HALBORN

Circle Internet Financial - EVM Bridge

Smart Contract Security Audit

Prepared by: **Halborn**

Date of Engagement: **October 10th, 2022 - November 4th, 2022**

Visit: **Halborn.com**

DOCUMENT REVISION HISTORY 4
CONTACTS 5
1 EXECUTIVE OVERVIEW 6
1.1 INTRODUCTION 7
1.2 AUDIT SUMMARY 7
1.3 TEST APPROACH & METHODOLOGY 7
RISK METHODOLOGY 8
1.4 SCOPE 10
2 ASSESSMENT SUMMARY & FINDINGS OVERVIEW 11
3 FINDINGS & TECH DETAILS 12
3.1 (HAL-01) INCOMPATIBILITY WITH NON-STANDARD ERC20 TOKENS - MEDIUM 14
Description 14
Code Location 14
Risk Level 16
Recommendation 16
Remediation Plan 16
3.2 (HAL-02) LACK OF INPUT VALIDATION IN REPLACEDEPOSITFORBURN MAY RESULT IN TOKEN LOSS - MEDIUM 17
Description 17
Proof of Concept 20
Recommendation 20
Remediation Plan 21
3.3 (HAL-03) LACK OF TRANSFEROWNERSHIP PATTERN - LOW 22
Description 22

Risk Level 23

Recommendation 23

Remediation Plan 23

3.4 (HAL-04) REMOVEREMOTETOKENMESSENGER EMITS EVENT BASING ON USER INPUT - INFORMATIONAL 24

Description 24

Recommendation 25

Remediation Plan 25

3.5 (HAL-05) UPDATEATTESTERMANAGER EMITS EVENT WITH INCORRECT DATA - INFORMATIONAL 26

Description 26

Recommendation 27

Remediation Plan 27

3.6 (HAL-06) GAS OVER-CONSUMPTION IN LOOPS - INFORMATIONAL 28

Description 28

Code Location 28

Proof of Concept 28

Risk Level 29

Recommendation 29

Remediation Plan 29

3.7 (HAL-07) UNNEEDED INITIALIZATION OF UINT256 VARIABLES TO 0 - INFORMATIONAL 30

Description 30

Code Location 30

Risk Level 30

Recommendation 30

Remediation Plan 30

4 MANUAL TESTING 31

5 AUTOMATED TESTING 37

5.1 STATIC ANALYSIS REPORT 38

Description 38

Slither results 38

5.2 AUTOMATED SECURITY SCAN 43

Description 43

MythX results 43

DOCUMENT REVISION HISTORY

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

CONTACTS

CONTACT	COMPANY	EMAIL
Rob Behnke	Halborn	Rob.Behnke@halborn.com
Steven Walbroehl	Halborn	Steven.Walbroehl@halborn.com
Gabi Urrutia	Halborn	Gabi.Urrutia@halborn.com
Grzegorz Trawinski	Halborn	Grzegorz.Trawinski@halborn.com
Kubilay Onur Gungor	Halborn	Kubilay.Gungor@halborn.com
Roberto Reigada	Halborn	Roberto.Reigada@halborn.com
Piotr Cielas	Halborn	Piotr.Cielas@halborn.com

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.

CRITICAL	HIGH	MEDIUM	LOW	INFORMATIONAL

10 - CRITICAL
9 - 8 - HIGH
7 - 6 - MEDIUM
5 - 4 - LOW
3 - 1 - VERY LOW AND INFORMATIONAL

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: 3071bb11a8f87dfaa65846f3f12bba2ddf16add8

OUT-OF-SCOPE:
Other smart contracts in the repository, external libraries and economical attacks.

2. ASSESSMENT SUMMARY & FINDINGS OVERVIEW

CRITICAL	HIGH	MEDIUM	LOW	INFORMATIONAL
0	0	2	1	4

LIKELIHOOD

IMPACT

(HAL-01) (HAL-02)				
(HAL-04) (HAL-05)	(HAL-03)			
(HAL-06) (HAL-07)				

EXECUTIVE OVERVIEW

SECURITY ANALYSIS	RISK LEVEL	REMEDIATION DATE
HAL-01 - INCOMPATIBILITY WITH NON-STANDARD ERC20 TOKENS	Medium	RISK ACCEPTED
HAL-02 - LACK OF INPUT VALIDATION IN REPLACEDEPOSITFORBURN MAY RESULT IN TOKEN LOSS	Medium	SOLVED - 12/05/2022
HAL-03 - LACK OF TRANSFER-OWNERSHIP PATTERN	Low	SOLVED - 12/05/2022
HAL-04 - REMOVEREMOTETOKENMESSENGER EMITS EVENT BASING ON USER INPUT	Informational	SOLVED - 12/05/2022
HAL-05 - UPDATEATTESTERMANAGER EMITS EVENT WITH INCORRECT DATA	Informational	SOLVED - 12/05/2022
HAL-06 - GAS OVER-CONSUMPTION IN LOOPS	Informational	SOLVED - 12/05/2022
HAL-07 - UNNEEDED INITIALIZATION OF UINT256 VARIABLES TO 0	Informational	SOLVED - 12/05/2022

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:

Listing 1: TokenMessenger.sol (Lines 434-441)

```
418     function _depositForBurn(
419         uint256 _amount,
420         uint32 _destinationDomain,
421         bytes32 _mintRecipient,
422         address _burnToken,
423         bytes32 _destinationCaller
424     ) internal returns (uint64 nonce) {
425         require(_amount > 0, "Amount must be nonzero");
426         require(_mintRecipient != bytes32(0), "Mint recipient must
 ↳  be nonzero");
427 
428         bytes32 _destinationTokenMessenger =
 ↳ _getRemoteTokenMessenger(
429             _destinationDomain
430         );
431 
```

```
432         ITokenMinter _localMinter = _getLocalMinter();
433         IMintBurnToken _mintBurnToken = IMintBurnToken(_burnToken)
 ↳ ;
434         require(
435             _mintBurnToken.transferFrom(
436                 msg.sender,
437                 address(_localMinter),
438                 _amount
439             ),
440             "Transfer operation failed"
441         );
442         _localMinter.burn(_burnToken, _amount);
443 
444         // Format message body
445         bytes memory _burnMessage = BurnMessage._formatMessage(
446             messageBodyVersion,
447             Message.addressToBytes32(_burnToken),
448             _mintRecipient,
449             _amount,
450             Message.addressToBytes32(msg.sender)
451         );
452 
453         uint64 _nonceReserved = _sendDepositForBurnMessage(
454             _destinationDomain,
455             _destinationTokenMessenger,
456             _destinationCaller,
457             _burnMessage
458         );
459 
460         emit DepositForBurn(
461             _nonceReserved,
462             _burnToken,
463             _amount,
464             msg.sender,
465             _mintRecipient,
466             _destinationDomain,
467             _destinationTokenMessenger,
468             _destinationCaller
469         );
470 
471         return _nonceReserved;
472     }
```

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)

```
247 function replaceDepositForBurn(
248         bytes memory originalMessage,
249         bytes calldata originalAttestation,
250         bytes32 newDestinationCaller,
251         bytes32 newMintRecipient
252     ) external {
253         bytes29 _originalMsg = originalMessage.ref(0);
254         bytes29 _originalMsgBody = _originalMsg._messageBody();
255         bytes32 _originalMsgSender = _originalMsgBody.
 ↳ _getMessageSender();
256         // _originalMsgSender must match msg.sender of original
 ↳ message
257         require(
258             msg.sender == Message.bytes32ToAddress(
 ↳ _originalMsgSender),
259             "Invalid sender for message"
260         );
261
```

```
262         bytes32 _burnToken = _originalMsgBody._getBurnToken();
263         uint256 _amount = _originalMsgBody._getAmount();
264
265         bytes memory _newMessageBody = BurnMessage._formatMessage(
266             messageBodyVersion,
267             _burnToken,
268             newMintRecipient,
269             _amount,
270             _originalMsgSender
271         );
272
273         localMessageTransmitter.replaceMessage(
274             originalMessage,
275             originalAttestation,
276             _newMessageBody,
277             newDestinationCaller
278         );
279
280         emit DepositForBurn(
281             _originalMsg._nonce(),
282             Message.bytes32ToAddress(_burnToken),
283             _amount,
284             msg.sender,
285             newMintRecipient,
286             _originalMsg._destinationDomain(),
287             _originalMsg._recipient(),
288             newDestinationCaller
289         );
290     }
```

