// HALBORN

Cere - Solidity Bridge

Smart Contract Security Audit

Prepared by: **Halborn**

Date of Engagement: **February 7th, 2022 - February 17th, 2022**

Visit: **Halborn.com**

DOCUMENT REVISION HISTORY

VERSION	MODIFICATION	DATE	AUTHOR
0.1	Document Creation	02/07/2022	Roberto Reigada
0.2	Document Updates	02/17/2022	Roberto Reigada
0.3	Draft Review	02/17/2022	Gabi Urrutia
1.0	Remediation Plan	02/22/2022	Roberto Reigada
1.1	Remediation Plan Review	02/22/2022	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
Roberto Reigada	Halborn	Roberto.Reigada@halborn.com

EXECUTIVE OVERVIEW

1.1 INTRODUCTION

Cerebellum Network engaged Halborn to conduct a security audit on their smart contracts beginning on February 7th, 2022 and ending on February 17th, 2022. The security assessment was scoped to the smart contracts provided in the GitHub repository Cerebellum-Network/chainbridge-solidity/.

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 accepted by Cerebellum Network team.

The most critical was in Bridge contract. The _relayerThreshold state variable records the minimum number of votes needed before a proposal could be executed by a relayer. On the other hand, in the constructor, there was not a minimum threshold for the amount of relayers. The Cerebellum Network team fixed the issue by setting the threshold to 2 and setting up 4 relayers.

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)

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.

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 smart contracts:

- ERC20Handler.sol
- ERC721Handler.sol
- GenericHandler.sol
- HandlerHelpers.sol
- Whitelist.sol
- Bridge.sol
- ERC20Safe.sol
- ERC721MinterBurnerPauser.sol
- ERC721Safe.sol

Commit ID: c1f040684391604638141c9bb3b719ec92942513

2. ASSESSMENT SUMMARY & FINDINGS OVERVIEW

CRITICAL	HIGH	MEDIUM	LOW	INFORMATIONAL
0	0	1	3	6

LIKELIHOOD

IMPACT

	(HAL-01)			
(HAL-02) (HAL-03)				
		(HAL-04)		
(HAL-05) (HAL-06) (HAL-07) (HAL-08) (HAL-09) (HAL-10)				

SECURITY ANALYSIS	RISK LEVEL	REMEDIATION DATE
HAL01 - LACK OF MINIMUM THRESHOLD FOR INITIALRELAYERS/RELAYERTHRESHOLD	Medium	SOLVED - 02/22/2022
HAL02 - LACK OF LIQUIDITY LOSS PROTECTION	Low	RISK ACCEPTED
HAL03 - ERC20SAFE.SAFECALL DOES NOT VERIFY THAT THE TOKEN ADDRESS IS A CONTRACT	Low	RISK ACCEPTED
HAL04 - MISSING ZERO ADDRESS CHECKS	Low	RISK ACCEPTED
HAL05 - WRONG INFORMATION DISPLAYED IN THE DEPOSIT RECORD WHEN SUPPLYING TRANSFER-ON-FEE OR DEFLATIONARY TOKENS	Informational	ACKNOWLEDGED
HAL06 - USE OF TRANSFER INSTEAD OF CALL TO TRANSFER ETHER	Informational	ACKNOWLEDGED
HAL07 - USING ++I CONSUMES LESS GAS THAN I++ IN LOOPS	Informational	ACKNOWLEDGED
HAL08 - UNNEEDED INITIALIZATION OF UINT256 VARIABLES TO 0	Informational	ACKNOWLEDGED
HAL09 - POSSIBLE MISUSE OF PUBLIC FUNCTIONS	Informational	ACKNOWLEDGED
HAL10 - UNUSED IMPORTS	Informational	ACKNOWLEDGED

FINDINGS & TECH DETAILS

3.1 (HAL-01) LACK OF MINIMUM THRESHOLD FOR INITIALRELAYERS/RELAYERTHRESHOLD - MEDIUM

Description:

In the contract `Bridge`, the `_relayerThreshold` state variable records the minimum number of votes needed before a proposal can be executed by a relayer. On the other hand, in the constructor, there is not a minimum threshold for the amount of relayers.

Halborn suggests setting a minimum of 2 for the `_relayerThreshold` and a minimum of 3 `initialRelayers`. This way, 2 different relayer votes will be needed before a proposal is executed. In case a relayer is compromised, the other 2 relayers will be able to mitigate his actions.

Furthermore, if a relayer is down, the other 2 relayers will be able to vote for the proposal and get it executed.

If there was just one relayer, and this relayer was down, users would never receive their funds in the destination chain.

Risk Level:

Likelihood - 2
Impact - 4

Recommendation:

It is recommended to set a minimum of 2 for the `_relayerThreshold` and a minimum of 3 `initialRelayers` to maintain redundancy in the system.

Remediation Plan:

SOLVED: The Cerebellum Network team fixed the issue by setting the threshold to 2 and setting up 4 relayers.

3.2 (HAL-02) LACK OF LIQUIDITY LOSS PROTECTION - LOW

Description:

In the contract Bridge the function adminWithdraw() allows the admin to withdraw any fund stored in the different handler contracts. A malicious admin could use this function to perform a rug pull in all the handlers.

Code location:

Listing 1: Bridge.sol (Line 283)

```
276 function adminWithdraw(
277     address handlerAddress,
278     address tokenAddress,
279     address recipient,
280     uint256 amountOrTokenID
281 ) external onlyAdmin {
282     IERCHandler handler = IERCHandler(handlerAddress);
283     handler.withdraw(tokenAddress, recipient, amountOrTokenID);
284 }
```

Risk Level:

Likelihood - 1
Impact - 3

Recommendation:

The adminWithdraw function allows the owner of the system to withdraw all the assets from the different handlers. The owner should be limited to the minimum operations possible. These functionalities should be split between multiple role based users with multi-signature wallets for each one. If possible, it is recommended to remove this function. If not, it is recommended to at least use a multisig wallet as the Admin wallet

adding also a Timelock.

Remediation Plan:

RISK ACCEPTED: The Cerebellum Network team accepts this risk as they use a multi-signature wallet to manage the admin wallet.

3.3 (HAL-03) ERC20SAFE.SAFECALL DOES NOT VERIFY THAT THE TOKEN ADDRESS IS A CONTRACT - LOW

Description:

In the contract ERC20Safe the function _safeCall() is used to perform all the token transfers:

Listing 2: ERC20Safe.sol

```
76     function _safeCall(IERC20 token, bytes memory data) private {
77         (bool success, bytes memory returndata) = address(token).
↳ call(data);
78         require(success, "ERC20: call failed");
79
80         if (returndata.length > 0) {
81
82             require(abi.decode(returndata, (bool)), "ERC20:
↳ operation did not succeed");
83         }
84     }
```

This function does not verify that the token address passed as a parameter is actually a contract allowing, as can be seen below, to perform a transfer using the zero address as the token parameter:

```
>>> contract_ERC20Safe = owner.deploy(ERC20Safe)
Transaction sent: 0x6413a8ed88023le1de7a83b4ea1101137e996d62ebdb536cc17d54fa61532f2d
  Gas price: 0.0 gwei   Gas limit: 800000000   Nonce: 6
  ERC20Safe.constructor confirmed   Block: 14191290   Gas used: 185719 (0.02%)
  ERC20Safe deployed at: 0x9b27b7532EcD0D4ba72e904119AC63c7786d126F

>>> tx1 = contract_ERC20Safe.fundERC20('0x0000000000000000000000000000000000000000', user2, 99999999999999999999999999999999999999999999, {'from': user1})
Transaction sent: 0xc7cb9ed8dbd9091a7c41926fb7bd07da80b7884106eda52e3265c5d138329c31
  Gas price: 0.0 gwei   Gas limit: 800000000   Nonce: 0
  ERC20Safe.fundERC20 confirmed   Block: 14191291   Gas used: 23750 (0.00%)

>>> calltrace(tx1)
Call trace for '0xc7cb9ed8dbd9091a7c41926fb7bd07da80b7884106eda52e3265c5d138329c31':
Initial call cost  [21916 gas]
ERC20Safe.fundERC20  0:332  [278 / 1834 gas]
└── ERC20Safe._safeTransferFrom  70:318  [1556 gas]
```

This was recently exploited in the Qubit Finance's bridge.

The likelihood of this exploit is very low as the token used in the deposits must pass this check:

```
require(_contractWhitelist[tokenAddress], "provided tokenAddress is not
 whitelisted");
```

This scenario would only become real if an admin called `Bridge.adminSetResource` with an invalid/wrong `tokenAddress`.

Risk Level:

Likelihood - 1
Impact - 3

Recommendation:

It is recommended to follow OpenZeppelin's approach and check in the `_safeCall()` function that the `token` address is a contract address. To achieve that, the function `functionCall()` instead of `call()` from OpenZeppelin's `Address.sol` contract can be used.

Also another possible addition would be checking the balance before and after the asset transfer to ensure that the number of the transferred asset compiles the expectation.

Remediation Plan:

RISK ACCEPTED: The Cerebellum Network team accepts the risk of this finding.

3.4 (HAL-04) MISSING ZERO ADDRESS CHECKS - LOW

Description:

Multiple contracts are missing address validation in their constructors. Every address should be validated and checked that is different from zero.

Code location:

ERC20Handler.sol
- Line 38: address bridgeAddress
- Line 40: address[] memory initialContractAddresses

ERC721Handler.sol
- Line 49: address bridgeAddress
- Line 51: address[] memory initialContractAddresses
- Line 52: address[] memory burnableContractAddresses

GenericHandler.sol
- Line 66: address bridgeAddress
- Line 68: address[] memory initialContractAddresses

Bridge.sol
- Line 110: address[] memory initialRelayers
- Line 141: grantRole(DEFAULT_ADMIN_ROLE, newAdmin);
- Line 180: grantRole(RELAYER_ROLE, relayerAddress);
- Line 207: _resourceIDToHandlerAddress[resourceID] = handlerAddress;
- Line 227: _resourceIDToHandlerAddress[resourceID] = handlerAddress;

Risk Level:

Likelihood - 3
Impact - 2

Recommendation:

It is recommended to validate that every address input is different from zero.

Remediation Plan:

RISK ACCEPTED: The Cerebellum Network team accepts the risk of this finding.

3.5 (HAL-05) WRONG INFORMATION DISPLAYED IN THE DEPOSIT RECORD WHEN SUPPLYING TRANSFER-ON-FEE OR DEFLATIONARY TOKENS - INFORMATIONAL

Description:

In the contract ERC20Handler.sol, the deposit function assumes that the total amount of tokens that will be locked in the smart contract after calling lockERC20(tokenAddress, depositer, address(this), amount); is equal to the amount parameter (and thus it creates a depositRecord accordingly).

Listing 3: ERC20Handler.sol - deposit (Lines 111,120)

```
82      function deposit(
83          bytes32 resourceID,
84          uint8   destinationChainID,
85          uint64  depositNonce,
86          address depositer,
87          bytes   calldata data
88      ) external override onlyBridge {
89          bytes   memory recipientAddress;
90          uint256        amount;
91          uint256        lenRecipientAddress;
92
93          assembly {
94
95              amount := calldataload(0xC4)
96
97              recipientAddress := mload(0x40)
98              lenRecipientAddress := calldataload(0xE4)
99              mstore(0x40, add(0x20, add(recipientAddress,
 ↳ lenRecipientAddress)))
100
101             calldatacopy(
```

```
102                 recipientAddress, // copy to
 ↳ destinationRecipientAddress
103                 0xE4, // copy from calldata @ 0x104
104                 sub(calldatasize(), 0xE) // copy size (
 ↳ calldatasize - 0x104)
105             )
106         }
107
108         address tokenAddress = _resourceIDToTokenContractAddress[
 ↳ resourceID];
109         require(_contractWhitelist[tokenAddress], "provided
 ↳ tokenAddress is not whitelisted");
110
111         lockERC20(tokenAddress, depositer, address(this), amount);
112
113         _depositRecords[destinationChainID][depositNonce] =
 ↳ DepositRecord(
114             tokenAddress,
115             uint8(lenRecipientAddress),
116             destinationChainID,
117             resourceID,
118             recipientAddress,
119             depositer,
120             amount
121         );
122     }
```

However, this may not be true if the tokenAddress is a transfer-on-fee token or a deflationary/rebasing token, causing the received amount to be less than the accounted amount in depositRecord.

Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

It is recommended to get the actual received amount by calculating the difference of token balance before and after the transfers.

Remediation Plan:

ACKNOWLEDGED: The Cerebellum Network team claims that Bridge uses only the CERE token. The CERE token is not a transfer-on-fee or a deflationary token.

3.6 (HAL-06) USE OF TRANSFER INSTEAD OF CALL TO TRANSFER ETHER - INFORMATIONAL

Description:

In the contract `Bridge`, the `transferFunds` function uses `address.transfer()` to transfer Ether to a set of addresses. Any smart contract that uses `transfer()` or `send()` is taking a hard dependency on gas costs by forwarding a fixed amount of gas: `2300`.

Since its introduction, `transfer()` has typically been recommended by the security community because it helps guard against reentrancy attacks. This guidance made sense under the assumption that gas costs wouldn't change, but that assumption turned out to be incorrect. As gas costs are subject to change, then smart contracts can't depend on any particular gas costs, that's why `transfer()` and `send()` should be avoided. `call()` should be used instead.

Code Location:

Listing 4: Bridge.sol (Line 429)

```
427     function transferFunds(address payable[] calldata addrs, uint
 ↳ [] calldata amounts) external onlyAdmin {
428         for (uint i = 0; i < addrs.length; i++) {
429             addrs[i].transfer(amounts[i]);
430         }
431     }
```

Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

It is recommended to use `call()` instead of `transfer()` to transfer Ether.

Remediation Plan:

ACKNOWLEDGED: The Cerebellum Network team acknowledged this finding.

3.7 (HAL-07) USING ++I CONSUMES LESS GAS THAN I++ IN LOOPS - INFORMATIONAL

Description:

In the loop below, the variable `i` is incremented using `i++`. It is known that, in loops, using `++i` costs less gas per iteration than `i++`.

