td>(SWC-123) Requirement Violation</td>
<td>Low</td>
<td>Requirement violation.</td>
</tr>
</tbody>
</table>

Report for src/roles/Rescuable.sol
https://dashboard.mythx.io/#/console/analyses/26f83c04-acb3-4736-b8b6-c6fb383468dd

<table>
<thead>
<tr>
<th>Line</th>
<th>SWC Title</th>
<th>Severity</th>
<th>Short Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>37</td>
<td>(SWC-123) Requirement Violation</td>
<td>Low</td>
<td>Requirement violation.</td>
</tr>
</tbody>
</table>
TokenController.sol

Report for src/roles/TokenController.sol
https://dashboard.mythx.io/#/console/analyses/a0ee6bfc-5455-4980-bfdd-9d3a610ba6be

<table>
<thead>
<tr>
<th>Line</th>
<th>SWC Title</th>
<th>Severity</th>
<th>Short Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>(SWC-103) Floating Pragma</td>
<td>Low</td>
<td>A floating pragma is set.</td>
</tr>
</tbody>
</table>

TokenMessenger.sol

Report for lib/openzeppelin-contracts/contracts/utils/Address.sol
https://dashboard.mythx.io/#/console/analyses/b2a5641f-343f-41d3-8b95-95aa7bf6c997

<table>
<thead>
<tr>
<th>Line</th>
<th>SWC Title</th>
<th>Severity</th>
<th>Short Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>119</td>
<td>(SWC-123) Requirement Violation</td>
<td>Low</td>
<td>Requirement violation.</td>
</tr>
</tbody>
</table>

Report for src/TokenMessenger.sol
https://dashboard.mythx.io/#/console/analyses/b2a5641f-343f-41d3-8b95-95aa7bf6c997

<table>
<thead>
<tr>
<th>Line</th>
<th>SWC Title</th>
<th>Severity</th>
<th>Short Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>(SWC-103) Floating Pragma</td>
<td>Low</td>
<td>A floating pragma is set.</td>
</tr>
<tr>
<td>30</td>
<td>(SWC-123) Requirement Violation</td>
<td>Low</td>
<td>Requirement violation.</td>
</tr>
</tbody>
</table>

TokenMinter.sol

Report for lib/openzeppelin-contracts/contracts/utils/Address.sol
https://dashboard.mythx.io/#/console/analyses/ac7a6b98-e1f3-45bd-9ee6-4d45de00f3a2

<table>
<thead>
<tr>
<th>Line</th>
<th>SWC Title</th>
<th>Severity</th>
<th>Short Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>119</td>
<td>(SWC-123) Requirement Violation</td>
<td>Low</td>
<td>Requirement violation.</td>
</tr>
</tbody>
</table>

Report for src/TokenMessenger.sol
https://dashboard.mythx.io/#/console/analyses/ac7a6b98-e1f3-45bd-9ee6-4d45de00f3a2

<table>
<thead>
<tr>
<th>Line</th>
<th>SWC Title</th>
<th>Severity</th>
<th>Short Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>(SWC-123) Requirement Violation</td>
<td>Low</td>
<td>Requirement violation.</td>
</tr>
</tbody>
</table>

Report for src/TokenMinter.sol
https://dashboard.mythx.io/#/console/analyses/ac7a6b98-e1f3-45bd-9ee6-4d45de00f3a2

<table>
<thead>
<tr>
<th>Line</th>
<th>SWC Title</th>
<th>Severity</th>
<th>Short Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>(SWC-103) Floating Pragma</td>
<td>Low</td>
<td>A floating pragma is set.</td>
</tr>
</tbody>
</table>
TypedMemView.sol

Report for TypedMemView.sol
https://dashboard.mythx.io/#/console/analyses/e2618948-159f-4c97-9fcc-8cc15e244c2f

<table>
<thead>
<tr>
<th>Line</th>
<th>SWC Title</th>
<th>Severity</th>
<th>Short Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>(SWC-103) Floating Pragma</td>
<td>Low</td>
<td>A floating pragma is set.</td>
</tr>
</tbody>
</table>

- Majority of identified issues are related to third-party libraries.
- Pausable.sol, Ownable.sol yielded no result.
- No major issues were discovered by Mythx software.
THANK YOU FOR CHOOSING

HALBORN