Listing 3: TokenMessenger.sol (Line 426)

```
418 function _depositForBurn(
419         uint256 _amount,
420         uint32 _destinationDomain,
421         bytes32 _mintRecipient,
422         address _burnToken,
423         bytes32 _destinationCaller
424     ) internal returns (uint64 nonce) {
425         require(_amount > 0, "Amount must be nonzero");
426         require(_mintRecipient != bytes32(0), "Mint recipient must
 ↳ be nonzero");
427
```

```
428         bytes32 _destinationTokenMessenger =
 ↳ _getRemoteTokenMessenger(
429             _destinationDomain
430         );
431
432         ITokenMinter _localMinter = _getLocalMinter();
433         IMintBurnToken _mintBurnToken = IMintBurnToken(_burnToken)
 ↳ ;
434         require(
435             _mintBurnToken.transferFrom(
436                 msg.sender,
437                 address(_localMinter),
438                 _amount
439             ),
440             "Transfer operation failed"
441         );
442         _localMinter.burn(_burnToken, _amount);
443
444         // Format message body
445         bytes memory _burnMessage = BurnMessage._formatMessage(
446             messageBodyVersion,
447             Message.addressToBytes32(_burnToken),
448             _mintRecipient,
449             _amount,
450             Message.addressToBytes32(msg.sender)
451         );
452
453         uint64 _nonceReserved = _sendDepositForBurnMessage(
454             _destinationDomain,
455             _destinationTokenMessenger,
456             _destinationCaller,
457             _burnMessage
458         );
459
460         emit DepositForBurn(
461             _nonceReserved,
462             _burnToken,
463             _amount,
464             msg.sender,
465             _mintRecipient,
466             _destinationDomain,
467             _destinationTokenMessenger,
468             _destinationCaller
469         );
```

```
470
471         return _nonceReserved;
472     }
```

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.

```
tx = sourceTokenMessenger.replaceDepositForBurn(originalMessage, signed_message.signature, destinationUser7Bytes, ZERO_ADDRESS, {'from': sourceUser4})
Transaction sent: 0x07f70546935227b36b0852266cf73d568885635260a6d38693f8f9c46119a679
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 92
  TokenMessenger.replaceDepositForBurn confirmed   Block: 417   Gas used: 62999 (0.52%)

destinationMessageTransmitter.receiveMessage(messageToSign, signed_message.signature, {'from': destinationUser7})
Transaction sent: 0x5ddeeca0c49237de492f6301d1bafafd1820fd7f4a60bf89d2657d08fd869915
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 13
  MessageTransmitter.receiveMessage confirmed   Block: 418   Gas used: 122256 (1.02%)

sourceMockMintBurnToken.balanceOf(sourceUser4) 980,000,000,000,000,000
destinationMockMintBurnToken.balanceOf(destinationUser6) 1,010,000,000,000,000,000
destinationMockMintBurnToken.balanceOf(destinationUser7) 1,000,000,000,000,000,000
sourceUser4.address 0x46C0a5326E643E4f71D3149d50B48216e174Ae84
destinationUser6.address 0x844ec86426F076647A5362706a04570A5965473B
destinationUser7.address 0x23BB2Bb6c340D4C91cAa478EdF6593fC5c4a6d4B
```

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 TRANSFEROWNERSHIP 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

```
79     function transferOwnership(address newOwner) external
↳ onlyOwner {
80         require(
81             newOwner != address(0),
82             "Ownable: new owner is the zero address"
83         );
84         emit OwnershipTransferred(_owner, newOwner);
85         setOwner(newOwner);
86     }
```

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

```
118     function updateAttesterManager(address newAttesterManager)
119         external
120         onlyAttesterManager
121     {
122         require(
123             newAttesterManager != address(0),
124             "Invalid attester manager address"
125         );
126         _setAttesterManager(newAttesterManager);
127         emit AttesterManagerUpdated(newAttesterManager,
↳ newAttesterManager);
```

```
128     }
```

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) REMOVEREMOTETOKENMESSENGER 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

```
94      /**
95       * @notice Emitted when a remote TokenMessenger is removed
96       * @param domain remote domain
97       * @param tokenMessenger TokenMessenger on remote domain
98       */
99      event RemoteTokenMessengerRemoved(
100         uint32 indexed domain,
101         bytes32 indexed tokenMessenger
102     );
```

Listing 7: TokenMessenger.sol (Lines 364,375)

```
364     function removeRemoteTokenMessenger(uint32 domain, bytes32
 ↳ tokenMessenger)
365         external
366         onlyOwner
367     {
368         // No TokenMessenger set for given remote domain.
369         require(
370             remoteTokenMessengers[domain] != bytes32(0),
371             "No TokenMessenger set"
372         );
373
```

```
374         delete remoteTokenMessengers[domain];
375         emit RemoteTokenMessengerRemoved(domain, tokenMessenger);
376     }
```

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 f2cc3448aaa827a029825a2f47256f86615f9744: 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

```
43      /**
44       * @dev Emitted when attester manager address is updated
45       * @param previousAttesterManager representing the address of
 ↳ the previous attester manager
46       * @param newAttesterManager representing the address of the
 ↳ new attester manager
47       */
48      event AttesterManagerUpdated(
49          address indexed previousAttesterManager,
50          address indexed newAttesterManager
51      );
```

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

```
118     function updateAttesterManager(address newAttesterManager)
119         external
120         onlyAttesterManager
121     {
122         require(
123             newAttesterManager != address(0),
124             "Invalid attester manager address"
125         );
126         _setAttesterManager(newAttesterManager);
127         emit AttesterManagerUpdated(newAttesterManager,
 ↳ newAttesterManager);
128     }
```

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 `i++`. It is known that, in loops, using `++i` costs less gas per iteration than `i++`.

Code Location:

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

Proof of Concept:

For example, based in the following test contract:

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 }
```

Differences in the gas costs:

```
>>> test_contract.postiincrement(1)
Transaction sent: 0xlecede6b109b707786d3685bd71dd9f22dc389957653036ca04c4cd2e72c5e0b
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 44
  test.postiincrement confirmed   Block: 13622335   Gas used: 21620 (0.32%)

<Transaction '0xlecede6b109b707786d3685bd71dd9f22dc389957653036ca04c4cd2e72c5e0b'>
>>> test_contract.preiincrement(1)
Transaction sent: 0x205f09a4d2268de4c1a40f35bb2ec2847bf2ab8d584909b42c71a022b047614a
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 45
  test.preiincrement confirmed   Block: 13622336   Gas used: 21593 (0.32%)

<Transaction '0x205f09a4d2268de4c1a40f35bb2ec2847bf2ab8d584909b42c71a022b047614a'>
>>> test_contract.postiincrement(10)
Transaction sent: 0x98c04430526a59balcf947c114b62666a4417165947d31bf300cd6ae68328033
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 46
  test.postiincrement confirmed   Block: 13622337   Gas used: 22673 (0.34%)

<Transaction '0x98c04430526a59balcf947c114b62666a4417165947d31bf300cd6ae68328033'>
>>> test_contract.preiincrement(10)
Transaction sent: 0xf060d04714eff8482a828342414d5a20be9958c822d42860e7992aba20elde05
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 47
  test.preiincrement confirmed   Block: 13622338   Gas used: 22601 (0.34%)

<Transaction '0xf060d04714eff8482a828342414d5a20be9958c822d42860e7992aba20elde05'>
```

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 `f2cc3448aaa827a029825a2f47256f86615f9744`: 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 `uint256`, it is already initialized to 0. `uint256 i = 0` reassigns the 0 to `i` which wastes gas.

Code Location:

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

Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

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

Remediation Plan:

SOLVED: The Circle team solved this issue in commit `f2cc3448aaa827a029825a2f47256f86615f9744`: the solution now does not initialize a `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:

```
[*] Deployment
message = Message.deploy({from: owner})
Transaction sent: 0x1dc72a8bd2651ec1ca871b5f3f90c13d18158fc45b7fbd9010ffd6cc8c2379bc
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 267
  Message.constructor confirmed   Block: 463   Gas used: 105628 (0.88%)
  Message deployed at: 0x9D2B02632C58219f283C6DCcaEbdC2Fd6699a068

Calling -> sourceMessageTransmitter = MessageTransmitter.deploy(sourceDomain, ZERO_ADDRESS, maxMessageBodySize, version, {from: owner})
Transaction sent: 0x42018bfb1a403cdcebc5ed6a133bb60ae691fd6975e1af8d1188cf5456985268
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 268
  MessageTransmitter.constructor confirmed   Block: 464   Gas used: 3134427 (26.12%)
  MessageTransmitter deployed at: 0xC754E5c842b5555231339401C5b8134391FC6566