Code Location:

ERC20Handler.sol
- Line 47: `for (uint256 i = 0; i < initialResourceIDs.length; i++){`

ERC721Handler.sol
- Line 59: `for (uint256 i = 0; i < initialResourceIDs.length; i++){`
- Line 63:
`for (uint256 i = 0; i < burnableContractAddresses.length; i++){`

GenericHandler.sol
- Line 83: `for (uint256 i = 0; i < initialResourceIDs.length; i++){`

Bridge.sol
- Line 120: `for (uint i; i < initialRelayers.length; i++){`
- Line 122: `_totalRelayers++;`
- Line 428: `for (uint i = 0; i < addrs.length; i++){`

Proof of Concept:

For example, based in the following test contract:

Listing 5: 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 }
```

```
>>> test_contract.postiincrement(1)
Transaction sent: 0x1ecede6b109b707786d3685bd71dd9f22dc389957653036ca04c4cd2e72c5e0b
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 44
  test.postiincrement confirmed   Block: 13622335   Gas used: 21620 (0.32%)

<Transaction '0x1ecede6b109b707786d3685bd71dd9f22dc389957653036ca04c4cd2e72c5e0b'>
>>> 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: 0x98c04430526a59b1cf947c114b62666a4417165947d31bf300cd6ae68328033
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 46
  test.postiincrement confirmed   Block: 13622337   Gas used: 22673 (0.34%)

<Transaction '0x98c04430526a59b1cf947c114b62666a4417165947d31bf300cd6ae68328033'>
>>> test_contract.preiincrement(10)
Transaction sent: 0xf060d04714eff8482a828342414d5a20be9958c822d42860e7992aba20e1de05
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 47
  test.preiincrement confirmed   Block: 13622338   Gas used: 22601 (0.34%)

<Transaction '0xf060d04714eff8482a828342414d5a20be9958c822d42860e7992aba20e1de05'>
```

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. This does not only apply to the iterator variable. It also applies to variables declared inside the loop code block.

Remediation Plan:

ACKNOWLEDGED: The Cerebellum Network team acknowledged this finding.

3.8 (HAL-08) 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:

ERC20Handler.sol
- Line 47: `for (uint256 i = 0; i < initialResourceIDs.length; i++){`

ERC721Handler.sol
- Line 59: `for (uint256 i = 0; i < initialResourceIDs.length; i++){`
- Line 63:
`for (uint256 i = 0; i < burnableContractAddresses.length; i++){`

GenericHandler.sol
- Line 83: `for (uint256 i = 0; i < initialResourceIDs.length; i++){`

Bridge.sol
- Line 428: `for (uint i = 0; i < addrs.length; 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 < initialResourceIDs.length; ++i){`.

Remediation Plan:

ACKNOWLEDGED: The Cerebellum Network team acknowledged this finding.

3.9 (HAL-09) POSSIBLE MISUSE OF PUBLIC FUNCTIONS - INFORMATIONAL

Description:

In the following contracts there are functions marked as public but they are never directly called within the same contract or in any of their descendants:

ERC721MinterBurnerPauser.sol
- mint() (ERC721MinterBurnerPauser.sol#44-49)
- pause() (ERC721MinterBurnerPauser.sol#60-63)
- unpause() (ERC721MinterBurnerPauser.sol#74-77)

ERC721Safe.sol
- fundERC721() (ERC721Safe.sol#21-24)

ERC20Safe.sol
- fundERC20() (ERC20Safe.sol#20-23)

Bridge.sol
- cancelProposal() (Bridge.sol#380-390)

Whitelist.sol
- enableWhitelist() (Whitelist.sol#69-73)
- disableWhitelist() (Whitelist.sol#80-84)
- removeFromWhitelist() (Whitelist.sol#112-116)
- addToWhitelist() (Whitelist.sol#100-104)

Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

If the functions are not intended to be called internally or by their descendants, it is better to mark all of these functions as external to reduce gas costs.

Remediation Plan:

ACKNOWLEDGED: The Cerebellum Network team acknowledged this finding.

3.10 (HAL-10) UNUSED IMPORTS - INFORMATIONAL

Description:

In the contracts ERC20Safe and ERC721Safe SafeMath.sol is imported, but it is not used anywhere in the code.

Code location:

Listing 6: ERC20Safe.sol (Lines 3,12)

```
3  import "@openzeppelin/contracts/math/SafeMath.sol";
4  import "@openzeppelin/contracts/token/ERC20/IERC20.sol";
5
6  /**
7      @title Manages deposited ERC20s.
8      @author ChainSafe Systems.
9      @notice This contract is intended to be used with ERC20Handler
↳  contract.
10  */
11 contract ERC20Safe {
12     using SafeMath for uint256;
```

Listing 7: ERC721Safe.sol (Lines 3,13)

```
3  import "@openzeppelin/contracts/math/SafeMath.sol";
4  import "@openzeppelin/contracts/token/ERC721/IERC721.sol";
5  import "./ERC721MinterBurnerPauser.sol";
6
7  /**
8      @title Manages deposited ERC721s.
9      @author ChainSafe Systems.
10     @notice This contract is intended to be used with
↳ ERC721Handler contract.
11  */
12 contract ERC721Safe {
13     using SafeMath for uint256;
```

Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

It is recommended to remove the unused imports to reduce the gas deployment costs.

Remediation Plan:

ACKNOWLEDGED: The Cerebellum Network team acknowledges this finding.

MANUAL TESTING

Halborn performed different manual tests in the contracts, trying to reproduce the flow described in the Chainbridge documentation:

4.1 ERC20HANDLER: AS SOURCE CHAIN

1. Some user calls the deposit function on the bridge contract. A depositRecord is created on the bridge and a call is delegated to a handler contract, in this test ERC20Handler which is specified by the provided resourceID.

2. The specified handler's deposit function validates the parameters provided by the user. If successful, a depositRecord is created on the handler.

3. If the call delegated to the handler is successful, the bridge emits a Deposit event.

4. Relayers parse the Deposit event and retrieve the associated DepositRecord from the handler to construct a message.

```
ERCDepositData -> 0x0000000000000000000000000000000000000000000000001158e460913d00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000145bb330eac6bbb05a55fc2b69d55c5812b9e77603

contract_Token1.balanceOf(depositer) -> 100000000000000000000
contract_Token1.balanceOf(contract_ERC20Handler) -> 0

Calling -> contract_Bridge_mainnet.deposit(137, bytes32_resourceID1, ERCDepositData, ('from': depositer))
Transaction sent: 0xe78839fe247757d226b14ebb38fca65a0fe6dd70e3cc6fa1025946a03db0b65b
  Gas price: 0.0 gwei   Gas limit: 800000000   Nonce: 1
  Bridge.deposit confirmed   Block: 14192669   Gas used: 268712 (0.03%)

Call trace for '0xe78839fe247757d226b14ebb38fca65a0fe6dd70e3cc6fa1025946a03db0b65b':
Initial call cost  [7344 gas]
Bridge.deposit  0:2149  [108889 / 246368 gas]
├── Whitelist.isWhitelistEnabled  190:197  [821 gas]
├── Whitelist.isWhitelisted  207:233  [917 gas]
└── ERC20Handler.deposit  [CALL]  64:2108  [111167 / 135741 gas]
    ├── address: contract_ERC20Handler.address
    ├── value: 0
    └── input arguments:
        ├── resourceID: 0x0000000000000000000000000000000000000000000000000000000000000001
        ├── destinationChainID: 137
        ├── depositNonce: 1
        ├── depositer: depositer.address
        └── data: 0x0000000000000000000000000000000000000000000000001158e460913d00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000145bb330eac6bbb05a55fc2b69d55c5812b9e77603

    ERC20Safe.lockERC20  862:1768  [49 / 24574 gas]
    └── ERC20Safe._safeTransferFrom  971:1761  [227 / 24525 gas]
        └── ERC20Safe._safeCall  1040:1755  [763 / 24298 gas]
            └── Token.transferFrom  [CALL]  1187:1682  [1052 / 23535 gas]
                ├── address: contract_Token1.address
                ├── value: 0
                ├── input arguments:
                │   ├── sender: depositer.address
                │   ├── recipient: contract_ERC20Handler.address
                │   └── amount: 200000000000000000000
                └── return value: True

                ERC20.transferFrom  1269:1664  [1177 / 22483 gas]
                ├── ERC20._transfer  1277:1476  [29014 / 29144 gas]
                │   ├── ERC20._beforeTokenTransfer  1304:1309  [15 gas]
                │   ├── SafeMath.sub  1312:1368  [56 gas]
                │   └── SafeMath.add  1406:1424  [59 gas]
                ├── Context._msgSender  1482:1486  [14 gas]
                ├── Context._msgSender  1527:1531  [14 gas]
                ├── SafeMath.sub  1556:1573  [56 gas]
                └── ERC20._approve  1576:1655  [-7922 gas]


txi.events -> {'Transfer': [OrderedDict({('from', '0x899f5D37F1854892a6DfB2BAcb3BfA1A2ee85DC5'), ('to', '0x347934666810791a7761B690Fa41834 2CaaA54CF'), ('value', 200000000000000000000)})], 'Approval': [OrderedDict({('owner', '0x899f5D37F18
54892a6DfB2BAcb3BfA1A2ee85DC5'), ('spender', '0x3479346668107 91a7761B690Fa41834 2CaaA54CF'), ('value', 0)})], 'Deposit': [OrderedDict({('destinationChainID', 137), ('resourceID', 0x0000000000000000000000000000000000000000000000000000000000000001), ('depositNonce', 1)})]}
contract_Token1.balanceOf(depositer) -> 800000000000000000
contract_Token1.balanceOf(contract_ERC20Handler) -> 200000000000000000

contract_ERC20Handler.getDepositRecord(txi.events[2]['depositNonce'], txi.events[2]['destinationChainID']) -> ('0xa263077a9130895C97524ac43d17C3d7E56B069A', 20, 137, 0x0000000000000000000000000000000000000000000000000000000000000001, 0x5
bb330eac6bbb05a55fc2b69d55c5812b9e77603, '0x899f5D37F1854892a6DfB2BAcb3BfA1A2ee85DC5', 200000000000000000000)
```

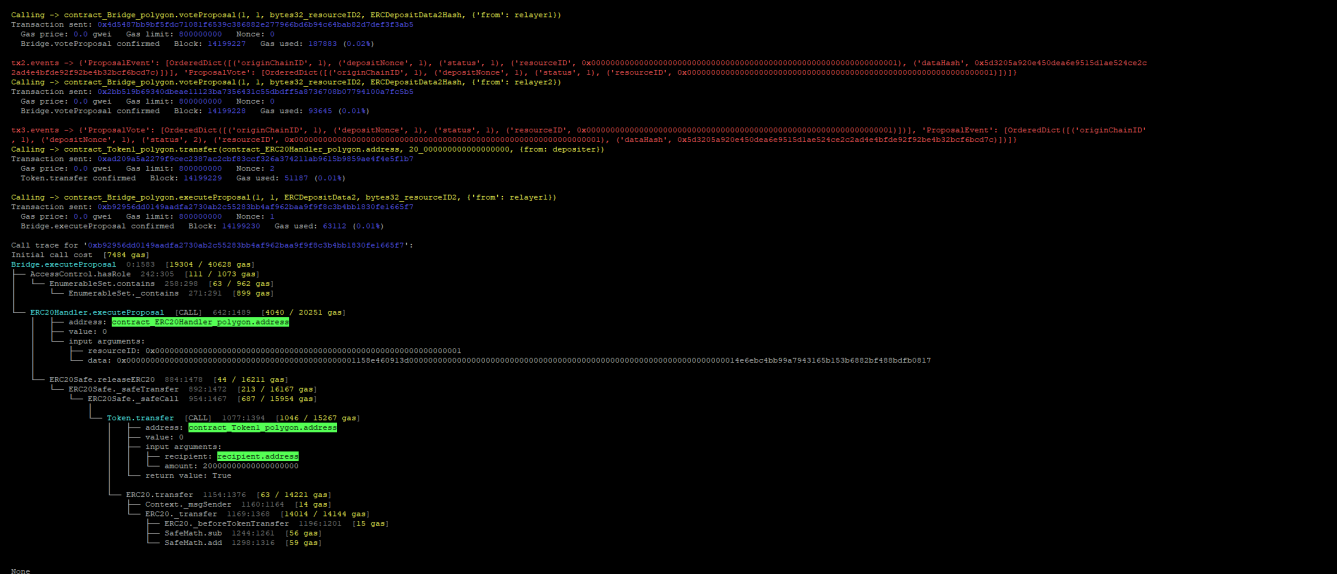
No major issues were found in this flow:
- depositer address will usually have to be whitelisted by an Admin. Although this functionality can be removed and it is possible that users can deposit without being whitelisted.

- The handler addresses are only set by admins.
- There is a require check in the line 298 of the Bridge contract that checks that the handler is set.
- The ERC20Handler contract checks that the tokenAddress is whitelisted. This is critical for the security of the bridge.
- There is no way to tamper with the calldata passed to the ERC20Handler .deposit() function.

On the other hand, the ERC20Handler contract does not check for deflationary/transfer-on-fee tokens.

4.2 ERC20HANDLER: AS DESTINATION CHAIN

1. A Relayer calls voteProposal on the Bridge contract. If a proposal corresponding with the parameters passed in does not exist, it is created and the Relayer's vote is recorded. If the proposal already exists, the Relayer's vote is simply recorded.
2. Once we have met some vote threshold for a proposal, in this test 2 votes was the threshold, the Bridge emits a ProposalFinalized event.
3. Upon seeing a ProposalFinalized event, Relayers call the executeDeposit function on the Bridge. executeDeposit delegates a call to a handler contract specified by the associated resourceID.
4. The specified handler's executeDeposit function validates the parameters provided and makes a call to some contract to complete the transfer.



No issues were found in this flow:

- Only relayers can vote or execute proposals.
- The handler addresses are only set by admins.
- The ERC20Handler contract once again checks that the tokenAddress is whitelisted.

4.3 ERC721HANDLER: AS SOURCE CHAIN

No issues were found in this flow:

- depositer address will usually have to be whitelisted by an Admin. Although this functionality can be removed and it is possible that users can deposit without being whitelisted.
- The handler addresses are only set by admins.
- There is a require check in the line 298 of the Bridge contract that checks that the handler is set.
- The ERC721Handler contract checks that the tokenAddress is whitelisted. This is critical for the security of the bridge.
- There is no way to tamper with the calldata passed to the ERC721Handler .deposit() function.