Calling -> destinationMessageTransmitter = MessageTransmitter.deploy(sourceDomain, ZERO_ADDRESS, maxMessageBodySize, version, {from: owner})
Transaction sent: 0x550932de436e42585f41aff870d9b16e21b7a4ea13c86b88d2fb9bc2ca8c996b
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 269
  MessageTransmitter.constructor confirmed   Block: 465   Gas used: 3134439 (26.12%)
  MessageTransmitter deployed at: 0x955573549C68302725da4D70a4B225aca1270193

sourceTokenMessenger = TokenMessenger.deploy(sourceMessageTransmitter, messageBodyVersion, {from: owner})
Transaction sent: 0x1db38d8f0e1b88575efbce5bcd98645b13d83b121caf2fecee6797b23b958a1b
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 270
  TokenMessenger.constructor confirmed   Block: 466   Gas used: 2240631 (18.67%)
  TokenMessenger deployed at: 0xc1970cb8d85f7aa5445220d54BF0C05C98e9B920

destinationMessageTransmitter = TokenMessenger.deploy(destinationMessageTransmitter, messageBodyVersion, {from: owner})
Transaction sent: 0xe49038f7696515d1b629b375f0ead267fdab6adef21cee2174ddad60aaa1a8ed
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 271
  TokenMessenger.constructor confirmed   Block: 467   Gas used: 2240631 (18.67%)
  TokenMessenger deployed at: 0xE68782218fC1D59b803f71DEA8756aCb2295B5B1

sourceTokenMinter = TokenMinter.deploy(AnyAddress, {'from': owner})
Transaction sent: 0x6cce5cfcda65b2c0b656426447ed1a87b00a9412bf491dcf10675c51ed3a612f
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 272
  TokenMinter.constructor confirmed   Block: 468   Gas used: 1444761 (12.04%)
  TokenMinter deployed at: 0xCc9BE8e0B7bcc2c677D64d674465cae97E34EA7C

destinationTokenMinter = TokenMinter.deploy(AnyAddress, {'from': owner})
Transaction sent: 0x390b63941d7a0d1c20fff49bca41a3da65cd1cb08e209a6f4c82f06ee5691bc
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 273
  TokenMinter.constructor confirmed   Block: 469   Gas used: 1444773 (12.04%)
  TokenMinter deployed at: 0x47A2e0266473976810B2197a745D909E1A221cc8

sourceTokenMinter.addLocalTokenMessenger(sourceTokenMessenger, {'from': owner})
Transaction sent: 0xa858e9200872ec24f9b6eec26ecd107cfd4b2653e25c652c0ee5e96d209a70b9
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 274
  TokenMinter.addLocalTokenMessenger confirmed   Block: 470   Gas used: 45465 (0.38%)

destinationTokenMinter.addLocalTokenMessenger(destinationTokenMessenger, {'from': owner})
Transaction sent: 0x28b8acd9db26cfa6fab7bd353209866c01a86c062d3adf0d52b2a1c4669025ea
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 275
  TokenMinter.addLocalTokenMessenger confirmed   Block: 471   Gas used: 45465 (0.38%)

sourceTokenMessenger.addLocalMinter(sourceTokenMinter, {'from': owner})
Transaction sent: 0x0adfc7fc965c2a046bda1ad998e93390d306fa35d68571b6ad19d5becb92ee9a
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 276
  TokenMessenger.addLocalMinter confirmed   Block: 472   Gas used: 45424 (0.38%)

destinationTokenMinter.addLocalMinter(destinationTokenMinter, {'from': owner})
Transaction sent: 0x4dad5cbe0bee13cefc38649f9851e421cf777f3b6b47b7d769ff620d60cf9705
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 277
  TokenMessenger.addLocalMinter confirmed   Block: 473   Gas used: 45424 (0.38%)

sourceMockMintBurnToken = MockMintBurnToken.deploy({'from': owner})
Transaction sent: 0xeb9a97fc79fa1c632bc50f563919329282l9e1c03b220c3077de1d264fbf3a8b
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 278
  MockMintBurnToken.constructor confirmed   Block: 474   Gas used: 547199 (4.56%)
  MockMintBurnToken deployed at: 0xEf502AB9A246E727853C12ADf93305378361fE26

destinationMockMintBurnToken = MockMintBurnToken.deploy({'from': owner})
Transaction sent: 0x040288e106eea9ccd92843f1964fc61e9888a3ebf7783ee95334b31595e933e7
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 279
  MockMintBurnToken.constructor confirmed   Block: 475   Gas used: 547199 (4.56%)
  MockMintBurnToken deployed at: 0x575F51164266D53441820c89BbB2d700Ec865109
```

MANUAL TESTING

```
sourceTokenMinter.setMaxBurnAmountPerTransaction(sourceMockMintBurnToken, burnLimitPerTransaction, {'from': sourceTokenController})
Transaction sent: 0xbab0ece1f08ed393ac0009415077566dd7d60712cbf70ba7b29ab26dcd1d5b02
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 24
  TokenMinter.setMaxBurnAmountPerTransaction confirmed   Block: 476   Gas used: 44370 (0.37%)

destinationTokenMinter.setMaxBurnAmountPerTransaction(destinationMockMintBurnToken, burnLimitPerTransaction, {'from': destinationTokenController})
Transaction sent: 0x82cbf0fbe2a6281578fcda91b06bae4cbb27c84884b35f6d84c381f73d8dfe74
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 24
  TokenMinter.setMaxBurnAmountPerTransaction confirmed   Block: 477   Gas used: 44358 (0.37%)

sourceTokenMinter.linkTokenPair(sourceMockMintBurnToken, destinationDomain, destinationMockMintBurnTokenBytes, {'from': sourceTokenController})
Transaction sent: 0x22fb9b8df7da235d1c9d064a7ad63a25f018a846e1b8cc8b99528aedcab3fd6c
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 25
  TokenMinter.linkTokenPair confirmed   Block: 478   Gas used: 47067 (0.39%)

destinationTokenMinter.linkTokenPair(destinationMockMintBurnTokenBytes, sourceDomain, sourceMockMintBurnToken, {'from': destinationTokenController})
Transaction sent: 0x1fd5ab15f99a6584953a1d286f688caf61f93f9c2c980d592a0cbc12856ddeea
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 25
  TokenMinter.linkTokenPair confirmed   Block: 479   Gas used: 47055 (0.39%)

sourceTokenMessenger.addRemoteTokenMessenger(destinationDomain, destinationTokenMessengerTokenBytes, {'from': owner})
Transaction sent: 0x3e1cf602b391c67285d399a127b5b5db2430f8fdd5072201fd907c262c3e2285
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 280
  TokenMessenger.addRemoteTokenMessenger confirmed   Block: 480   Gas used: 45206 (0.38%)

destinationTokenMessenger.addRemoteTokenMessenger(sourceDomain, sourceTokenMessengerBytes, {'from': owner})
Transaction sent: 0xa9679fdf7b1f0baae5591e0ee34575eeeab6f6dbd17acf509b0c4d36f3d2ead5
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 281
  TokenMessenger.addRemoteTokenMessenger confirmed   Block: 481   Gas used: 45194 (0.38%)

sourceLocalToken 0xEf502AB9A246E727853C12ADf93305378361fE26
destinationLocalToken 0x575F51164266D53441820c89BbB2d700Ec865109
sourceMockMintBurnToken 0xEf502AB9A246E727853C12ADf93305378361fE26
destinationMockMintBurnToken 0x575F51164266D53441820c89BbB2d700Ec865109
Transaction sent: 0x4f24a23921a69e43177689620a16505afad72b3ff99ef05c880a190968513259
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 282
  MockMintBurnToken.mint confirmed   Block: 482   Gas used: 67596 (0.56%)

Transaction sent: 0x417ef8c408223714088cb4b5f494d123ce66c90b19011ca29efa1d8a23cd230a
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 283
  MockMintBurnToken.mint confirmed   Block: 483   Gas used: 52596 (0.44%)

Transaction sent: 0x561816b36672c8ee6f9b361863baf64c8dcd362abd7aba2b9be0d6cd8b9dc94a
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 284
  MockMintBurnToken.mint confirmed   Block: 484   Gas used: 67596 (0.56%)

Transaction sent: 0x14e61b913d8ba298e0ff99fd7040bfeddfcae06cbf1cb216197b95c622f1e902
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 285
  MockMintBurnToken.mint confirmed   Block: 485   Gas used: 52596 (0.44%)

sourceMockMintBurnToken.approve(sourceTokenMessenger, burnLimitPerTransaction + 1, {'from': sourceUser4})
Transaction sent: 0xa1ccd99eee4d4c44d104e3eba04e1b0a0407c8d04c742da791b29a69c3d82860
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 104
  MockMintBurnToken.approve confirmed   Block: 486   Gas used: 44023 (0.37%)

sourceMockMintBurnToken.balanceOf(sourceUser4) 1,000,000,000,000,000,000
sourceMockMintBurnToken.balanceOf(sourceUser5) 1,000,000,000,000,000,000
destinationMockMintBurnToken.balanceOf(destinationUser6) 1,000,000,000,000,000,000
destinationMockMintBurnToken.balanceOf(destinationUser7) 1,000,000,000,000,000,000
sourceTokenMessenger.depositForBurn(burnLimitPerTransaction + 1, destinationDomain, destinationUser6Bytes, sourceMockMintBurnToken {'from': sourceUser4})
Transaction sent: 0x9aa4579c666ffb3ef64872c2f04ddcc0e62043f5095cdcfa4f9b0b7df7944e79
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 105
  TokenMessenger.depositForBurn confirmed (Burn amount exceeds per tx limit)   Block: 487   Gas used: 69835 (0.58%)

sourceTokenMessenger.depositForBurn(0, destinationDomain, destinationUser6Bytes, sourceMockMintBurnToken {'from': sourceUser4})
Transaction sent: 0x9504f246fe48346538ca89a54adfdc60c9f1a018b244f56dd6a8151e44301944
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 106
  TokenMessenger.depositForBurn confirmed (Amount must be nonzero)   Block: 488   Gas used: 22534 (0.19%)

sourceTokenMessenger.depositForBurn(burnLimitPerTransaction * 2, destinationDomain, destinationUser6Bytes, sourceMockMintBurnToken {'from': sourceUser4})
Transaction sent: 0xca4e371ad47969e558605d88595360f5aad8616ad5947a56bad6ab78db410285
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 107
  TokenMessenger.depositForBurn confirmed (ERC20: transfer amount exceeds allowance)   Block: 489   Gas used: 28216 (0.24%)

---
--- depositForBurn ---
---
sourceTokenMessenger.depositForBurn(burnLimitPerTransaction, destinationDomain, destinationUser6Bytes, sourceMockMintBurnToken {'from': sourceUser5})
Transaction sent: 0xf2d2fccc3d8724d390f92e32cc1ff8ceee8c73ac897d1a509a1a75bebfe3597a
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 27
  TokenMessenger.depositForBurn confirmed (ERC20: transfer amount exceeds allowance)   Block: 490   Gas used: 28216 (0.24%)

sourceTokenMessenger.depositForBurn(burnLimitPerTransaction, destinationDomain, destinationUser6Bytes, sourceMockMintBurnToken {'from': sourceUser4})
Transaction sent: 0x9acc6e573b13078cc917b8b06f52b79d12d9885b3b4ef9cb96b12fc8239aa7b4
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 108
  TokenMessenger.depositForBurn confirmed   Block: 491   Gas used: 103462 (0.86%)

sourceMockMintBurnToken.balanceOf(sourceUser4) 990,000,000,000,000,000
sourceMockMintBurnToken.balanceOf(sourceUser5) 1,000,000,000,000,000,000
destinationMockMintBurnToken.balanceOf(destinationUser6) 1,000,000,000,000,000,000
destinationMockMintBurnToken.balanceOf(destinationUser7) 1,000,000,000,000,000,000
sourceUser4.address 0x46C0a5326E643E4f71D3149d50B48216e174Ae84
sourceUser5.address 0x807c47A89F720fe4Ee9b8343c286Fc886f43191b
destinationUser6.address 0x844ec86426F076647A5362706a04570A5965473B
destinationUser7.address 0x23BB2Bb6c340D4C91cAa478EdF6593fC5c4a6d4B
sourceMessageTransmitter.isEnabledAttester(attester) True
destinationMessageTransmitter.isEnabledAttester(attester) True
```

MANUAL TESTING

```
---
--- receiveMessage ---
---
destinationMessageTransmitter.receiveMessage(messageToSign, signed_message.signature, {'from': owner})
Transaction sent: 0xfbd8b384191dc5a2c4fd3f34439e7b2d8ce4dd27d323453761022a7c713bb6bc
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 286
  MessageTransmitter.receiveMessage confirmed   Block: 492   Gas used: 104859 (0.87%)

sourceMockMintBurnToken.balanceOf(sourceUser4) 990,000,000,000,000,000
sourceMockMintBurnToken.balanceOf(sourceUser5) 1,000,000,000,000,000,000
destinationMockMintBurnToken.balanceOf(destinationUser6) 1,010,000,000,000,000,000
destinationMockMintBurnToken.balanceOf(destinationUser7) 1,000,000,000,000,000,000
sourceUser4.address 0x46C0a5326E643E4f71D3149d50B48216e174Ae84
sourceUser5.address 0x807c47A89F720fe4Ee9b8343c286Fc886f43191b
destinationUser6.address 0x844ec86426F076647A5362706a04570A5965473B
destinationUser7.address 0x23BB2Bb6c340D4C91cAa478EdF6593fC5c4a6d4B
--- replay receiveMessage ---
destinationMessageTransmitter.receiveMessage(messageToSign, signed_message.signature, {'from': owner})
Transaction sent: 0x29e512ca3b59ce657094dbfa37bf1ea480eea7c04040882efe57b51c2415fd8a
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 287
  MessageTransmitter.receiveMessage confirmed (Nonce already used)   Block: 493   Gas used: 39639 (0.33%)
```

```
---
--- depositForBurnWithCaller ---
---
sourceMockMintBurnToken.approve(sourceTokenMessenger, burnLimitPerTransaction, {'from': sourceUser5})
Transaction sent: 0x364d81f7f07b4a94be04c692b1e94885b9087e9c1e96fb8871bed2fa57f484b7
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 28
  MockMintBurnToken.approve confirmed   Block: 494   Gas used: 44011 (0.37%)

sourceTokenMessenger.depositForBurnWithCaller(burnLimitPerTransaction, destinationDomain, destinationUser7Bytes, sourceMockMintBurnToken, destinationUser7Bytes, {'from': sourceUser5})
Transaction sent: 0x2d1ec0b8ea802379a9f72f234a7ebade550227e2f90e4521bd6133caa650bfbe
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 29
  TokenMessenger.depositForBurnWithCaller confirmed   Block: 495   Gas used: 73996 (0.62%)
sourceMockMintBurnToken.balanceOf(sourceUser4) 990,000,000,000,000,000
sourceMockMintBurnToken.balanceOf(sourceUser5) 990,000,000,000,000,000
destinationMockMintBurnToken.balanceOf(destinationUser6) 1,010,000,000,000,000,000
destinationMockMintBurnToken.balanceOf(destinationUser7) 1,000,000,000,000,000,000
sourceUser4.address 0x46C0a5326E643E4f71D3149d50B48216e174Ae84
sourceUser5.address 0x807c47A89F720fe4Ee9b8343c286Fc886f43191b
destinationUser6.address 0x844ec86426F076647A5362706a04570A5965473B
destinationUser7.address 0x23BB2Bb6c340D4C91cAa478EdF6593fC5c4a6d4B
---
--- receiveMessage ---
---
destinationMessageTransmitter.receiveMessage(messageToSign, signed_message.signature, {'from': owner})
Transaction sent: 0xbbb8f7a8735da0910c1bc48fe73b001c20f486bce6916b885e04af490bd9cb4c
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 288
  MessageTransmitter.receiveMessage confirmed (Invalid caller for message)   Block: 496   Gas used: 39340 (0.33%)

destinationMessageTransmitter.receiveMessage(messageToSign, signed_message.signature, {'from': owner})
Transaction sent: 0x89e08a4356ef148c7f2c74e8eaa1d91ec036a269834952a24713eeaa4d0f6998
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 16
  MessageTransmitter.receiveMessage confirmed   Block: 497   Gas used: 107513 (0.90%)

sourceMockMintBurnToken.balanceOf(sourceUser4) 990,000,000,000,000,000
sourceMockMintBurnToken.balanceOf(sourceUser5) 990,000,000,000,000,000
destinationMockMintBurnToken.balanceOf(destinationUser6) 1,010,000,000,000,000,000
destinationMockMintBurnToken.balanceOf(destinationUser7) 1,010,000,000,000,000,000
sourceUser4.address 0x46C0a5326E643E4f71D3149d50B48216e174Ae84
sourceUser5.address 0x807c47A89F720fe4Ee9b8343c286Fc886f43191b
destinationUser6.address 0x844ec86426F076647A5362706a04570A5965473B
destinationUser7.address 0x23BB2Bb6c340D4C91cAa478EdF6593fC5c4a6d4B
```