```
Calling -> contract_Bridge_mainnet.deposit(137, bytes32_resourceID1, ERCDepositData1, {'from': depositer})
Transaction sent: 0xe4170dea1fae8d7159c5b8dba59f45039cdf932ed144ada8af4f6cc0fe4fc6a
  Gas price: 0.0 gwei   Gas limit: 800000000   Nonce: 1
  Bridge.deposit confirmed   Block: 14198867   Gas used: 338506 (0.04%)

Call trace for '0xe4170dea1fae8d7159c5b8dba59f45039cdf932ed144ada8af4f6cc0fe4fc6a':
Initial call cost  [-37740 gas]
Bridge.deposit  0:4906  [108889 / 316246 gas]
├── Whitelist.isWhitelistEnabled  190:257  [821 gas]
├── Whitelist.isWhitelisted  207:233  [917 gas]
└── ERC721Handler.deposit  [CALL]  641:4865  [134773 / 205619 gas]
    ├── address: contract_ERC721Handler_mainnet.address
    ├── value: 0
    └── input arguments:
        ├── resourceID: 0x0000000000000000000000000000000000000000000000000000000000000001
        ├── destinationChainID: 137
        ├── depositNonce: 1
        ├── depositer: depositer.address
        └── data: 0x0000000000000000000000000000000000000000000000000000000000000001000000000000000000000000000000000000000000000000000000000000014e68c250a0013fb7c6f617b950d07dea98a84e2b7

    ├── ERC165Checker.supportsInterface  969:2165  [84 / 8592 gas]
    ├── ERC165Checker.supportsERC165  975:1768  [114 / 5710 gas]
    │   ├── ERC165Checker._supportsERC165Interface  984:1363  [93 / 2798 gas]
    │   │   └── ERC165Checker._callERC165SupportsInterface  993:1342  [789 / 2705 gas]
    │   │       │
    │   │       └── ERC721MinterBurnerPauser.supportsInterface  [STATICCALL]  1150:1256  [999 / 1916 gas]
    │   │           ├── address: contract_ERC721MinterBurnerPauser_mainnet.address
    │   │           ├── input arguments:
    │   │           │   └── interfaceId: 0x01ffc9a7
    │   │           └── return value: True
    │   │
    │   │           ERC165.supportsInterface  1216:1243  [917 gas]
    │   └── ERC165Checker._supportsERC165Interface  1379:1750  [93 / 3798 gas]
    │       └── ERC165Checker._callERC165SupportsInterface  1388:1737  [789 / 2705 gas]
    │           │
    │           └── ERC721MinterBurnerPauser.supportsInterface  [STATICCALL]  1545:1656  [999 / 1916 gas]
    │               ├── address: contract_ERC721MinterBurnerPauser_mainnet.address
    │               ├── input arguments:
    │               │   └── interfaceId: 0xffffffff
    │               └── return value: False
    │
    │               ERC165.supportsInterface  1611:1638  [917 gas]
    ├── ERC165Checker._supportsERC165Interface  1779:2158  [93 / 3798 gas]
    │   └── ERC165Checker._callERC165SupportsInterface  1788:2137  [789 / 2705 gas]
    │       │
    │       └── ERC721MinterBurnerPauser.supportsInterface  [STATICCALL]  1945:2056  [999 / 1916 gas]
    │           ├── address: contract_ERC721MinterBurnerPauser_mainnet.address
    │           ├── input arguments:
    │           │   └── interfaceId: 0x5b5e139f
    │           └── return value: True
    │
    │           ERC165.supportsInterface  2011:2038  [917 gas]
    │
    ├── ERC721MinterBurnerPauser.tokenURI  [STATICCALL]  2224:2906  [1330 / 6377 gas]
    │   ├── address: contract_ERC721MinterBurnerPauser_mainnet.address
    │   ├── input arguments:
    │   │   └── tokenId: 1
    │   └── return value: https://halborn.com/1
    │
    ├── ERC721.tokenURI  2286:2799  [1673 / 5047 gas]
    │   ├── ERC721._exists  2292:2341  [43 / 991 gas]
    │   │   └── EnumerableMap.contains  2298:2335  [49 / 948 gas]
    │   │       └── EnumerableSet._contains  2306:2326  [899 gas]
    │   ├── ERC721.baseURI  2421:2508  [1887 gas]
    │   └── Strings.toString  2530:2679  [496 gas]
    └── ERC721Safe.lockERC721  3126:4421  [963 / 55877 gas]
        │
        └── ERC721MinterBurnerPauser.transferFrom  [CALL]  3166:4405  [1005 / 54914 gas]
            ├── address: contract_ERC721MinterBurnerPauser_mainnet.address
            ├── value: 0
            └── input arguments:
                ├── from: depositer.address
                ├── to: contract_ERC721Handler_mainnet.address
                └── tokenId: 1

            ERC721.transferFrom  3279:4403  [86 / 53909 gas]
            ├── Context._msgSender  3284:3288  [14 gas]
            ├── ERC721._isApprovedOrOwner  3292:3593  [225 / 5952 gas]
            │   ├── ERC721._exists  3298:3347  [43 / 991 gas]
            │   │   └── EnumerableMap.contains  3305:3341  [49 / 948 gas]
            │   │       └── EnumerableSet._contains  3312:3332  [899 gas]
            │   ├── ERC721.ownerOf  3356:3461  [118 / 2809 gas]
            │   │   └── EnumerableMap.get  3382:3455  [54 / 2691 gas]
            │   │       └── EnumerableMap._get  3390:3445  [2637 gas]
            │   └── ERC721.getApproved  3494:3573  [936 / 1927 gas]
            │       └── ERC721._exists  3500:3549  [43 / 991 gas]
            │           └── EnumerableMap.contains  3507:3543  [49 / 948 gas]
            │               └── EnumerableSet._contains  3514:3534  [899 gas]
            └── ERC721._transfer  3603:4398  [2368 / 47857 gas]
                ├── ERC721.ownerOf  3615:3720  [115 / 2806 gas]
                │   └── EnumerableMap.get  3641:3716  [54 / 2691 gas]
                │       └── EnumerableMap._get  3649:3704  [2637 gas]
                ├── ERC721MinterBurnerPauser._beforeTokenTransfer  3747:3791  [39 / 946 gas]
                │   └── ERC721Pausable._beforeTokenTransfer  3754:3786  [71 / 907 gas]
                │       ├── ERC721.approve  3761:3766  [15 gas]
                │       └── Pausable.paused  3770:3777  [821 gas]
                ├── ERC721._approve  3797:3961  [-7114 / -4308 gas]
                │   └── ERC721.ownerOf  3836:3941  [115 / 2806 gas]
                │       └── EnumerableMap.get  3862:3935  [54 / 2691 gas]
                │           └── EnumerableMap._get  3870:3925  [2637 gas]
                ├── EnumerableSet.remove  3964:4155  [49 / -23680 gas]
                │   └── EnumerableSet._remove  3991:4146  [-23729 gas]
                ├── EnumerableSet.add  4179:4274  [49 / 62795 gas]
                │   └── EnumerableSet._add  4186:4268  [61847 / 62746 gas]
                │       └── EnumerableSet._contains  4193:4213  [899 gas]
                └── EnumerableMap.set  4282:4367  [72 / 6930 gas]
                    └── EnumerableMap._set  4296:4357  [6858 gas]


txl.events -> {'Approval': [OrderedDict([('owner', '0xEF4E39Cf936278B9C66A803c9929E75D3CFB637'), ('approved', '0x0000000000000000000000000000000000000000'), ('tokenId', 1)])], 'Transfer': [OrderedDict([('from', '0xEF4E39Cf936278B9C66A803c9929E75D3CFB637'), ('to', '0x6ae54bTAd120602B363DF2B64837525D90d26C885'), ('tokenId', 1)])], 'Deposit': [OrderedDict([('destinationChainID', 137), ('resourceID', '0x0000000000000000000000000000000000000000000000000000000000000001'), ('depositNonce', 1)])]}
contract_ERC721MinterBurnerPauser_mainnet.balanceOf(depositer) -> 0
contract_ERC721MinterBurnerPauser_mainnet.balanceOf(contract_ERC721Handler_mainnet) -> 1

contract_ERC721Handler_mainnet.getDepositRecord(txl.events[2]['depositNonce'], txl.events[2]['destinationChainID']) -> ('0xCe2740176090eeab68f9227792B45804622b07C', 20, 137, 0x0000000000000000000000000000000000000000000000000000000000000001, 0xe68c250a0013fb7c6f617b950d07dea98a84e2b7, '0xEF4E39Cf936278B9C66A803c9929E75D3CFB637', 1, 0x68747470733a2f2f68616c6269746f726e6361636f6d2f31)
```

4.4 ERC721HANDLER: AS DESTINATION CHAIN

No issues were found in this flow:

- Only relayers can vote or execute proposals.
- The handler addresses are only set by admins.
- The ERC721Handler contract checks that the tokenAddress is whitelisted.

```
Calling -> contract_Bridge_polygon.executeProposal(1, 1, ERCDepositData2, bytes32_resourceID2, {'from': relayer1})
Transaction sent: 0x15deee2768bb57dfc50891cad32741daa15f1360c987e547bf41526d2040bbf4
  Gas price: 0.0 gwei   Gas limit: 800000000   Nonce: 1
  Bridge.executeProposal confirmed   Block: 14199212   Gas used: 102308 (0.01%)

Call trace for '0x15deee2768bb57dfc50891cad32741daa15f1360c987e547bf41526d2040bbf4':
Initial call cost  [-37068 gas]
Bridge.executeProposal  0:2246  [19334 / 79376 gas]
├── AccessControl.hasRole  242:305  [111 / 1073 gas]
│   └── EnumerableSet.contains  258:298  [63 / 962 gas]
│       └── EnumerableSet._contains  271:291  [899 gas]
│
└── ERC721Handler.executeProposal  [CALL]  642:2152  [5078 / 58969 gas]
    │   ├── address: contract_ERC721Handler_polygon.address
    │   ├── value: 0
    │   └── input arguments:
    │       ├── resourceID: 0x0000000000000000000000000000000000000000000000000000000000000001
    │       └── data: 0x0000000000000000000000000000000000000000000000000000000000000001000000000000000000000000000000000000000000000000000000000000014f1a0d76942544c3820c97cf89e3e7fc75ae755060000000000000000000000000000000000000000000000
0000000000000000151468747470733a2f2f68616c626f726e2e636f6d2f
    │
    └── ERC721Safe.lockERC721  944:2140  [966 / 53891 gas]
        │
        └── ERC721MinterBurnerPauser.transferFrom  [CALL]  1014:2124  [1005 / 52925 gas]
            │   ├── address: contract_ERC721MinterBurnerPauser_polygon.address
            │   ├── value: 0
            │   └── input arguments:
            │       ├── from: contract_ERC721Handler_polygon.address
            │       ├── to: recipient.address
            │       └── tokenId: 1
            │
            └── ERC721.transferFrom  1097:2122  [86 / 51920 gas]
                ├── Context._msgSender  1102:1106  [14 gas]
                ├── ERC721._isApprovedOrOwner  1110:1312  [163 / 3963 gas]
                │   ├── ERC721._exists  1116:1165  [43 / 991 gas]
                │   │   └── EnumerableMap.contains  1123:1159  [49 / 948 gas]
                │   │       └── EnumerableSet._contains  1130:1150  [899 gas]
                │   └── ERC721.ownerOf  1174:1279  [118 / 2809 gas]
                │       └── EnumerableMap.get  1200:1273  [54 / 2691 gas]
                │           └── EnumerableMap._get  1208:1263  [2637 gas]
                └── ERC721._transfer  1322:2117  [2368 / 47857 gas]
                    ├── ERC721.ownerOf  1334:1439  [115 / 2806 gas]
                    │   └── EnumerableMap.get  1360:1433  [54 / 2691 gas]
                    │       └── EnumerableMap._get  1368:1423  [2637 gas]
                    ├── ERC721MinterBurnerPauser._beforeTokenTransfer  1466:1510  [39 / 946 gas]
                    │   └── ERC721Pausable._beforeTokenTransfer  1473:1505  [71 / 907 gas]
                    │       ├── ERC721.approve  1480:1485  [15 gas]
                    │       └── Pausable.paused  1489:1496  [821 gas]
                    ├── ERC721._approve  1516:1680  [-7114 / -4308 gas]
                    │   └── ERC721.ownerOf  1555:1660  [115 / 2806 gas]
                    │       └── EnumerableMap.get  1581:1654  [54 / 2691 gas]
                    │           └── EnumerableMap._get  1589:1644  [2637 gas]
                    ├── EnumerableSet.remove  1703:1874  [49 / -23680 gas]
                    │   └── EnumerableSet._remove  1710:1865  [-23729 gas]
                    ├── EnumerableSet.add  1898:1993  [49 / 62795 gas]
                    │   └── EnumerableSet._add  1905:1984  [61847 / 62746 gas]
                    │       └── EnumerableSet._contains  1912:1932  [899 gas]
                    └── EnumerableMap.set  2001:2086  [72 / 6930 gas]
                        └── EnumerableMap._set  2015:2076  [6858 gas]


tx4.events -> {'Approval': [OrderedDict([('owner', '0xfCaBf373A2f1ebEd4BA450ffB39d512E46043159'), ('approved', '0x0000000000000000000000000000000000000000'), ('tokenId', 1)])], 'Transfer': [OrderedDict([('from', '0xfCaBf373A2f1ebEd4BA450
ffB39d512E46043159'), ('to', '0xF1a0d76942544c3820C97Cf89E3E7fC75Ae75506'), ('tokenId', 1)])], 'ProposalEvent': [OrderedDict([('originChainID', 1), ('depositNonce', 1), ('status', 3), ('resourceID', 0x0000000000000000000000000000000000000000

000000000000000000000001), ('dataHash', 0xe92a54f1af66641fbd34efbb45d672821956d7ded90d4d3104eff52155cd6a4d)])]}

contract_ERC721MinterBurnerPauser_polygon.balanceOf(recipient) -> 1
```

4.5 GENERICHANDLER: AS SOURCE CHAIN

No issues were found in this flow:

- depositer address will usually have to be whitelisted by an Admin. Although this functionality can be removed and it is possible that users can deposit without being whitelisted.
- The handler addresses are only set by admins.
- There is a require check in the line 298 of the Bridge contract that checks that the handler is set.
- The GenericHandler contract checks that the contractAddress is whitelisted. This is critical for the security of the bridge.

```
Calling -> contract_Bridge_mainnet.deposit(137, bytes32_resourceID1, ERCDepositData1, {'from': depositer})
Transaction sent: 0xd7e3b466e68a2a9f8d323ff812a31d555fe9501f7404252a51f869f498f947d9
  Gas price: 0.0 gwei   Gas limit: 800000000   Nonce: 3
  Bridge.deposit confirmed   Block: 14199646   Gas used: 226452 (0.03%)

Call trace for '0xd7e3b466e68a2a9f8d323ff812a31d555fe9501f7404252a51f869f498f947d9':
Initial call cost  [21904 gas]
Bridge.deposit  0:1618  [88816 / 204548 gas]
├── Whitelist.isWhitelistEnabled  190:197  [821 gas]
├── Whitelist.isWhitelisted  207:233  [917 gas]
│
└── GenericHandler.deposit  [CALL]  620:1577  [90162 / 113994 gas]
    │   ├── address: contract_GenericHandler_mainnet.address
    │   ├── value: 0
    │   └── input arguments:
    │       ├── resourceID: 0x0000000000000000000000000000000000000000000000000000000000000001
    │       ├── destinationChainID: 137
    │       ├── depositNonce: 1
    │       ├── depositer: depositer.address
    │       └── data: 0x0000000000000000000000000000000000000000000000000000000000000020000000000000000000000000000000000000000000000000000000000000007b
    │
    └── CentrifugeAsset.store  [CALL]  1168:1269  [23832 gas]
            ├── address: contract_CentrifugeAsset_mainnet.address
            ├── value: 0
            └── input arguments:
                └── asset: 0x000000000000000000000000000000000000000000000000000000000000007b


tx1.events -> {'AssetStored': [OrderedDict([('asset', 0x000000000000000000000000000000000000000000000000000000000000007b)])], 'Deposit': [OrderedDict([('destinationChainID', 137), ('resourceID', 0x0000000000000000000000000000000000000000
000000000000000000000001), ('depositNonce', 1)])]}
contract_CentrifugeAsset_mainnet._assetsStored(eth_abi.encode_single('(bytes32)', [to_bytes(123)])) -> True

contract_GenericHandler_mainnet.getDepositRecord(tx1.events[1]['depositNonce'], tx1.events[1]['destinationChainID']) -> (137, '0x95a045392111Ba1Ec498C784531CA932F63a1995', 0x000000000000000000000000000000000000000000000000000000000000000
1, 0x000000000000000000000000000000000000000000000000000000000000007b)
```

MANUAL TESTING

4.6 GENERICHANDLER: AS DESTINATION CHAIN

No issues were found in this flow:

- Only relayers can vote or execute proposals.
- The handler addresses are only set by admins.
- The GenericHandler contract checks that the contractAddress is whitelisted.

```
Calling -> contract_Bridge_polygon.voteProposal(1, 1, bytes32_resourceID2, ERCDepositData2Hash, {'from': relayer1})
Transaction sent: 0x6bd743871ae04491500fb882c3c3f5b62671894128b4b002a1ace6a8b8a65fee
  Gas price: 0.0 gwei   Gas limit: 800000000   Nonce: 1
  Bridge.voteProposal confirmed   Block: 14199992   Gas used: 187883 (0.02%)

tx2.events -> {'ProposalEvent': [OrderedDict([('originChainID', 1), ('depositNonce', 1), ('status', 1), ('resourceID', 0x0000000000000000000000000000000000000000000000000000000000000001), ('dataHash', 0xb5635cea5e8144eca50a456350fd741e48
8894f9e06fdf91f328a389ac28685e)])], 'ProposalVote': [OrderedDict([('originChainID', 1), ('depositNonce', 1), ('status', 1), ('resourceID', 0x0000000000000000000000000000000000000000000000000000000000000001)])]}
Calling -> contract_Bridge_polygon.voteProposal(1, 1, bytes32_resourceID2, ERCDepositData2Hash, {'from': relayer2})
Transaction sent: 0x7a90c887283f040169f099dfe57b3bb3e381b31c71690f280e76a8f8546aec07
  Gas price: 0.0 gwei   Gas limit: 800000000   Nonce: 1
  Bridge.voteProposal confirmed   Block: 14199993   Gas used: 93645 (0.01%)

tx3.events -> {'ProposalVote': [OrderedDict([('originChainID', 1), ('depositNonce', 1), ('status', 1), ('resourceID', 0x0000000000000000000000000000000000000000000000000000000000000001)])], 'ProposalEvent': [OrderedDict([('originChainID'
, 1), ('depositNonce', 1), ('status', 2), ('resourceID', 0x0000000000000000000000000000000000000000000000000000000000000001), ('dataHash', 0xb5635cea5e8144eca50a456350fd741e488894f9e06fdf91f328a389ac28685e)])]}

contract_CentrifugeAsset_polygon._assetsStored(eth_abi.encode_single('(bytes32)', [to_bytes(321)])) -> False
Calling -> contract_Bridge_polygon.executeProposal(1, 1, ERCDepositData2, bytes32_resourceID2, {'from': relayer1})
Transaction sent: 0xc12483012ac6b83518dae434fc666e4c90cb038ec1e3ab432f9f4815c393dc01
  Gas price: 0.0 gwei   Gas limit: 800000000   Nonce: 2
  Bridge.executeProposal confirmed   Block: 14199994   Gas used: 71979 (0.01%)

Call trace for '0xc12483012ac6b83518dae434fc666e4c90cb038ec1e3ab432f9f4815c393dc01':
Initial call cost  [22056 gas]
Bridge.executeProposal  0:1348  [19289 / 49923 gas]
├── AccessControl.hasRole  242:305  [111 / 1073 gas]
│   └── EnumerableSet.contains  258:298  [63 / 962 gas]
│       └── EnumerableSet._contains  271:291  [899 gas]
│
└── GenericHandler.executeProposal  [CALL]  642:1254  [5729 / 29561 gas]
    │   ├── address: contract_GenericHandler_polygon.address
    │   ├── value: 0
    │   └── input arguments:
    │       ├── resourceID: 0x0000000000000000000000000000000000000000000000000000000000000001
    │       └── data: 0x00000000000000000000000000000000000000000000000000000000000000200000000000000000000000000000000000000000000000000000000000000141
    │
    └── CentrifugeAsset.store  [CALL]  1118:1219  [23832 gas]
            ├── address: contract_CentrifugeAsset_polygon.address
            ├── value: 0
            └── input arguments:
                └── asset: 0x0000000000000000000000000000000000000000000000000000000000000141


None
tx4.events -> {'AssetStored': [OrderedDict([('asset', 0x0000000000000000000000000000000000000000000000000000000000000141)])], 'ProposalEvent': [OrderedDict([('originChainID', 1), ('depositNonce', 1), ('status', 3), ('resourceID', 0x00000
00000000000000000000000000000000000000000000000000000000001), ('dataHash', 0xb5635cea5e8144eca50a456350fd741e488894f9e06fdf91f328a389ac28685e)])]}

contract_CentrifugeAsset_polygon._assetsStored(eth_abi.encode_single('(bytes32)', [to_bytes(321)])) -> True
```

MANUAL TESTING

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:

ERC20Handler.sol

```
ERC20Handler.constructor(address,bytes32[],address[]).bridgeAddress (contracts/handlers/ERC20Handler.sol#38) lacks a zero-check on :
        - _bridgeAddress = bridgeAddress (contracts/handlers/ERC20Handler.sol#45)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation

Reentrancy in ERC20Handler.deposit(bytes32,uint8,uint64,address,bytes) (contracts/handlers/ERC20Handler.sol#82-122):
        External calls:
        - lockERC20(tokenAddress,depositer,address(this),amount) (contracts/handlers/ERC20Handler.sol#111)
            - (success,returndata) = address(token).call(data) (contracts/ERC20Safe.sol#77)
        State variables written after the call(s):
        - _depositRecords[destinationChainID][depositNonce] = DepositRecord(tokenAddress,uint8(lenRecipientAddress),destinationChainID,resourceID,recipientAddress,depositer,amount) (contracts/handlers/ERC20Handler.sol#113-121)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2

ERC20Handler.deposit(bytes32,uint8,uint64,address,bytes) (contracts/handlers/ERC20Handler.sol#82-122) uses assembly
        - INLINE ASM (contracts/handlers/ERC20Handler.sol#93-106)
ERC20Handler.executeProposal(bytes32,bytes) (contracts/handlers/ERC20Handler.sol#134-164) uses assembly
        - INLINE ASM (contracts/handlers/ERC20Handler.sol#138-151)
        - INLINE ASM (contracts/handlers/ERC20Handler.sol#156-158)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

SafeMath.add(uint256,uint256) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#85-89) is never used and should be removed
SafeMath.div(uint256,uint256) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#135-138) is never used and should be removed
SafeMath.div(uint256,uint256,string) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#190-193) is never used and should be removed
SafeMath.mod(uint256,uint256) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#152-155) is never used and should be removed
SafeMath.mod(uint256,uint256,string) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#210-213) is never used and should be removed
SafeMath.mul(uint256,uint256) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#116-121) is never used and should be removed
SafeMath.sub(uint256,uint256) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#101-104) is never used and should be removed
SafeMath.sub(uint256,uint256,string) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#170-173) is never used and should be removed
SafeMath.tryAdd(uint256,uint256) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#24-28) is never used and should be removed
SafeMath.tryDiv(uint256,uint256) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#60-63) is never used and should be removed
SafeMath.tryMod(uint256,uint256) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#70-73) is never used and should be removed
SafeMath.tryMul(uint256,uint256) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#45-53) is never used and should be removed
SafeMath.trySub(uint256,uint256) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#35-38) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/math/SafeMath.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (contracts/ERC20Safe.sol#1) is too complex
Pragma version>=0.6.0<0.8.0 (contracts/handlers/ERC20Handler.sol#1) is too complex
Pragma version>=0.6.0<0.8.0 (contracts/handlers/HandlerHelpers.sol#1) is too complex
Pragma version>=0.6.0<0.8.0 (contracts/interfaces/IDepositExecute.sol#1) is too complex
Pragma version>=0.6.0<0.8.0 (contracts/interfaces/IERCHandler.sol#1) is too complex
solc-0.6.6 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in ERC20Safe._safeCall(IERC20,bytes) (contracts/ERC20Safe.sol#76-84):
        - (success,returndata) = address(token).call(data) (contracts/ERC20Safe.sol#77)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Variable ERC20Handler._depositRecords (contracts/handlers/ERC20Handler.sol#25) is not in mixedCase
Variable HandlerHelpers._bridgeAddress (contracts/handlers/HandlerHelpers.sol#11) is not in mixedCase
Variable HandlerHelpers._resourceIDToTokenContractAddress (contracts/handlers/HandlerHelpers.sol#14) is not in mixedCase
Variable HandlerHelpers._tokenContractAddressToResourceID (contracts/handlers/HandlerHelpers.sol#17) is not in mixedCase
Variable HandlerHelpers._contractWhitelist (contracts/handlers/HandlerHelpers.sol#20) is not in mixedCase
Variable HandlerHelpers._burnList (contracts/handlers/HandlerHelpers.sol#23) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

fundERC20(address,address,uint256) should be declared external:
        - ERC20Safe.fundERC20(address,address,uint256) (contracts/ERC20Safe.sol#20-23)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