```
---
--- depositForBurn (and replaceDepositForBurn later) ---
---
sourceMockMintBurnToken.approve(sourceTokenMessenger, burnLimitPerTransaction + 1, {'from': sourceUser4})
Transaction sent: 0x4090b8bce6b52c51d35c87410de72178ebd99135e788ec826e5bdafc286f85cd
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 109
  MockMintBurnToken.approve confirmed   Block: 498   Gas used: 29023 (0.24%)

sourceTokenMessenger.depositForBurn(burnLimitPerTransaction, destinationDomain, destinationUser6Bytes, sourceMockMintBurnToken {'from': sourceUser4})
Transaction sent: 0x2506053511540922158aa985746918e6d3f23a7e039f82e9b260c78018928abc9
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 110
  TokenMessenger.depositForBurn confirmed   Block: 499   Gas used: 88462 (0.74%)

sourceMockMintBurnToken.balanceOf(sourceUser4) 980,000,000,000,000,000
sourceMockMintBurnToken.balanceOf(sourceUser5) 990,000,000,000,000,000
destinationMockMintBurnToken.balanceOf(destinationUser6) 1,010,000,000,000,000,000
destinationMockMintBurnToken.balanceOf(destinationUser7) 1,010,000,000,000,000,000
sourceUser4.address 0x46C0a5326E643E4f71D3149d50B48216e174Ae84
sourceUser5.address 0x807c47A89F720fe4Ee9b8343c286Fc886f43191b
destinationUser6.address 0x844ec86426F076647A5362706a04570A5965473B
destinationUser7.address 0x23BB2Bb6c340D4C91cAa478EdF6593fC5c4a6d4B
```

MANUAL TESTING

```
tx = sourceTokenMessenger.replaceDepositForBurn(originalMessage, signed_message.signature, ZERO_ADDRESS, ZERO_ADDRESS, {'from': sourceUser5})
Transaction sent: 0x684340ea2fded167a6c0b55c4f54a55202156c3a7a15d63ec5b05dec8b43eb53
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 30
  TokenMessenger.replaceDepositForBurn confirmed (Invalid sender for message)   Block: 500   Gas used: 29935 (0.25%)

tx = sourceTokenMessenger.replaceDepositForBurn(originalMessage, signed_message.signature, ZERO_ADDRESS, ZERO_ADDRESS, {'from': sourceUser4})
Transaction sent: 0x3eac78b76a7ae03f036e82a73aec6967285481920d79651170133c709f67cc2e
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 111
  TokenMessenger.replaceDepositForBurn confirmed   Block: 501   Gas used: 62747 (0.52%)

tx = sourceTokenMessenger.replaceDepositForBurn(originalMessage, signed_message.signature, destinationUser7Bytes, ZERO_ADDRESS, {'from': sourceUser4})
Transaction sent: 0x3eb979702ab139076c95063104c67cc80353642b87928a2c31805cc3fd4762a0
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 112
  TokenMessenger.replaceDepositForBurn confirmed   Block: 502   Gas used: 62987 (0.52%)

tx = sourceTokenMessenger.replaceDepositForBurn(originalMessage, signed_message.signature, destinationUser7Bytes, destinationUser7Bytes, {'from': sourceUser4})
Transaction sent: 0xdbfedec990e08a36f5d4ec269e1290ae2a4d01d276034416f7809873b7df5fa4
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 113
  TokenMessenger.replaceDepositForBurn confirmed   Block: 503   Gas used: 63227 (0.53%)

sourceMessageTransmitter.replaceMessage(originalMessage, signed_message.signature, forgedMessageBody, destinationUser7Bytes,  {'from': sourceUser4})
Transaction sent: 0xcf9775191a8ad160e691b880efe345ae43daf32e6568598f46a1457bd4ccca33
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 114
  MessageTransmitter.replaceMessage confirmed (Sender not permitted to use nonce)   Block: 504   Gas used: 43384 (0.36%)

destinationMessageTransmitter.receiveMessage(messageToSign, signed_message.signature, {'from': owner})
Transaction sent: 0xef013679ef0375770332f7b69348116 2cb82ac9e68513735d589aea2b6c496df
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 289
  MessageTransmitter.receiveMessage confirmed (Invalid caller for message)   Block: 505   Gas used: 39340 (0.33%)

destinationMessageTransmitter.receiveMessage(messageToSign, signed_message.signature, {'from': destinationUser7})
Transaction sent: 0x3eddf7f83d3853be5df0af0f37cc41e0dde5e19bfc6f5ad775eb96eba023de7d
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 17
  MessageTransmitter.receiveMessage confirmed   Block: 506   Gas used: 107513 (0.90%)

sourceMockMintBurnToken.balanceOf(sourceUser4) 980,000,000,000,000,000
sourceMockMintBurnToken.balanceOf(sourceUser5) 990,000,000,000,000,000
destinationMockMintBurnToken.balanceOf(destinationUser6) 1,010,000,000,000,000,000
destinationMockMintBurnToken.balanceOf(destinationUser7) 1,020,000,000,000,000,000
sourceUser4.address 0x46C0a5326E643E4f71D3149d50B48216e174Ae84
sourceUser5.address 0x807c47A89F720fe4Ee9b8343c286Fc886f43191b
destinationUser6.address 0x844ec86426F076647A5362706a04570A5965473B
destinationUser7.address 0x23BB2Bb6c340D4C91cAa478EdF6593fC5c4a6d4B

---
--- 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.enableAttester(attester2, {'from': owner})
Transaction sent: 0xd1e80b8b01d76e124b4855b70a29711c93fb0e32ad4538aadfe9394d2e3bbda1
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 309
  MessageTransmitter.enableAttester confirmed   Block: 536   Gas used: 71616 (0.60%)

sourceMessageTransmitter.enableAttester(attester3, {'from': owner})
Transaction sent: 0x79bf154f81084677f8e791b02a37569d8734ec1155a1ead8600a06c1950c6f1e
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 310
  MessageTransmitter.enableAttester confirmed   Block: 537   Gas used: 71616 (0.60%)

destinationMessageTransmitter.enableAttester(attester2, {'from': owner})
Transaction sent: 0xb3b86bded58641ba8cf8372346036be7eeee115077b204a993fca714ef4e7ff4
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 311
  MessageTransmitter.enableAttester confirmed   Block: 538   Gas used: 71616 (0.60%)

destinationMessageTransmitter.enableAttester(attester3, {'from': owner})
Transaction sent: 0xe192b873ede307d9bdc1446317af9fd550375e130c48f5b9174b0c04c155b9c7
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 312
  MessageTransmitter.enableAttester confirmed   Block: 539   Gas used: 71616 (0.60%)

destinationMessageTransmitter.enableAttester(attester2, {'from': owner})
Transaction sent: 0xb606d26b8abca2b9315f06f4e6953b78e2ca41aaab7cefe5f1e1313bcaa7fba4
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 313
  MessageTransmitter.enableAttester confirmed (Attester already enabled)   Block: 540   Gas used: 23775 (0.20%)

destinationMessageTransmitter.enableAttester(attester3, {'from': owner})
Transaction sent: 0x2332f4e24f022d5a84e2220aa4a0c1fed8be1ece63596e000564b70f319563c5
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 314
  MessageTransmitter.enableAttester confirmed (Attester already enabled)   Block: 541   Gas used: 23775 (0.20%)
```