ERC721Handler.sol

```
ERC721._mint(address,uint256) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#333-344) ignores return value by _holderTokens[to].add(tokenId) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#339)
ERC721._mint(address,uint256) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#333-344) ignores return value by _tokenOwners.set(tokenId,to) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#341)
ERC721._burn(uint256) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#356-374) ignores return value by _holderTokens[owner].remove(tokenId) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#369)
ERC721._burn(uint256) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#356-374) ignores return value by _tokenOwners.remove(tokenId) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#371)
ERC721._transfer(address,address,uint256) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#387-402) ignores return value by _holderTokens[from].remove(tokenId) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#396)
ERC721._transfer(address,address,uint256) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#387-402) ignores return value by _holderTokens[to].add(tokenId) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#397)
ERC721._transfer(address,address,uint256) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#387-402) ignores return value by _tokenOwners.set(tokenId,to) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#399)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

ERC721MinterBurnerPauser.constructor(string,string,string).name (contracts/ERC721MinterBurnerPauser.sol#24) shadows:
        - ERC721.name() (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#121-123) (function)
        - IERC721Metadata.name() (node_modules/@openzeppelin/contracts/token/ERC721/IERC721Metadata.sol#16) (function)
ERC721MinterBurnerPauser.constructor(string,string,string).symbol (contracts/ERC721MinterBurnerPauser.sol#24) shadows:
        - ERC721.symbol() (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#128-130) (function)
        - IERC721Metadata.symbol() (node_modules/@openzeppelin/contracts/token/ERC721/IERC721Metadata.sol#21) (function)
ERC721MinterBurnerPauser.constructor(string,string,string).baseURI (contracts/ERC721MinterBurnerPauser.sol#24) shadows:
        - ERC721.baseURI() (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#158-160) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

ERC721Handler.constructor(address,bytes32[],address[],address[]).bridgeAddress (contracts/handlers/ERC721Handler.sol#49) lacks a zero-check on :
        - _bridgeAddress = bridgeAddress (contracts/handlers/ERC721Handler.sol#57)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation
```

```
Reentrancy in ERC721Handler.deposit(bytes32,uint8,uint64,address,bytes) (contracts/handlers/ERC721Handler.sol#101-159):
        External calls:
        - tokenAddress.supportsInterface(_INTERFACE_ERC721_METADATA) (contracts/handlers/ERC721Handler.sol#138)
        - burnERC721(tokenAddress,tokenID) (contracts/handlers/ERC721Handler.sol#144)
                - erc721.burn(tokenID) (contracts/ERC721Safe.sol#70)
        - lockERC721(tokenAddress,depositer,address(this),tokenID) (contracts/handlers/ERC721Handler.sol#146)
                - erc721.transferFrom(owner,recipient,tokenID) (contracts/ERC721Safe.sol#35)
        State variables written after the call(s):
        - _depositRecords[destinationChainID][depositNonce] = DepositRecord(tokenAddress,uint8(lenDestinationRecipientAddress),uint8(destinationChainID),resourceID,destinationRecipientAddress,depositer,tokenID,metaData) (contracts/handle
rs/ERC721Handler.sol#148-158)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2

Address.isContract(address) (node_modules/@openzeppelin/contracts/utils/Address.sol#26-35) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/utils/Address.sol#33)
Address._verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#171-188) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/utils/Address.sol#180-183)
ERC721Handler.deposit(bytes32,uint8,uint64,address,bytes) (contracts/handlers/ERC721Handler.sol#101-159) uses assembly
        - INLINE ASM (contracts/handlers/ERC721Handler.sol#112-132)
ERC721Handler.executeProposal(bytes32,bytes) (contracts/handlers/ERC721Handler.sol#173-231) uses assembly
        - INLINE ASM (contracts/handlers/ERC721Handler.sol#178-215)
        - INLINE ASM (contracts/handlers/ERC721Handler.sol#219-221)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Different versions of Solidity is used:
        - Version used: ['>=0.6.0<0.8.0', '>=0.6.2<0.8.0']
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/GSN/Context.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/access/AccessControl.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/introspection/ERC165.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts/introspection/ERC165Checker.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/introspection/IERC165.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/math/SafeMath.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/ERC721Burnable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/ERC721Pausable.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/IERC721.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/IERC721Enumerable.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/IERC721Metadata.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/IERC721Receiver.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts/utils/Address.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/Counters.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/EnumerableMap.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/EnumerableSet.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/Pausable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/Strings.sol#3)
        - >=0.6.0<0.8.0 (contracts/ERC721MinterBurnerPauser.sol#1)
        - >=0.6.0<0.8.0 (contracts/ERC721Safe.sol#1)
        - >=0.6.0<0.8.0 (contracts/handlers/ERC721Handler.sol#1)
        - ABIEncoderV2 (contracts/handlers/ERC721Handler.sol#2)
        - >=0.6.0<0.8.0 (contracts/handlers/HandlerHelpers.sol#1)
        - >=0.6.0<0.8.0 (contracts/interfaces/IDepositExecute.sol#1)
        - >=0.6.0<0.8.0 (contracts/interfaces/IERCHandler.sol#1)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

AccessControl._setRoleAdmin(bytes32,bytes32) (node_modules/@openzeppelin/contracts/access/AccessControl.sol#201-204) is never used and should be removed
Address.functionCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#75-81) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#104-106) is never used and should be removed
Address.functionDelegateCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#153-155) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#163-169) is never used and should be removed
Address.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#129-131) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#139-145) is never used and should be removed
Address.sendValue(address,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#53-59) is never used and should be removed

Context._msgData() (node_modules/@openzeppelin/contracts/utils/Context.sol#20-23) is never used and should be removed
Counters.current(Counters.Counter) (node_modules/@openzeppelin/contracts/utils/Counters.sol#28-30) is never used and should be removed
Counters.decrement(Counters.Counter) (node_modules/@openzeppelin/contracts/utils/Counters.sol#37-39) is never used and should be removed
Counters.increment(Counters.Counter) (node_modules/@openzeppelin/contracts/utils/Counters.sol#32-35) is never used and should be removed
ERC165Checker.getSupportedInterfaces(address,bytes4[]) (node_modules/@openzeppelin/contracts/introspection/ERC165Checker.sol#53-66) is never used and should be removed
ERC165Checker.supportsAllInterfaces(address,bytes4[]) (node_modules/@openzeppelin/contracts/introspection/ERC165Checker.sol#77-92) is never used and should be removed
ERC721._safeMint(address,uint256) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#308-310) is never used and should be removed
ERC721._safeMint(address,uint256,bytes) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#316-319) is never used and should be removed
EnumerableMap._get(EnumerableMap.Map,bytes32) (node_modules/@openzeppelin/contracts/utils/EnumerableMap.sol#163-167) is never used and should be removed
EnumerableMap._tryGet(EnumerableMap.Map,bytes32) (node_modules/@openzeppelin/contracts/utils/EnumerableMap.sol#150-154) is never used and should be removed
EnumerableMap.get(EnumerableMap.UintToAddressMap,uint256) (node_modules/@openzeppelin/contracts/utils/EnumerableMap.sol#253-255) is never used and should be removed
EnumerableMap.tryGet(EnumerableMap.UintToAddressMap,uint256) (node_modules/@openzeppelin/contracts/utils/EnumerableMap.sol#241-244) is never used and should be removed
EnumerableSet.add(EnumerableSet.Bytes32Set,bytes32) (node_modules/@openzeppelin/contracts/utils/EnumerableSet.sol#147-149) is never used and should be removed
EnumerableSet.at(EnumerableSet.Bytes32Set,uint256) (node_modules/@openzeppelin/contracts/utils/EnumerableSet.sol#185-187) is never used and should be removed
EnumerableSet.contains(EnumerableSet.Bytes32Set,bytes32) (node_modules/@openzeppelin/contracts/utils/EnumerableSet.sol#164-166) is never used and should be removed
EnumerableSet.contains(EnumerableSet.UintSet,uint256) (node_modules/@openzeppelin/contracts/utils/EnumerableSet.sol#273-275) is never used and should be removed
EnumerableSet.length(EnumerableSet.Bytes32Set) (node_modules/@openzeppelin/contracts/utils/EnumerableSet.sol#171-173) is never used and should be removed
EnumerableSet.remove(EnumerableSet.Bytes32Set,bytes32) (node_modules/@openzeppelin/contracts/utils/EnumerableSet.sol#157-159) is never used and should be removed
SafeMath.add(uint256,uint256) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#85-88) is never used and should be removed
SafeMath.div(uint256,uint256) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#135-138) is never used and should be removed
SafeMath.div(uint256,uint256,string) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#190-193) is never used and should be removed
SafeMath.mod(uint256,uint256) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#152-155) is never used and should be removed
SafeMath.mod(uint256,uint256,string) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#210-213) is never used and should be removed
SafeMath.mul(uint256,uint256) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#116-121) is never used and should be removed
SafeMath.sub(uint256,uint256) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#101-104) is never used and should be removed
SafeMath.sub(uint256,uint256,string) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#170-173) is never used and should be removed
SafeMath.tryAdd(uint256,uint256) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#24-28) is never used and should be removed
SafeMath.tryDiv(uint256,uint256) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#60-63) is never used and should be removed
SafeMath.tryMod(uint256,uint256) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#70-73) is never used and should be removed
SafeMath.tryMul(uint256,uint256) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#45-53) is never used and should be removed
SafeMath.trySub(uint256,uint256) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#35-38) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/GSN/Context.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/access/AccessControl.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/introspection/ERC165.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts/introspection/ERC165Checker.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/introspection/IERC165.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/math/SafeMath.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/ERC721Burnable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/ERC721Pausable.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/IERC721.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/IERC721Enumerable.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/IERC721Metadata.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/IERC721Receiver.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts/utils/Address.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/Counters.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/EnumerableMap.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/EnumerableSet.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/Pausable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/Strings.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (contracts/ERC721MinterBurnerPauser.sol#1) is too complex
Pragma version>=0.6.0<0.8.0 (contracts/ERC721Safe.sol#1) is too complex
Pragma version>=0.6.0<0.8.0 (contracts/handlers/ERC721Handler.sol#1) is too complex
Pragma version>=0.6.0<0.8.0 (contracts/handlers/HandlerHelpers.sol#1) is too complex
Pragma version>=0.6.0<0.8.0 (contracts/interfaces/IDepositExecute.sol#1) is too complex
Pragma version>=0.6.0<0.8.0 (contracts/interfaces/IERCHandler.sol#1) is too complex
solc-0.6.6 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in ERC165Checker._callERC165SupportsInterface(address,bytes4) (node_modules/@openzeppelin/contracts/introspection/ERC165Checker.sol#121-130):
        - (success,result) = account.staticcall{gas: 30000}(encodedParams) (node_modules/@openzeppelin/contracts/introspection/ERC165Checker.sol#127)
Low level call in Address.sendValue(address,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#53-59):
        - (success) = recipient.call{value: amount}() (node_modules/@openzeppelin/contracts/utils/Address.sol#57)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#114-121):
        - (success,returndata) = target.call{value: value}(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#119)
Low level call in Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#139-145):
        - (success,returndata) = target.staticcall(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#143)
Low level call in Address.functionDelegateCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#163-169):
        - (success,returndata) = target.delegatecall(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#167)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Parameter ERC721.safeTransferFrom(address,address,uint256,bytes)._data (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#245) is not in mixedCase
Parameter ERC721MinterBurnerPauser.mint(address,uint256,string)._data (contracts/ERC721MinterBurnerPauser.sol#44) is not in mixedCase
Variable ERC721Handler._depositRecords (contracts/handlers/ERC721Handler.sol#34) is not in mixedCase
Variable HandlerHelpers._bridgeAddress (contracts/handlers/HandlerHelpers.sol#11) is not in mixedCase
Variable HandlerHelpers._resourceIDToTokenContractAddress (contracts/handlers/HandlerHelpers.sol#14) is not in mixedCase
Variable HandlerHelpers._tokenContractAddressToResourceID (contracts/handlers/HandlerHelpers.sol#17) is not in mixedCase
Variable HandlerHelpers._contractWhitelist (contracts/handlers/HandlerHelpers.sol#20) is not in mixedCase
Variable HandlerHelpers._burnList (contracts/handlers/HandlerHelpers.sol#23) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (node_modules/@openzeppelin/contracts/utils/Context.sol#21)" inContext (node_modules/@openzeppelin/contracts/utils/Context.sol#15-24)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

getRoleMemberCount(bytes32) should be declared external:
        - AccessControl.getRoleMemberCount(bytes32) (node_modules/@openzeppelin/contracts/access/AccessControl.sol#95-97)
getRoleMember(bytes32,uint256) should be declared external:
        - AccessControl.getRoleMember(bytes32,uint256) (node_modules/@openzeppelin/contracts/access/AccessControl.sol#111-113)
getRoleAdmin(bytes32) should be declared external:
        - AccessControl.getRoleAdmin(bytes32) (node_modules/@openzeppelin/contracts/access/AccessControl.sol#121-123)
grantRole(bytes32,address) should be declared external:
        - AccessControl.grantRole(bytes32,address) (node_modules/@openzeppelin/contracts/access/AccessControl.sol#135-139)
revokeRole(bytes32,address) should be declared external:
        - AccessControl.revokeRole(bytes32,address) (node_modules/@openzeppelin/contracts/access/AccessControl.sol#150-154)
renounceRole(bytes32,address) should be declared external:
        - AccessControl.renounceRole(bytes32,address) (node_modules/@openzeppelin/contracts/access/AccessControl.sol#170-174)
supportsInterface(bytes4) should be declared external:
        - ERC165.supportsInterface(bytes4) (node_modules/@openzeppelin/contracts/introspection/ERC165.sol#35-37)
balanceOf(address) should be declared external:
        - ERC721.balanceOf(address) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#106-109)
name() should be declared external:
        - ERC721.name() (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#121-123)
symbol() should be declared external:
        - ERC721.symbol() (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#128-130)
tokenURI(uint256) should be declared external:
        - ERC721.tokenURI(uint256) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#135-151)
tokenOfOwnerByIndex(address,uint256) should be declared external:
        - ERC721.tokenOfOwnerByIndex(address,uint256) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#165-167)
totalSupply() should be declared external:
        - ERC721.totalSupply() (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#172-175)
```

AUTOMATED TESTING

```
tokenByIndex(uint256) should be declared external:
        - ERC721.tokenByIndex(uint256) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#180-183)
approve(address,uint256) should be declared external:
        - ERC721.approve(address,uint256) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#188-197)
setApprovalForAll(address,bool) should be declared external:
        - ERC721.setApprovalForAll(address,bool) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#211-216)
transferFrom(address,address,uint256) should be declared external:
        - ERC721.transferFrom(address,address,uint256) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#228-233)
safeTransferFrom(address,address,uint256) should be declared external:
        - ERC721.safeTransferFrom(address,address,uint256) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#238-240)
burn(uint256) should be declared external:
        - ERC721Burnable.burn(uint256) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721Burnable.sol#20-24)
mint(address,uint256,string) should be declared external:
        - ERC721MinterBurnerPauser.mint(address,uint256,string) (contracts/ERC721MinterBurnerPauser.sol#44-49)
pause() should be declared external:
        - ERC721MinterBurnerPauser.pause() (contracts/ERC721MinterBurnerPauser.sol#60-63)
unpause() should be declared external:
        - ERC721MinterBurnerPauser.unpause() (contracts/ERC721MinterBurnerPauser.sol#74-77)
fundERC721(address,address,uint256) should be declared external:
        - ERC721Safe.fundERC721(address,address,uint256) (contracts/ERC721Safe.sol#21-24)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

GenericHandler.sol

```
GenericHandler.constructor(address,bytes32[],address[],bytes4[],bytes4[]).bridgeAddress (contracts/handlers/GenericHandler.sol#66) lacks a zero-check on :
        - _bridgeAddress = bridgeAddress (contracts/handlers/GenericHandler.sol#81)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation

Reentrancy in GenericHandler.deposit(bytes32,uint8,uint64,address,bytes) (contracts/handlers/GenericHandler.sol#141-179):
        External calls:
        - (success) = contractAddress.call(callData) (contracts/handlers/GenericHandler.sol#169)
        State variables written after the call(s):
        - _depositRecords[destinationChainID][depositNonce] = DepositRecord(destinationChainID,depositer,resourceID,metadata) (contracts/handlers/GenericHandler.sol#173-178)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2

GenericHandler.deposit(bytes32,uint8,uint64,address,bytes) (contracts/handlers/GenericHandler.sol#141-179) uses assembly
        - INLINE ASM (contracts/handlers/GenericHandler.sol#145-161)
GenericHandler.executeProposal(bytes32,bytes) (contracts/handlers/GenericHandler.sol#191-219) uses assembly
        - INLINE ASM (contracts/handlers/GenericHandler.sol#193-208)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Pragma version>=0.6.0<0.8.0 (contracts/handlers/GenericHandler.sol#1) is too complex
Pragma version>=0.6.0<0.8.0 (contracts/interfaces/IGenericHandler.sol#1) is too complex
solc-0.6.6 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in GenericHandler.deposit(bytes32,uint8,uint64,address,bytes) (contracts/handlers/GenericHandler.sol#141-179):
        - (success) = contractAddress.call(callData) (contracts/handlers/GenericHandler.sol#169)
Low level call in GenericHandler.executeProposal(bytes32,bytes) (contracts/handlers/GenericHandler.sol#191-219):
        - (success) = contractAddress.call(callData) (contracts/handlers/GenericHandler.sol#216)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Variable GenericHandler._bridgeAddress (contracts/handlers/GenericHandler.sol#12) is not in mixedCase
Variable GenericHandler._depositRecords (contracts/handlers/GenericHandler.sol#22) is not in mixedCase
Variable GenericHandler._resourceIDToContractAddress (contracts/handlers/GenericHandler.sol#25) is not in mixedCase
Variable GenericHandler._contractAddressToResourceID (contracts/handlers/GenericHandler.sol#28) is not in mixedCase
Variable GenericHandler._contractAddressToDepositFunctionSignature (contracts/handlers/GenericHandler.sol#31) is not in mixedCase
Variable GenericHandler._contractAddressToExecuteFunctionSignature (contracts/handlers/GenericHandler.sol#34) is not in mixedCase
Variable GenericHandler._contractWhitelist (contracts/handlers/GenericHandler.sol#37) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
```

HandlerHelpers.sol

```
HandlerHelpers._bridgeAddress (contracts/handlers/HandlerHelpers.sol#11) is never initialized. It is used in:
        - HandlerHelpers._onlyBridge() (contracts/handlers/HandlerHelpers.sol#30-32)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-state-variables

Pragma version>=0.6.0<0.8.0 (contracts/handlers/HandlerHelpers.sol#1) is too complex
Pragma version>=0.6.0<0.8.0 (contracts/interfaces/IERCHandler.sol#1) is too complex
solc-0.6.6 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Variable HandlerHelpers._bridgeAddress (contracts/handlers/HandlerHelpers.sol#11) is not in mixedCase
Variable HandlerHelpers._resourceIDToTokenContractAddress (contracts/handlers/HandlerHelpers.sol#14) is not in mixedCase
Variable HandlerHelpers._tokenContractAddressToResourceID (contracts/handlers/HandlerHelpers.sol#17) is not in mixedCase
Variable HandlerHelpers._contractWhitelist (contracts/handlers/HandlerHelpers.sol#20) is not in mixedCase
Variable HandlerHelpers._burnList (contracts/handlers/HandlerHelpers.sol#23) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

HandlerHelpers._bridgeAddress (contracts/handlers/HandlerHelpers.sol#11) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant
```

Whitelist.sol

```
Address.isContract(address) (node_modules/@openzeppelin/contracts/utils/Address.sol#26-35) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/utils/Address.sol#33)
Address._verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#171-188) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/utils/Address.sol#180-183)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Whitelist.isWhitelisted(address) (contracts/utils/Whitelist.sol#123-125) compares to a boolean constant:
        -_whitelist[addressToCheck] == true (contracts/utils/Whitelist.sol#124)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#boolean-equality

Different versions of Solidity is used:
        - Version used: ['>=0.6.0<0.8.0', '>=0.6.2<0.8.0']
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/access/AccessControl.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts/utils/Address.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/EnumerableSet.sol#3)
        - >=0.6.0<0.8.0 (contracts/utils/Whitelist.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

AccessControl._setRoleAdmin(bytes32,bytes32) (node_modules/@openzeppelin/contracts/access/AccessControl.sol#201-204) is never used and should be removed
AccessControl._setupRole(bytes32,address) (node_modules/@openzeppelin/contracts/access/AccessControl.sol#192-194) is never used and should be removed
Address._verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#171-188) is never used and should be removed
Address.functionCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#79-81) is never used and should be removed
Address.functionCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#89-91) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#104-106) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#114-121) is never used and should be removed
Address.functionDelegateCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#153-155) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#163-169) is never used and should be removed
Address.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#129-131) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#139-145) is never used and should be removed
Address.isContract(address) (node_modules/@openzeppelin/contracts/utils/Address.sol#26-35) is never used and should be removed
Address.sendValue(address,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#53-59) is never used and should be removed
Context._msgData() (node_modules/@openzeppelin/contracts/utils/Context.sol#20-23) is never used and should be removed
EnumerableSet.add(EnumerableSet.Bytes32Set,bytes32) (node_modules/@openzeppelin/contracts/utils/EnumerableSet.sol#147-149) is never used and should be removed
EnumerableSet.add(EnumerableSet.UintSet,uint256) (node_modules/@openzeppelin/contracts/utils/EnumerableSet.sol#256-258) is never used and should be removed
EnumerableSet.at(EnumerableSet.Bytes32Set,uint256) (node_modules/@openzeppelin/contracts/utils/EnumerableSet.sol#185-187) is never used and should be removed
EnumerableSet.at(EnumerableSet.UintSet,uint256) (node_modules/@openzeppelin/contracts/utils/EnumerableSet.sol#294-296) is never used and should be removed
EnumerableSet.contains(EnumerableSet.Bytes32Set,bytes32) (node_modules/@openzeppelin/contracts/utils/EnumerableSet.sol#164-166) is never used and should be removed
EnumerableSet.contains(EnumerableSet.UintSet,uint256) (node_modules/@openzeppelin/contracts/utils/EnumerableSet.sol#273-275) is never used and should be removed
EnumerableSet.length(EnumerableSet.Bytes32Set) (node_modules/@openzeppelin/contracts/utils/EnumerableSet.sol#171-173) is never used and should be removed
EnumerableSet.length(EnumerableSet.UintSet) (node_modules/@openzeppelin/contracts/utils/EnumerableSet.sol#280-282) is never used and should be removed
EnumerableSet.remove(EnumerableSet.Bytes32Set,bytes32) (node_modules/@openzeppelin/contracts/utils/EnumerableSet.sol#157-159) is never used and should be removed
EnumerableSet.remove(EnumerableSet.UintSet,uint256) (node_modules/@openzeppelin/contracts/utils/EnumerableSet.sol#266-268) is never used and should be removed
Whitelist._onlyWhitelisted() (contracts/utils/Whitelist.sol#135-138) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/access/AccessControl.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts/utils/Address.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/EnumerableSet.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (contracts/utils/Whitelist.sol#3) is too complex
solc-0.6.6 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#53-59):
        - (success) = recipient.call{value: amount}() (node_modules/@openzeppelin/contracts/utils/Address.sol#57)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#114-121):
        - (success,returndata) = target.call{value: value}(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#119)
Low level call in Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#139-145):
        - (success,returndata) = target.staticcall(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#143)
Low level call in Address.functionDelegateCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#163-169):
        - (success,returndata) = target.delegatecall(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#167)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Redundant expression "this (node_modules/@openzeppelin/contracts/utils/Context.sol#21)" inContext (node_modules/@openzeppelin/contracts/utils/Context.sol#15-24)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

getRoleMemberCount(bytes32) should be declared external:
        - AccessControl.getRoleMemberCount(bytes32) (node_modules/@openzeppelin/contracts/access/AccessControl.sol#95-97)
getRoleMember(bytes32,uint256) should be declared external:
```

```
        - AccessControl.getRoleMember(bytes32,uint256) (node_modules/@openzeppelin/contracts/access/AccessControl.sol#111-113)
getRoleAdmin(bytes32) should be declared external:
        - AccessControl.getRoleAdmin(bytes32) (node_modules/@openzeppelin/contracts/access/AccessControl.sol#121-123)
grantRole(bytes32,address) should be declared external:
        - AccessControl.grantRole(bytes32,address) (node_modules/@openzeppelin/contracts/access/AccessControl.sol#135-139)
revokeRole(bytes32,address) should be declared external:
        - AccessControl.revokeRole(bytes32,address) (node_modules/@openzeppelin/contracts/access/AccessControl.sol#150-154)
renounceRole(bytes32,address) should be declared external:
        - AccessControl.renounceRole(bytes32,address) (node_modules/@openzeppelin/contracts/access/AccessControl.sol#170-174)
enableWhitelist() should be declared external:
        - Whitelist.enableWhitelist() (contracts/utils/Whitelist.sol#69-73)
disableWhitelist() should be declared external:
        - Whitelist.disableWhitelist() (contracts/utils/Whitelist.sol#80-84)
addToWhitelist(address) should be declared external:
        - Whitelist.addToWhitelist(address) (contracts/utils/Whitelist.sol#100-104)
removeFromWhitelist(address) should be declared external:
        - Whitelist.removeFromWhitelist(address) (contracts/utils/Whitelist.sol#112-116)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

Bridge.sol

```
Bridge.constructor(uint8,address[],uint256,uint256,uint256).i (contracts/Bridge.sol#120) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

Bridge.transferFunds(address[],uint256[]) (contracts/Bridge.sol#427-431) has external calls inside a loop: addrs[i].transfer(amounts[i]) (contracts/Bridge.sol#429)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation/#calls-inside-a-loop

Reentrancy in Bridge.deposit(uint8,bytes32,bytes) (contracts/Bridge.sol#294-307):
        External calls:
        - depositHandler.deposit(resourceID,destinationChainID,depositNonce,msg.sender,data) (contracts/Bridge.sol#304)
        Event emitted after the call(s):
        - Deposit(destinationChainID,resourceID,depositNonce) (contracts/Bridge.sol#306)
Reentrancy in Bridge.executeProposal(uint8,uint64,bytes,bytes32) (contracts/Bridge.sol#403-419):
        External calls:
        - depositHandler.executeProposal(proposal._resourceID,data) (contracts/Bridge.sol#416)
        Event emitted after the call(s):
        - ProposalEvent(chainID,depositNonce,proposal._status,proposal._resourceID,proposal._dataHash) (contracts/Bridge.sol#418)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

Address.isContract(address) (node_modules/@openzeppelin/contracts/utils/Address.sol#26-35) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/utils/Address.sol#33)
Address._verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#171-188) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/utils/Address.sol#180-183)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Whitelist.isWhitelisted(address) (contracts/utils/Whitelist.sol#123-125) compares to a boolean constant:
        -_whitelist[addressToCheck] == true (contracts/utils/Whitelist.sol#124)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#boolean-equality

Different versions of Solidity is used:
        - Version used: ['>=0.6.0<0.8.0', '>=0.6.2<0.8.0']
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/access/AccessControl.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts/utils/Address.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/EnumerableSet.sol#3)
        - >=0.6.0<0.8.0 (contracts/Bridge.sol#1)
        - ABIEncoderV2 (contracts/Bridge.sol#2)
        - >=0.6.0<0.8.0 (contracts/interfaces/IBridge.sol#1)
        - >=0.6.0<0.8.0 (contracts/interfaces/IDepositExecute.sol#1)
        - >=0.6.0<0.8.0 (contracts/interfaces/IERCHandler.sol#1)
        - >=0.6.0<0.8.0 (contracts/interfaces/IGenericHandler.sol#1)
        - >=0.6.0<0.8.0 (contracts/utils/Pausable.sol#3)
        - >=0.6.0<0.8.0 (contracts/utils/SafeMath.sol#3)
        - >=0.6.0<0.8.0 (contracts/utils/Whitelist.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

Bridge.constructor(uint8,address[],uint256,uint256,uint256) (contracts/Bridge.sol#110-125) has costly operations inside a loop:
        - _totalRelayers ++ (contracts/Bridge.sol#122)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#costly-operations-inside-a-loop

Address._verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#171-188) is never used and should be removed
Address.functionCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#78-81) is never used and should be removed
Address.functionCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#88-91) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#104-106) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#114-121) is never used and should be removed
Address.functionDelegateCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#153-155) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#163-169) is never used and should be removed
Address.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#129-131) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#139-145) is never used and should be removed
Address.isContract(address) (node_modules/@openzeppelin/contracts/utils/Address.sol#26-35) is never used and should be removed
Address.sendValue(address,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#53-59) is never used and should be removed
Context._msgData() (node_modules/@openzeppelin/contracts/utils/Context.sol#20-23) is never used and should be removed
EnumerableSet.add(EnumerableSet.Bytes32Set,bytes32) (node_modules/@openzeppelin/contracts/utils/EnumerableSet.sol#147-149) is never used and should be removed
EnumerableSet.add(EnumerableSet.UintSet,uint256) (node_modules/@openzeppelin/contracts/utils/EnumerableSet.sol#256-258) is never used and should be removed
EnumerableSet.at(EnumerableSet.Bytes32Set,uint256) (node_modules/@openzeppelin/contracts/utils/EnumerableSet.sol#185-187) is never used and should be removed
EnumerableSet.at(EnumerableSet.UintSet,uint256) (node_modules/@openzeppelin/contracts/utils/EnumerableSet.sol#294-296) is never used and should be removed
EnumerableSet.contains(EnumerableSet.Bytes32Set,bytes32) (node_modules/@openzeppelin/contracts/utils/EnumerableSet.sol#164-166) is never used and should be removed
EnumerableSet.contains(EnumerableSet.UintSet,uint256) (node_modules/@openzeppelin/contracts/utils/EnumerableSet.sol#273-275) is never used and should be removed
EnumerableSet.length(EnumerableSet.Bytes32Set) (node_modules/@openzeppelin/contracts/utils/EnumerableSet.sol#171-173) is never used and should be removed
EnumerableSet.length(EnumerableSet.UintSet) (node_modules/@openzeppelin/contracts/utils/EnumerableSet.sol#280-282) is never used and should be removed
EnumerableSet.remove(EnumerableSet.Bytes32Set,bytes32) (node_modules/@openzeppelin/contracts/utils/EnumerableSet.sol#157-159) is never used and should be removed
EnumerableSet.remove(EnumerableSet.UintSet,uint256) (node_modules/@openzeppelin/contracts/utils/EnumerableSet.sol#266-268) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/access/AccessControl.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts/utils/Address.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/EnumerableSet.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (contracts/Bridge.sol#1) is too complex
Pragma version>=0.6.0<0.8.0 (contracts/interfaces/IBridge.sol#1) is too complex
Pragma version>=0.6.0<0.8.0 (contracts/interfaces/IDepositExecute.sol#1) is too complex
Pragma version>=0.6.0<0.8.0 (contracts/interfaces/IERCHandler.sol#1) is too complex
Pragma version>=0.6.0<0.8.0 (contracts/interfaces/IGenericHandler.sol#1) is too complex
Pragma version>=0.6.0<0.8.0 (contracts/utils/Pausable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (contracts/utils/SafeMath.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (contracts/utils/Whitelist.sol#3) is too complex
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#53-59):
        - (success) = recipient.call{value: amount}() (node_modules/@openzeppelin/contracts/utils/Address.sol#57)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#114-121):
        - (success,returndata) = target.call{value: value}(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#119)
Low level call in Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#139-145):
        - (success,returndata) = target.staticcall(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#143)
Low level call in Address.functionDelegateCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#163-169):
        - (success,returndata) = target.delegatecall(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#167)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Variable Bridge._chainID (contracts/Bridge.sol#19) is not in mixedCase
Variable Bridge._relayerThreshold (contracts/Bridge.sol#20) is not in mixedCase
Variable Bridge._totalRelayers (contracts/Bridge.sol#21) is not in mixedCase
Variable Bridge._totalProposals (contracts/Bridge.sol#22) is not in mixedCase
Variable Bridge._fee (contracts/Bridge.sol#23) is not in mixedCase
Variable Bridge._expiry (contracts/Bridge.sol#24) is not in mixedCase
Variable Bridge._depositCounts (contracts/Bridge.sol#40) is not in mixedCase
Variable Bridge._resourceIDToHandlerAddress (contracts/Bridge.sol#42) is not in mixedCase
Variable Bridge._depositRecords (contracts/Bridge.sol#44) is not in mixedCase
Variable Bridge._proposals (contracts/Bridge.sol#46) is not in mixedCase
Variable Bridge._hasVotedOnProposal (contracts/Bridge.sol#48) is not in mixedCase
Function IBridge._chainID() (contracts/interfaces/IBridge.sol#12) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (node_modules/@openzeppelin/contracts/utils/Context.sol#21)" inContext (node_modules/@openzeppelin/contracts/utils/Context.sol#15-24)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

getRoleMemberCount(bytes32) should be declared external:
        - AccessControl.getRoleMemberCount(bytes32) (node_modules/@openzeppelin/contracts/access/AccessControl.sol#95-97)
getRoleMember(bytes32,uint256) should be declared external:
        - AccessControl.getRoleMember(bytes32,uint256) (node_modules/@openzeppelin/contracts/access/AccessControl.sol#111-113)
getRoleAdmin(bytes32) should be declared external:
        - AccessControl.getRoleAdmin(bytes32) (node_modules/@openzeppelin/contracts/access/AccessControl.sol#121-123)
cancelProposal(uint8,uint64,bytes32) should be declared external:
        - Bridge.cancelProposal(uint8,uint64,bytes32) (contracts/Bridge.sol#380-390)
paused() should be declared external:
        - Pausable.paused() (contracts/utils/Pausable.sol#37-39)
enableWhitelist() should be declared external:
        - Whitelist.enableWhitelist() (contracts/utils/Whitelist.sol#69-73)
disableWhitelist() should be declared external:
        - Whitelist.disableWhitelist() (contracts/utils/Whitelist.sol#80-84)
removeFromWhitelist(address) should be declared external:
        - Whitelist.removeFromWhitelist(address) (contracts/utils/Whitelist.sol#112-116)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

ERC20Safe.sol