```
sourceMessageTransmitter.isEnabledAttester(attester1) True
destinationMessageTransmitter.isEnabledAttester(attester1) True
sourceMessageTransmitter.isEnabledAttester(attester2) True
destinationMessageTransmitter.isEnabledAttester(attester2) True
sourceMessageTransmitter.isEnabledAttester(attester3) True
destinationMessageTransmitter.isEnabledAttester(attester3) True
sourceMessageTransmitter.setSignatureThreshold(4, {'from': owner})
Transaction sent: 0x6234657f521e8f569cca4b86d220d1453e11c8794054335e46e3fe8730a81c3f
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 315
  MessageTransmitter.setSignatureThreshold confirmed (New signature threshold too high)   Block: 542   Gas used: 23293 (0.19%)

sourceMessageTransmitter.setSignatureThreshold(2, {'from': owner})
Transaction sent: 0x4ed11b261664358e8e9c3cff012ad5ca049286ff7611f5da703956c0471c2156
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 316
  MessageTransmitter.setSignatureThreshold confirmed   Block: 543   Gas used: 31364 (0.26%)

sourceMessageTransmitter.setSignatureThreshold(3, {'from': owner})
Transaction sent: 0x6cf4ae28659f28eb5ec36c0366ca4c97fd3414390ae2172309192163d6e465c9
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 317
  MessageTransmitter.setSignatureThreshold confirmed   Block: 544   Gas used: 31364 (0.26%)

destinationMessageTransmitter.setSignatureThreshold(3, {'from': owner})
Transaction sent: 0xf8b68adc1f0496bb32eedefddbd46e7bb4ec75e4ad71c00df68f310fc6ec8e28
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 318
  MessageTransmitter.setSignatureThreshold confirmed   Block: 545   Gas used: 31364 (0.26%)
```

```
---
--- receiveMessage ---
---
destinationMessageTransmitter.receiveMessage(messageToSign, signed_message.signature, {'from': owner})
Transaction sent: 0x397e27a20aa42bbbeb6738b5bd42109f91745012ffdc31bc0cbc9289692c30d0
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 352
  MessageTransmitter.receiveMessage confirmed (Invalid attestation length)   Block: 589   Gas used: 27715 (0.23%)

Unsorted: [<LocalAccount '0x4A5a1A07282739D8629096F4142bc1f3934a77DB'>, <LocalAccount '0x2d785Af0cf25aAEFdbfcB31220EEDDe05C9dd0de'>, <LocalAccount '0xb0F7f903A96890c457e6FA7E2485C0d0E35B3e96'>]
Sorted:   [<LocalAccount '0x2d785Af0cf25aAEFdbfcB31220EEDDe05C9dd0de'>, <LocalAccount '0x4A5a1A07282739D8629096F4142bc1f3934a77DB'>, <LocalAccount '0xb0F7f903A96890c457e6FA7E2485C0d0E35B3e96'>]
destinationMessageTransmitter.receiveMessage(messageToSign, signed_message.signature(duplicate), {'from': owner})
Transaction sent: 0xb236618cba01f44805029945c6c3a0d30a615c8928f4ba3ccdb4cdf9c2e9f59a
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 353
  MessageTransmitter.receiveMessage confirmed (Invalid signature order or dupe)   Block: 590   Gas used: 48655 (0.41%)

destinationMessageTransmitter.receiveMessage(messageToSign, signed_message.signature(wrong order), {'from': owner})
Transaction sent: 0x9131063a0925936a13123376b823ab94a08b0b373ead51ad57a23fa731d446fc
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 354
  MessageTransmitter.receiveMessage confirmed (Invalid signature order or dupe)   Block: 591   Gas used: 42011 (0.35%)

destinationMessageTransmitter.receiveMessage(messageToSign, signed_message.signature, {'from': owner})
Transaction sent: 0x062f060730a48d5bf2cf8e4b6d3ca935e8e42369ac05ede43f9c0204426b6be3
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 355
  MessageTransmitter.receiveMessage confirmed   Block: 592   Gas used: 120235 (1.00%)

sourceMockMintBurnToken.balanceOf(sourceUser4) 990,000,000,000,000,000
sourceMockMintBurnToken.balanceOf(sourceUser5) 1,000,000,000,000,000,000
destinationMockMintBurnToken.balanceOf(destinationUser6) 1,010,000,000,000,000,000
destinationMockMintBurnToken.balanceOf(destinationUser7) 1,000,000,000,000,000,000
sourceUser4.address 0x46C0a5326E643E4f71D3149d50B48216e174Ae84
sourceUser5.address 0x807c47A89F720fe4Ee9b8343c286Fc886f43191b
destinationUser6.address 0x844ec86426F076647A536270da84570A59655438
destinationUser7.address 0x23BB2Bb6c340D4C91cAa478EdF6593fC5c4a6d4B
```

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

```
BurnMessage._formatMessage(uint32,bytes32,bytes32,uint256,bytes32) (src/messages/BurnMessage.sol#56-71) is never used and should be removed
BurnMessage._getAmount(bytes29) (src/messages/BurnMessage.sol#113-115) is never used and should be removed
BurnMessage._getBurnToken(bytes29) (src/messages/BurnMessage.sol#91-93) is never used and should be removed
BurnMessage._getMessageSender(bytes29) (src/messages/BurnMessage.sol#78-84) is never used and should be removed
BurnMessage._getMintRecipient(bytes29) (src/messages/BurnMessage.sol#100-106) is never used and should be removed
BurnMessage._getVersion(bytes29) (src/messages/BurnMessage.sol#122-124) is never used and should be removed
BurnMessage._isValidBurnMessage(bytes29,uint32) (src/messages/BurnMessage.sol#132-139) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
```

Message.sol

```
Message._destinationCaller(bytes29) (src/messages/Message.sol#123-129) is never used and should be removed
Message._destinationDomain(bytes29) (src/messages/Message.sol#99-105) is never used and should be removed
Message._formatMessage(uint32,uint32,uint32,uint64,bytes32,bytes32,bytes32,bytes) (src/messages/Message.sol#65-86) is never used and should
be removed
Message._messageBody(bytes29) (src/messages/Message.sol#132-139) is never used and should be removed
Message._nonce(bytes29) (src/messages/Message.sol#108-110) is never used and should be removed
Message._recipient(bytes29) (src/messages/Message.sol#118-120) is never used and should be removed
Message._recipientAddress(bytes29) (src/messages/Message.sol#142-148) is never used and should be removed
Message._sender(bytes29) (src/messages/Message.sol#113-115) is never used and should be removed
Message._sourceDomain(bytes29) (src/messages/Message.sol#94-96) is never used and should be removed
Message._version(bytes29) (src/messages/Message.sol#89-91) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Parameter Message.bytes32ToAddress(bytes32)._buf (src/messages/Message.sol#162) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
```

MessageTransmitter.sol

```
Reentrancy in MessageTransmitter.receiveMessage(bytes,bytes) (src/MessageTransmitter.sol#247-304):
        External calls:
        - require(bool,string)(IMessageHandler(_m._recipientAddress()).handleReceiveMessage(_sourceDomain,_sender,_messageBody),handleReceiv
eMessage() failed) (src/MessageTransmitter.sol#286-293)
        Event emitted after the call(s):
        - MessageReceived(msg.sender,_sourceDomain,_nonce,_sender,_messageBody) (src/MessageTransmitter.sol#296-302)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

Parameter Message.bytes32ToAddress(bytes32)._buf (src/messages/Message.sol#162) is not in mixedCase
Parameter Pausable.updatePauser(address)._newPauser (src/roles/Pausable.sol#80) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
```

Pausable.sol

```
Parameter Pausable.updatePauser(address)._newPauser (src/roles/Pausable.sol#80) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
```

TokenController.sol

```
TokenController._getLocalToken(uint32,bytes32) (src/roles/TokenController.sol#135-146) is never used and should be removed
TokenController._setTokenController(address) (src/roles/TokenController.sol#118-125) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
```

TokenMessenger.sol

```
Reentrancy in TokenMessenger._depositForBurn(uint256,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/TokenMe
ssenger.sol#434-441)
        - _localMinter.burn(_burnToken,_amount) (src/TokenMessenger.sol#442)
        - _nonceReserved = _sendDepositForBurnMessage(_destinationDomain,_destinationTokenMessenger,_destinationCaller,_burnMessage) (src/To
kenMessenger.sol#453-458)
                - localMessageTransmitter.sendMessage(_destinationDomain,_destinationTokenMessenger,_burnMessage) (src/TokenMessenger.sol#49
2-497)
                - localMessageTransmitter.sendMessageWithCaller(_destinationDomain,_destinationTokenMessenger,_destinationCaller,_burnMessag
e) (src/TokenMessenger.sol#499-505)
        Event emitted after the call(s):
        - DepositForBurn(_nonceReserved,_burnToken,_amount,msg.sender,_mintRecipient,_destinationDomain,_destinationTokenMessenger,_destinat
ionCaller) (src/TokenMessenger.sol#460-469)
Reentrancy in TokenMessenger._mintAndWithdraw(address,uint32,bytes32,address,uint256) (src/TokenMessenger.sol#517-533):
        External calls:
        - _mintToken = _minter.mint(_remoteDomain,_burnToken,_mintRecipient,_amount) (src/TokenMessenger.sol#525-530)
        Event emitted after the call(s):
        - MintAndWithdraw(_mintRecipient,_amount,_mintToken) (src/TokenMessenger.sol#532)
Reentrancy in TokenMessenger.replaceDepositForBurn(bytes,bytes,bytes32,bytes32) (src/TokenMessenger.sol#247-290):
        External calls:
        - localMessageTransmitter.replaceMessage(originalMessage,originalAttestation,_newMessageBody,newDestinationCaller) (src/TokenMesseng
er.sol#273-278)
        Event emitted after the call(s):
        - DepositForBurn(_originalMsg._nonce(),Message.bytes32ToAddress(_burnToken),_amount,msg.sender,newMintRecipient,_originalMsg._destin
ationDomain(),_originalMsg._recipient(),newDestinationCaller) (src/TokenMessenger.sol#280-289)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