```
ERC20Safe._safeTransfer(IERC20,address,uint256) (contracts/ERC20Safe.sol#55-57) is never used and should be removed
ERC20Safe.lockERC20(address,address,address,uint256) (contracts/ERC20Safe.sol#32-35) is never used and should be removed
ERC20Safe.releaseERC20(address,address,uint256) (contracts/ERC20Safe.sol#43-46) is never used and should be removed
SafeMath.add(uint256,uint256) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#85-89) is never used and should be removed
SafeMath.div(uint256,uint256) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#135-138) is never used and should be removed
SafeMath.div(uint256,uint256,string) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#190-193) is never used and should be removed
SafeMath.mod(uint256,uint256) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#152-155) is never used and should be removed
SafeMath.mod(uint256,uint256,string) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#210-213) is never used and should be removed
SafeMath.mul(uint256,uint256) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#116-121) is never used and should be removed
SafeMath.sub(uint256,uint256) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#101-104) is never used and should be removed
SafeMath.sub(uint256,uint256,string) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#170-173) is never used and should be removed
SafeMath.tryAdd(uint256,uint256) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#24-28) is never used and should be removed
SafeMath.tryDiv(uint256,uint256) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#60-63) is never used and should be removed
SafeMath.tryMod(uint256,uint256) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#70-73) is never used and should be removed
SafeMath.tryMul(uint256,uint256) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#45-53) is never used and should be removed
SafeMath.trySub(uint256,uint256) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#35-38) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/math/SafeMath.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (contracts/ERC20Safe.sol#1) is too complex
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in ERC20Safe._safeCall(IERC20,bytes) (contracts/ERC20Safe.sol#76-84):
        - (success,returndata) = address(token).call(data) (contracts/ERC20Safe.sol#77)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

fundERC20(address,address,uint256) should be declared external:
        - ERC20Safe.fundERC20(address,address,uint256) (contracts/ERC20Safe.sol#20-23)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

ERC721MinterBurnerPauser.sol

```
ERC721._mint(address,uint256) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#333-344) ignores return value by _holderTokens[to].add(tokenId) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#339)
ERC721._mint(address,uint256) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#333-344) ignores return value by _tokenOwners.set(tokenId,to) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#341)
ERC721._burn(uint256) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#356-374) ignores return value by _holderTokens[owner].remove(tokenId) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#369)
ERC721._burn(uint256) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#356-374) ignores return value by _tokenOwners.remove(tokenId) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#371)
ERC721._transfer(address,address,uint256) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#387-402) ignores return value by _holderTokens[from].remove(tokenId) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#39
6)
ERC721._transfer(address,address,uint256) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#387-402) ignores return value by _holderTokens[to].add(tokenId) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#397)
ERC721._transfer(address,address,uint256) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#387-402) ignores return value by _tokenOwners.set(tokenId,to) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#399)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

ERC721MinterBurnerPauser.constructor(string,string,string).name (contracts/ERC721MinterBurnerPauser.sol#24) shadows:
        - ERC721.name() (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#121-123) (function)
        - IERC721Metadata.name() (node_modules/@openzeppelin/contracts/token/ERC721/IERC721Metadata.sol#16) (function)
ERC721MinterBurnerPauser.constructor(string,string,string).symbol (contracts/ERC721MinterBurnerPauser.sol#24) shadows:
        - ERC721.symbol() (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#128-130) (function)
        - IERC721Metadata.symbol() (node_modules/@openzeppelin/contracts/token/ERC721/IERC721Metadata.sol#21) (function)
ERC721MinterBurnerPauser.constructor(string,string,string).baseURI (contracts/ERC721MinterBurnerPauser.sol#24) shadows:
        - ERC721.baseURI() (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#158-160) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

Address.isContract(address) (node_modules/@openzeppelin/contracts/utils/Address.sol#26-35) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/utils/Address.sol#33)
Address._verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#171-188) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/utils/Address.sol#180-183)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Different versions of Solidity is used:
        - Version used: ['>=0.6.0<0.8.0', '>=0.6.2<0.8.0']
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/GSN/Context.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/access/AccessControl.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/introspection/ERC165.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/introspection/IERC165.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/math/SafeMath.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/ERC721Burnable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/ERC721Pausable.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/IERC721.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/IERC721Enumerable.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/IERC721Metadata.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/IERC721Receiver.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts/utils/Address.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/Counters.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/EnumerableMap.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/EnumerableSet.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/Pausable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/Strings.sol#3)
        - >=0.6.0<0.8.0 (contracts/ERC721MinterBurnerPauser.sol#1)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

AccessControl._setRoleAdmin(bytes32,bytes32) (node_modules/@openzeppelin/contracts/access/AccessControl.sol#201-204) is never used and should be removed
Address.functionCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#79-81) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#104-106) is never used and should be removed
Address.functionDelegateCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#153-155) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#163-169) is never used and should be removed
Address.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#129-131) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#139-145) is never used and should be removed
Address.sendValue(address,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#53-59) is never used and should be removed
Context._msgData() (node_modules/@openzeppelin/contracts/utils/Context.sol#20-23) is never used and should be removed
Counters.current(Counters.Counter) (node_modules/@openzeppelin/contracts/utils/Counters.sol#28-30) is never used and should be removed
Counters.decrement(Counters.Counter) (node_modules/@openzeppelin/contracts/utils/Counters.sol#37-39) is never used and should be removed
Counters.increment(Counters.Counter) (node_modules/@openzeppelin/contracts/utils/Counters.sol#32-35) is never used and should be removed
ERC721._safeMint(address,uint256) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#308-310) is never used and should be removed

ERC721._safeMint(address,uint256,bytes) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#316-319) is never used and should be removed
EnumerableMap._get(EnumerableMap.Map,bytes32) (node_modules/@openzeppelin/contracts/utils/EnumerableMap.sol#163-167) is never used and should be removed
EnumerableMap._tryGet(EnumerableMap.Map,bytes32) (node_modules/@openzeppelin/contracts/utils/EnumerableMap.sol#150-154) is never used and should be removed
EnumerableMap.get(EnumerableMap.UintToAddressMap,uint256) (node_modules/@openzeppelin/contracts/utils/EnumerableMap.sol#253-255) is never used and should be removed
EnumerableMap.tryGet(EnumerableMap.UintToAddressMap,uint256) (node_modules/@openzeppelin/contracts/utils/EnumerableMap.sol#241-244) is never used and should be removed
EnumerableSet.add(EnumerableSet.Bytes32Set,bytes32) (node_modules/@openzeppelin/contracts/utils/EnumerableSet.sol#147-149) is never used and should be removed
EnumerableSet.at(EnumerableSet.Bytes32Set,uint256) (node_modules/@openzeppelin/contracts/utils/EnumerableSet.sol#185-187) is never used and should be removed
EnumerableSet.contains(EnumerableSet.Bytes32Set,bytes32) (node_modules/@openzeppelin/contracts/utils/EnumerableSet.sol#164-166) is never used and should be removed
EnumerableSet.contains(EnumerableSet.UintSet,uint256) (node_modules/@openzeppelin/contracts/utils/EnumerableSet.sol#273-275) is never used and should be removed
EnumerableSet.length(EnumerableSet.Bytes32Set) (node_modules/@openzeppelin/contracts/utils/EnumerableSet.sol#171-173) is never used and should be removed
EnumerableSet.remove(EnumerableSet.Bytes32Set,bytes32) (node_modules/@openzeppelin/contracts/utils/EnumerableSet.sol#157-159) is never used and should be removed
SafeMath.add(uint256,uint256) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#85-89) is never used and should be removed
SafeMath.div(uint256,uint256) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#135-138) is never used and should be removed
SafeMath.div(uint256,uint256,string) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#190-193) is never used and should be removed
SafeMath.mod(uint256,uint256) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#152-155) is never used and should be removed
SafeMath.mod(uint256,uint256,string) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#210-213) is never used and should be removed
SafeMath.mul(uint256,uint256) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#116-121) is never used and should be removed
SafeMath.sub(uint256,uint256) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#101-104) is never used and should be removed
SafeMath.sub(uint256,uint256,string) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#170-173) is never used and should be removed
SafeMath.tryAdd(uint256,uint256) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#24-28) is never used and should be removed
SafeMath.tryDiv(uint256,uint256) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#60-63) is never used and should be removed
SafeMath.tryMod(uint256,uint256) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#70-73) is never used and should be removed
SafeMath.tryMul(uint256,uint256) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#45-53) is never used and should be removed
SafeMath.trySub(uint256,uint256) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#35-38) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/GSN/Context.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/access/AccessControl.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/introspection/ERC165.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/introspection/IERC165.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/math/SafeMath.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/ERC721Burnable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/ERC721Pausable.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/IERC721.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/IERC721Enumerable.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/IERC721Metadata.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/IERC721Receiver.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts/utils/Address.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/Counters.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/EnumerableMap.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/EnumerableSet.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/Pausable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/Strings.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (contracts/ERC721MinterBurnerPauser.sol#1) is too complex
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#53-59):
        - (success) = recipient.call{value: amount}() (node_modules/@openzeppelin/contracts/utils/Address.sol#57)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#114-121):
        - (success,returndata) = target.call{value: value}(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#119)
Low level call in Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#139-145):
        - (success,returndata) = target.staticcall(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#143)
Low level call in Address.functionDelegateCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#163-169):
        - (success,returndata) = target.delegatecall(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#167)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Parameter ERC721.safeTransferFrom(address,address,uint256,bytes)._data (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#245) is not in mixedCase
Parameter ERC721MinterBurnerPauser.mint(address,uint256,string)._data (contracts/ERC721MinterBurnerPauser.sol#44) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (node_modules/@openzeppelin/contracts/utils/Context.sol#21)" inContext (node_modules/@openzeppelin/contracts/utils/Context.sol#15-24)
```

```
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

getRoleMemberCount(bytes32) should be declared external:
        - AccessControl.getRoleMemberCount(bytes32) (node_modules/@openzeppelin/contracts/access/AccessControl.sol#95-97)
getRoleMember(bytes32,uint256) should be declared external:
        - AccessControl.getRoleMember(bytes32,uint256) (node_modules/@openzeppelin/contracts/access/AccessControl.sol#111-113)
getRoleAdmin(bytes32) should be declared external:
        - AccessControl.getRoleAdmin(bytes32) (node_modules/@openzeppelin/contracts/access/AccessControl.sol#121-123)
grantRole(bytes32,address) should be declared external:
        - AccessControl.grantRole(bytes32,address) (node_modules/@openzeppelin/contracts/access/AccessControl.sol#135-139)
revokeRole(bytes32,address) should be declared external:
        - AccessControl.revokeRole(bytes32,address) (node_modules/@openzeppelin/contracts/access/AccessControl.sol#150-154)
renounceRole(bytes32,address) should be declared external:
        - AccessControl.renounceRole(bytes32,address) (node_modules/@openzeppelin/contracts/access/AccessControl.sol#170-174)
supportsInterface(bytes4) should be declared external:
        - ERC165.supportsInterface(bytes4) (node_modules/@openzeppelin/contracts/introspection/ERC165.sol#35-37)
balanceOf(address) should be declared external:
        - ERC721.balanceOf(address) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#106-109)
name() should be declared external:
        - ERC721.name() (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#121-123)
symbol() should be declared external:
        - ERC721.symbol() (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#128-130)
tokenURI(uint256) should be declared external:
        - ERC721.tokenURI(uint256) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#135-151)
tokenOfOwnerByIndex(address,uint256) should be declared external:
        - ERC721.tokenOfOwnerByIndex(address,uint256) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#165-167)
totalSupply() should be declared external:
        - ERC721.totalSupply() (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#172-175)
tokenByIndex(uint256) should be declared external:
        - ERC721.tokenByIndex(uint256) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#180-183)
approve(address,uint256) should be declared external:
        - ERC721.approve(address,uint256) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#188-197)
setApprovalForAll(address,bool) should be declared external:
        - ERC721.setApprovalForAll(address,bool) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#211-216)
transferFrom(address,address,uint256) should be declared external:
        - ERC721.transferFrom(address,address,uint256) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#228-233)
safeTransferFrom(address,address,uint256) should be declared external:
        - ERC721.safeTransferFrom(address,address,uint256) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#238-240)
burn(uint256) should be declared external:
        - ERC721Burnable.burn(uint256) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721Burnable.sol#20-24)
mint(address,uint256,string) should be declared external:
        - ERC721MinterBurnerPauser.mint(address,uint256,string) (contracts/ERC721MinterBurnerPauser.sol#44-49)
pause() should be declared external:
        - ERC721MinterBurnerPauser.pause() (contracts/ERC721MinterBurnerPauser.sol#60-63)
unpause() should be declared external:
        - ERC721MinterBurnerPauser.unpause() (contracts/ERC721MinterBurnerPauser.sol#74-77)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

ERC721Safe.sol