Message._destinationCaller(bytes29) (src/messages/Message.sol#123-129) is never used and should be removed
Message._formatMessage(uint32,uint32,uint32,uint64,bytes32,bytes32,bytes32,bytes) (src/messages/Message.sol#65-86) is never used and should
be removed
Message._recipientAddress(bytes29) (src/messages/Message.sol#142-148) is never used and should be removed
Message._sender(bytes29) (src/messages/Message.sol#113-115) is never used and should be removed
Message._sourceDomain(bytes29) (src/messages/Message.sol#94-96) is never used and should be removed
Message._version(bytes29) (src/messages/Message.sol#89-91) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Parameter Message.bytes32ToAddress(bytes32)._buf (src/messages/Message.sol#162) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
```

AUTOMATED TESTING

TokenMinter.sol

```
TokenMinter.constructor(address)._tokenController (src/TokenMinter.sol#61) shadows:
        - TokenController._tokenController (src/roles/TokenController.sol#73) (state variable)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

Reentrancy in TokenMessenger._depositForBurn(uint256,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#434-441)
        - _localMinter.burn(_burnToken,_amount) (src/TokenMessenger.sol#442)
        - _nonceReserved = _sendDepositForBurnMessage(_destinationDomain,_destinationTokenMessenger,_destinationCaller,_burnMessage) (src/TokenMessenger.sol#453-458)
                - localMessageTransmitter.sendMessage(_destinationDomain,_destinationTokenMessenger,_burnMessage) (src/TokenMessenger.sol#492-497)
                - localMessageTransmitter.sendMessageWithCaller(_destinationDomain,_destinationTokenMessenger,_destinationCaller,_burnMessage) (src/TokenMessenger.sol#499-505)
        Event emitted after the call(s):
        - DepositForBurn(_nonceReserved,_burnToken,_amount,msg.sender,_mintRecipient,_destinationDomain,_destinationTokenMessenger,_destinationCaller) (src/TokenMessenger.sol#460-469)
Reentrancy in TokenMessenger._mintAndWithdraw(address,uint32,bytes32,address,uint256) (src/TokenMessenger.sol#517-533):
        External calls:
        - _mintToken = _minter.mint(_remoteDomain,_burnToken,_mintRecipient,_amount) (src/TokenMessenger.sol#525-530)
        Event emitted after the call(s):
        - MintAndWithdraw(_mintRecipient,_amount,_mintToken) (src/TokenMessenger.sol#532)
Reentrancy in TokenMessenger.replaceDepositForBurn(bytes,bytes,bytes32,bytes32) (src/TokenMessenger.sol#247-290):
        External calls:
        - localMessageTransmitter.replaceMessage(originalMessage,originalAttestation,_newMessageBody,newDestinationCaller) (src/TokenMessenger.sol#273-278)
        Event emitted after the call(s):
        - DepositForBurn(_originalMsg._nonce(),Message.bytes32ToAddress(_burnToken),_amount,msg.sender,newMintRecipient,_originalMsg._destinationDomain(),_originalMsg._recipient(),newDestinationCaller) (src/TokenMessenger.sol#280-289)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

Message._destinationCaller(bytes29) (src/messages/Message.sol#123-129) is never used and should be removed
Message._formatMessage(uint32,uint32,uint32,uint64,bytes32,bytes32,bytes32,bytes) (src/messages/Message.sol#65-86) is never used and should be removed
Message._recipientAddress(bytes29) (src/messages/Message.sol#142-148) is never used and should be removed
Message._sender(bytes29) (src/messages/Message.sol#113-115) is never used and should be removed
Message._sourceDomain(bytes29) (src/messages/Message.sol#94-96) is never used and should be removed
Message._version(bytes29) (src/messages/Message.sol#89-91) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Parameter Message.bytes32ToAddress(bytes32)._buf (src/messages/Message.sol#162) is not in mixedCase
Parameter Pausable.updatePauser(address)._newPauser (src/roles/Pausable.sol#80) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
```

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,uint40) (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#309-321) uses assembly
        - INLINE ASM (TypedMemView.sol#311-316)
TypedMemView.ref(bytes,uint40) (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#505-519) uses assembly
        - INLINE ASM (TypedMemView.sol#515-518)
TypedMemView.keccak(bytes29) (TypedMemView.sol#560-567) uses assembly
        - INLINE ASM (TypedMemView.sol#563-566)
TypedMemView.sha2(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(bytes29[],uint256) (TypedMemView.sol#729-746) uses assembly
        - INLINE ASM (TypedMemView.sol#730-737)
TypedMemView.joinKeccak(bytes29[]) (TypedMemView.sol#753-760) uses assembly
        - INLINE ASM (TypedMemView.sol#755-758)
TypedMemView.joinSha2(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/crytic/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/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

SafeMath.add(uint256,uint256) (SafeMath.sol#73-77) is never used and should be removed
SafeMath.div(uint256,uint256) (SafeMath.sol#55-60) is never used and should be removed
SafeMath.mul(uint256,uint256) (SafeMath.sol#39-50) is never used and should be removed
SafeMath.sub(uint256,uint256) (SafeMath.sol#65-68) 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
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

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#65) is not in mixedCase
Parameter SafeMath.sub(uint256,uint256)._b (SafeMath.sol#65) 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
```

AUTOMATED TESTING

```
Parameter TypedMemView.nibbleHex(uint8)._b (TypedMemView.sol#77) is not in mixedCase
Parameter TypedMemView.byteHex(uint8)._b (TypedMemView.sol#104) is not in mixedCase
Parameter TypedMemView.encodeHex(uint256)._b (TypedMemView.sol#118) is not in mixedCase
Parameter TypedMemView.reverseUint256(uint256)._b (TypedMemView.sol#143) is not in mixedCase
Parameter TypedMemView.leftMask(uint8)._len (TypedMemView.sol#167) is not in mixedCase
Parameter TypedMemView.isType(bytes29,uint40)._expected (TypedMemView.sol#236) is not in mixedCase
Parameter TypedMemView.assertType(bytes29,uint40)._expected (TypedMemView.sol#247) is not in mixedCase
Parameter TypedMemView.castTo(bytes29,uint40)._newType (TypedMemView.sol#270) is not in mixedCase
Parameter TypedMemView.unsafeBuildUnchecked(uint256,uint256,uint256)._type (TypedMemView.sol#290) is not in mixedCase
Parameter TypedMemView.unsafeBuildUnchecked(uint256,uint256,uint256)._loc (TypedMemView.sol#290) is not in mixedCase
Parameter TypedMemView.unsafeBuildUnchecked(uint256,uint256,uint256)._len (TypedMemView.sol#290) is not in mixedCase
Parameter TypedMemView.build(uint256,uint256,uint256)._type (TypedMemView.sol#309) is not in mixedCase
Parameter TypedMemView.build(uint256,uint256,uint256)._loc (TypedMemView.sol#309) is not in mixedCase
Parameter TypedMemView.build(uint256,uint256,uint256)._len (TypedMemView.sol#309) is not in mixedCase
Parameter TypedMemView.slice(bytes29,uint256,uint256,uint40)._index (TypedMemView.sol#428) is not in mixedCase
Parameter TypedMemView.slice(bytes29,uint256,uint256,uint40)._len (TypedMemView.sol#428) is not in mixedCase
Parameter TypedMemView.prefix(bytes29,uint256,uint40)._len (TypedMemView.sol#447) is not in mixedCase
Parameter TypedMemView.postfix(bytes29,uint256,uint40)._len (TypedMemView.sol#458) is not in mixedCase
Parameter TypedMemView.indexErrOverrun(uint256,uint256,uint256,uint256)._loc (TypedMemView.sol#471) is not in mixedCase
Parameter TypedMemView.indexErrOverrun(uint256,uint256,uint256,uint256)._len (TypedMemView.sol#472) is not in mixedCase
Parameter TypedMemView.indexErrOverrun(uint256,uint256,uint256,uint256)._index (TypedMemView.sol#473) is not in mixedCase
Parameter TypedMemView.indexErrOverrun(uint256,uint256,uint256,uint256)._slice (TypedMemView.sol#474) is not in mixedCase
Parameter TypedMemView.index(bytes29,uint256,uint8)._index (TypedMemView.sol#505) is not in mixedCase
Parameter TypedMemView.index(bytes29,uint256,uint8)._bytes (TypedMemView.sol#505) is not in mixedCase
Parameter TypedMemView.indexUint(bytes29,uint256,uint8)._index (TypedMemView.sol#529) is not in mixedCase
Parameter TypedMemView.indexUint(bytes29,uint256,uint8)._bytes (TypedMemView.sol#529) is not in mixedCase
Parameter TypedMemView.indexLEUint(bytes29,uint256,uint8)._index (TypedMemView.sol#540) is not in mixedCase
Parameter TypedMemView.indexLEUint(bytes29,uint256,uint8)._bytes (TypedMemView.sol#540) is not in mixedCase
Parameter TypedMemView.indexAddress(bytes29,uint256)._index (TypedMemView.sol#551) is not in mixedCase
Parameter TypedMemView.unsafeCopyTo(bytes29,uint256)._newLoc (TypedMemView.sol#673) is not in mixedCase
Parameter TypedMemView.unsafeJoin(bytes29[],uint256)._location (TypedMemView.sol#729) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
```

```
TypedMemView.reverseUint256(uint256) (TypedMemView.sol#143-160) uses literals with too many digits:
        - v = ((v >> 32) & 0x00000000FFFFFFFF00000000FFFFFFFF00000000FFFFFFFF00000000FFFFFFFF) | ((v & 0x00000000FFFFFFFF00000000FFFFFFFF00000000FFFFFFFF00000000FFFFFFFF) << 32) (TypedMemView.sol#153-154)
TypedMemView.reverseUint256(uint256) (TypedMemView.sol#143-160) uses literals with too many digits:
        - v = ((v >> 64) & 0x0000000000000000FFFFFFFFFFFFFFFF0000000000000000FFFFFFFFFFFFFFFF) | ((v & 0x0000000000000000FFFFFFFFFFFFFFFF0000000000000000FFFFFFFFFFFFFFFF) << 64) (TypedMemView.sol#156-157)
TypedMemView.leftMask(uint8) (TypedMemView.sol#167-176) uses literals with too many digits:
        - mask = 0x8000000000000000000000000000000000000000000000000000000000000000 >>' _len - 1 (TypedMemView.sol#171-174)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits
```

- Majority of identified issues are related to third-party libraries.
- Reentrency 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

Report for lib/openzeppelin-contracts/contracts/utils/EnumerableSet.sol
https://dashboard.mythx.io/#/console/analyses/c23ad45d-0f68-4146-acde-5228baef01ba

Line	SWC Title	Severity	Short Description
69	(SWC-101) Integer Overflow and Underflow	High	The arithmetic operator can overflow.
81	(SWC-101) Integer Overflow and Underflow	Unknown	Compiler-rewritable "<uint> - 1" discovered
81	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
82	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
82	(SWC-101) Integer Overflow and Underflow	Unknown	Compiler-rewritable "<uint> - 1" discovered
87	(SWC-110) Assert Violation	Unknown	Out of bounds array access
90	(SWC-110) Assert Violation	Unknown	Out of bounds array access
92	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
132	(SWC-110) Assert Violation	Unknown	Out of bounds array access

Report for src/roles/Attestable.sol
https://dashboard.mythx.io/#/console/analyses/c23ad45d-0f68-4146-acde-5228baef01ba

Line	SWC Title	Severity	Short Description
15	(SWC-103) Floating Pragma	Low	A floating pragma is set.
226	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
233	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
234	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
234	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered

BurnMessage.sol

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

Line	SWC Title	Severity	Short Description
15	(SWC-103) Floating Pragma	Low	A floating pragma is set.

Message.sol

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

Line	SWC Title	Severity	Short Description
15	(SWC-103) Floating Pragma	Low	A floating pragma is set.

MessageTransmitter.sol

Report for lib/memview-sol/contracts/SafeMath.sol
https://dashboard.mythx.io/#/console/analyses/b75381a3-7a04-4094-b00d-d84ee4b65a56

Line	SWC Title	Severity	Short Description
47	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
48	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
59	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
67	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
74	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered

Report for lib/memview-sol/contracts/TypedMemView.sol
https://dashboard.mythx.io/#/console/analyses/b75381a3-7a04-4094-b00d-d84ee4b65a56

Line	SWC Title	Severity	Short Description
119	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-=" discovered
120	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
128	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-=" discovered
129	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
363	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
386	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
395	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
417	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
512	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
530	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
530	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
711	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
740	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
741	(SWC-110) Assert Violation	Unknown	Out of bounds array access
742	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
743	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
788	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered

Report for lib/openzeppelin-contracts/contracts/math/SafeMath.sol
https://dashboard.mythx.io/#/console/analyses/b75381a3-7a04-4094-b00d-d84ee4b65a56

Line	SWC Title	Severity	Short Description
25	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
37	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
50	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
51	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
62	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
72	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "%" discovered
86	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
103	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
118	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
119	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
137	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
154	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "%" discovered
172	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
192	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
212	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "%" discovered

Report for lib/openzeppelin-contracts/contracts/utils/EnumerableSet.sol
https://dashboard.mythx.io/#/console/analyses/b75381a3-7a04-4094-b00d-d84ee4b65a56

Line	SWC Title	Severity	Short Description
81	(SWC-101) Integer Overflow and Underflow	Unknown	Compiler-rewritable "<uint> - 1" discovered
81	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
82	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
82	(SWC-101) Integer Overflow and Underflow	Unknown	Compiler-rewritable "<uint> - 1" discovered
87	(SWC-110) Assert Violation	Unknown	Out of bounds array access
90	(SWC-110) Assert Violation	Unknown	Out of bounds array access
92	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
132	(SWC-110) Assert Violation	Unknown	Out of bounds array access

Report for src/MessageTransmitter.sol
https://dashboard.mythx.io/#/console/analyses/b75381a3-7a04-4094-b00d-d84ee4b65a56

Line	SWC Title	Severity	Short Description
15	(SWC-103) Floating Pragma	Low	A floating pragma is set.
387	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered

Report for src/messages/Message.sol
https://dashboard.mythx.io/#/console/analyses/b75381a3-7a04-4094-b00d-d84ee4b65a56

Line	SWC Title	Severity	Short Description
136	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered

Report for src/roles/Attestable.sol
https://dashboard.mythx.io/#/console/analyses/b75381a3-7a04-4094-b00d-d84ee4b65a56

Line	SWC Title	Severity	Short Description
226	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
233	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
234	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
234	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered

Rescuable.sol

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

Line	SWC Title	Severity	Short Description
119	(SWC-123) Requirement Violation	Low	Requirement violation.

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

Line	SWC Title	Severity	Short Description
37	(SWC-123) Requirement Violation	Low	Requirement violation.

TokenController.sol

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

Line	SWC Title	Severity	Short Description
15	(SWC-103) Floating Pragma	Low	A floating pragma is set.

TokenMessenger.sol

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

Line	SWC Title	Severity	Short Description
119	(SWC-123) Requirement Violation	Low	Requirement violation.

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

Line	SWC Title	Severity	Short Description
15	(SWC-103) Floating Pragma	Low	A floating pragma is set.
30	(SWC-123) Requirement Violation	Low	Requirement violation.

TokenMinter.sol

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

Line	SWC Title	Severity	Short Description
119	(SWC-123) Requirement Violation	Low	Requirement violation.

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

Line	SWC Title	Severity	Short Description
30	(SWC-123) Requirement Violation	Low	Requirement violation.

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

Line	SWC Title	Severity	Short Description
15	(SWC-103) Floating Pragma	Low	A floating pragma is set.

TypedMemView.sol

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

Line	SWC Title	Severity	Short Description
2	(SWC-103) Floating Pragma	Low	A floating pragma is set.

- 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