```
ERC721._mint(address,uint256) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#333-344) ignores return value by _holderTokens[to].add(tokenId) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#339)
ERC721._mint(address,uint256) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#333-344) ignores return value by _tokenOwners.set(tokenId,to) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#341)
ERC721._burn(uint256) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#354-374) ignores return value by _holderTokens[owner].remove(tokenId) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#369)
ERC721._burn(uint256) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#356-374) ignores return value by _tokenOwners.remove(tokenId) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#371)
ERC721._transfer(address,address,uint256) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#387-402) ignores return value by _holderTokens[from].remove(tokenId) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#39
6)
ERC721._transfer(address,address,uint256) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#387-402) ignores return value by _holderTokens[to].add(tokenId) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#397)
ERC721._transfer(address,address,uint256) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#387-402) ignores return value by _tokenOwners.set(tokenId,to) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#399)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

ERC721MinterBurnerPauser.constructor(string,string,string).name (contracts/ERC721MinterBurnerPauser.sol#26) shadows:
        - ERC721.name() (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#121-123) (function)
        - IERC721Metadata.name() (node_modules/@openzeppelin/contracts/token/ERC721/IERC721Metadata.sol#16) (function)
ERC721MinterBurnerPauser.constructor(string,string,string).symbol (contracts/ERC721MinterBurnerPauser.sol#24) shadows:
        - ERC721.symbol() (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#128-130) (function)
        - IERC721Metadata.symbol() (node_modules/@openzeppelin/contracts/token/ERC721/IERC721Metadata.sol#21) (function)
ERC721MinterBurnerPauser.constructor(string,string,string).baseURI (contracts/ERC721MinterBurnerPauser.sol#24) shadows:
        - ERC721.baseURI() (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#158-160) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

Address.isContract(address) (node_modules/@openzeppelin/contracts/utils/Address.sol#26-35) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/utils/Address.sol#33)
Address._verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#171-188) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/utils/Address.sol#180-183)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Different versions of Solidity is used:
        - Version used: ['>=0.6.0<0.8.0', '>=0.6.2<0.8.0']
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/GSN/Context.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/access/AccessControl.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/introspection/ERC165.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/introspection/IERC165.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/math/SafeMath.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/ERC721Burnable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/ERC721Pausable.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/IERC721.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/IERC721Enumerable.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/IERC721Metadata.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/IERC721Receiver.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts/utils/Address.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/Counters.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/EnumerableMap.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/EnumerableSet.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/Pausable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/Strings.sol#3)
        - >=0.6.0<0.8.0 (contracts/ERC721MinterBurnerPauser.sol#1)
        - >=0.6.0<0.8.0 (contracts/ERC721Safe.sol#1)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

AccessControl._setRoleAdmin(bytes32,bytes32) (node_modules/@openzeppelin/contracts/access/AccessControl.sol#201-204) is never used and should be removed
Address.functionCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#79-81) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#104-106) is never used and should be removed
Address.functionDelegateCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#153-155) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#163-169) is never used and should be removed
Address.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#129-131) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#139-145) is never used and should be removed
Address.sendValue(address,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#53-59) is never used and should be removed
Context._msgData() (node_modules/@openzeppelin/contracts/utils/Context.sol#20-23) is never used and should be removed
Counters.current(Counters.Counter) (node_modules/@openzeppelin/contracts/utils/Counters.sol#28-30) is never used and should be removed
Counters.decrement(Counters.Counter) (node_modules/@openzeppelin/contracts/utils/Counters.sol#37-39) is never used and should be removed
Counters.increment(Counters.Counter) (node_modules/@openzeppelin/contracts/utils/Counters.sol#32-35) is never used and should be removed

ERC721._safeMint(address,uint256) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#308-310) is never used and should be removed
ERC721._safeMint(address,uint256,bytes) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#316-319) is never used and should be removed
ERC721Safe.burnERC721(address,uint256) (contracts/ERC721Safe.sol#60-71) is never used and should be removed
ERC721Safe.lockERC721(address,address,address,uint256) (contracts/ERC721Safe.sol#33-37) is never used and should be removed
ERC721Safe.mintERC721(address,address,uint256,bytes) (contracts/ERC721Safe.sol#55-61) is never used and should be removed
ERC721Safe.releaseERC721(address,address,address,uint256) (contracts/ERC721Safe.sol#46-49) is never used and should be removed
EnumerableMap._get(EnumerableMap.Map,bytes32) (node_modules/@openzeppelin/contracts/utils/EnumerableMap.sol#163-167) is never used and should be removed
EnumerableMap._tryGet(EnumerableMap.Map,bytes32) (node_modules/@openzeppelin/contracts/utils/EnumerableMap.sol#150-154) is never used and should be removed
EnumerableMap.get(EnumerableMap.UintToAddressMap,uint256) (node_modules/@openzeppelin/contracts/utils/EnumerableMap.sol#253-255) is never used and should be removed
EnumerableMap.tryGet(EnumerableMap.UintToAddressMap,uint256) (node_modules/@openzeppelin/contracts/utils/EnumerableMap.sol#241-244) is never used and should be removed
EnumerableSet.add(EnumerableSet.Bytes32Set,bytes32) (node_modules/@openzeppelin/contracts/utils/EnumerableSet.sol#147-149) is never used and should be removed
EnumerableSet.at(EnumerableSet.Bytes32Set,uint256) (node_modules/@openzeppelin/contracts/utils/EnumerableSet.sol#185-187) is never used and should be removed
EnumerableSet.contains(EnumerableSet.Bytes32Set,bytes32) (node_modules/@openzeppelin/contracts/utils/EnumerableSet.sol#164-166) is never used and should be removed
EnumerableSet.contains(EnumerableSet.UintSet,uint256) (node_modules/@openzeppelin/contracts/utils/EnumerableSet.sol#273-275) is never used and should be removed
EnumerableSet.length(EnumerableSet.Bytes32Set) (node_modules/@openzeppelin/contracts/utils/EnumerableSet.sol#171-173) is never used and should be removed
EnumerableSet.remove(EnumerableSet.Bytes32Set,bytes32) (node_modules/@openzeppelin/contracts/utils/EnumerableSet.sol#157-159) is never used and should be removed
SafeMath.add(uint256,uint256) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#85-89) is never used and should be removed
SafeMath.div(uint256,uint256) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#135-138) is never used and should be removed
SafeMath.div(uint256,uint256,string) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#190-193) is never used and should be removed
SafeMath.mod(uint256,uint256) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#152-155) is never used and should be removed
SafeMath.mod(uint256,uint256,string) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#210-213) is never used and should be removed
SafeMath.mul(uint256,uint256) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#116-121) is never used and should be removed
SafeMath.sub(uint256,uint256) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#101-104) is never used and should be removed
SafeMath.sub(uint256,uint256,string) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#170-173) is never used and should be removed
SafeMath.tryAdd(uint256,uint256) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#24-28) is never used and should be removed
SafeMath.tryDiv(uint256,uint256) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#60-63) is never used and should be removed
SafeMath.tryMod(uint256,uint256) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#70-73) is never used and should be removed
SafeMath.tryMul(uint256,uint256) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#45-53) is never used and should be removed
SafeMath.trySub(uint256,uint256) (node_modules/@openzeppelin/contracts/math/SafeMath.sol#35-38) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/GSN/Context.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/access/AccessControl.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/introspection/ERC165.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/introspection/IERC165.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/math/SafeMath.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/ERC721Burnable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/ERC721Pausable.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/IERC721.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/IERC721Enumerable.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/IERC721Metadata.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/IERC721Receiver.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts/utils/Address.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/Counters.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/EnumerableMap.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/EnumerableSet.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/Pausable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/Strings.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (contracts/ERC721MinterBurnerPauser.sol#1) is too complex
Pragma version>=0.6.0<0.8.0 (contracts/ERC721Safe.sol#1) is too complex
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
```

```
Low level call in Address.sendValue(address,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#53-59):
        - (success) = recipient.call{value: amount}() (node_modules/@openzeppelin/contracts/utils/Address.sol#57)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#114-121):
        - (success,returndata) = target.call{value: value}(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#119)
Low level call in Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#139-145):
        - (success,returndata) = target.staticcall(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#143)
Low level call in Address.functionDelegateCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#163-169):
        - (success,returndata) = target.delegatecall(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#167)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Parameter ERC721.safeTransferFrom(address,address,uint256,bytes)._data (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#245) is not in mixedCase
Parameter ERC721MinterBurnerPauser.mint(address,uint256,string)._data (contracts/ERC721MinterBurnerPauser.sol#44) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (node_modules/@openzeppelin/contracts/utils/Context.sol#21)" inContext (node_modules/@openzeppelin/contracts/utils/Context.sol#15-24)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

getRoleMemberCount(bytes32) should be declared external:
        - AccessControl.getRoleMemberCount(bytes32) (node_modules/@openzeppelin/contracts/access/AccessControl.sol#95-97)
getRoleMember(bytes32,uint256) should be declared external:
        - AccessControl.getRoleMember(bytes32,uint256) (node_modules/@openzeppelin/contracts/access/AccessControl.sol#111-113)
getRoleAdmin(bytes32) should be declared external:
        - AccessControl.getRoleAdmin(bytes32) (node_modules/@openzeppelin/contracts/access/AccessControl.sol#121-123)
grantRole(bytes32,address) should be declared external:
        - AccessControl.grantRole(bytes32,address) (node_modules/@openzeppelin/contracts/access/AccessControl.sol#135-139)
revokeRole(bytes32,address) should be declared external:
        - AccessControl.revokeRole(bytes32,address) (node_modules/@openzeppelin/contracts/access/AccessControl.sol#150-154)
renounceRole(bytes32,address) should be declared external:
        - AccessControl.renounceRole(bytes32,address) (node_modules/@openzeppelin/contracts/access/AccessControl.sol#170-174)
supportsInterface(bytes4) should be declared external:
        - ERC165.supportsInterface(bytes4) (node_modules/@openzeppelin/contracts/introspection/ERC165.sol#35-37)
balanceOf(address) should be declared external:
        - ERC721.balanceOf(address) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#106-109)
name() should be declared external:
        - ERC721.name() (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#121-123)
symbol() should be declared external:
        - ERC721.symbol() (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#128-130)
tokenURI(uint256) should be declared external:
        - ERC721.tokenURI(uint256) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#135-151)
tokenOfOwnerByIndex(address,uint256) should be declared external:
        - ERC721.tokenOfOwnerByIndex(address,uint256) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#165-167)
totalSupply() should be declared external:
        - ERC721.totalSupply() (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#172-175)
tokenByIndex(uint256) should be declared external:
        - ERC721.tokenByIndex(uint256) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#180-183)
approve(address,uint256) should be declared external:
        - ERC721.approve(address,uint256) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#188-197)
setApprovalForAll(address,bool) should be declared external:
        - ERC721.setApprovalForAll(address,bool) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#211-216)
transferFrom(address,address,uint256) should be declared external:
        - ERC721.transferFrom(address,address,uint256) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#228-233)
safeTransferFrom(address,address,uint256) should be declared external:
        - ERC721.safeTransferFrom(address,address,uint256) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#238-240)
burn(uint256) should be declared external:
        - ERC721Burnable.burn(uint256) (node_modules/@openzeppelin/contracts/token/ERC721/ERC721Burnable.sol#20-24)
mint(address,uint256,string) should be declared external:
        - ERC721MinterBurnerPauser.mint(address,uint256,string) (contracts/ERC721MinterBurnerPauser.sol#44-49)
pause() should be declared external:
        - ERC721MinterBurnerPauser.pause() (contracts/ERC721MinterBurnerPauser.sol#60-63)
unpause() should be declared external:
        - ERC721MinterBurnerPauser.unpause() (contracts/ERC721MinterBurnerPauser.sol#74-77)
fundERC721(address,address,uint256) should be declared external:


        - ERC721Safe.fundERC721(address,address,uint256) (contracts/ERC721Safe.sol#21-24)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

- The reentrancies and the uninitialized state variables flagged by Slither are false positives. 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:

ERC20Handler.sol

Report for contracts/handlers/ERC20Handler.sol
https://dashboard.mythx.io/#/console/analyses/581af3ef-000b-431f-811f-b1ec610e7426

Line	SWC Title	Severity	Short Description
1	(SWC-103) Floating Pragma	Low	A floating pragma is set.
13	(SWC-123) Requirement Violation	Low	Requirement violation.
113	(SWC-107) Reentrancy	Low	Read of persistent state following external call.

ERC721Handler.sol

Report for contracts/handlers/ERC721Handler.sol
https://dashboard.mythx.io/#/console/analyses/387b8a42-130e-4720-ba92-067a015fb2fb

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

GenericHandler.sol

Report for handlers/GenericHandler.sol
https://dashboard.mythx.io/#/console/analyses/a3146d6d-064e-471d-8215-8c6f61fd5d56

Line	SWC Title	Severity	Short Description
1	(SWC-103) Floating Pragma	Low	A floating pragma is set.
173	(SWC-107) Reentrancy	Low	Read of persistent state following external call.

HandlerHelpers.sol

Report for handlers/HandlerHelpers.sol
https://dashboard.mythx.io/#/console/analyses/36e575bb-cc7e-4631-b9bf-b9116ebd7b5b

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

Whitelist.sol

Report for contracts/utils/Whitelist.sol
https://dashboard.mythx.io/#/console/analyses/414e8e7d-2bbd-44cb-a316-107534575c43

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

Bridge.sol

Report for contracts/Bridge.sol
https://dashboard.mythx.io/#/console/analyses/a79953f9-8609-4af0-8b8d-e4cd6e7e2645

Line	SWC Title	Severity	Short Description
1	(SWC-103) Floating Pragma	Low	A floating pragma is set.
17	(SWC-123) Requirement Violation	Low	Requirement violation.
209	(SWC-123) Requirement Violation	Low	Requirement violation.
337	(SWC-120) Weak Sources of Randomness from Chain Attributes	Low	Potential use of "block.number" as source of randomness.
343	(SWC-120) Weak Sources of Randomness from Chain Attributes	Low	Potential use of "block.number" as source of randomness.
385	(SWC-120) Weak Sources of Randomness from Chain Attributes	Low	Potential use of "block.number" as source of randomness.
429	(SWC-113) DoS with Failed Call	Medium	Multiple calls are executed in the same transaction.

ERC20Safe.sol

Report for contracts/ERC20Safe.sol
https://dashboard.mythx.io/#/console/analyses/f303d09a-6d25-460f-bd32-047e6433d22f

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

ERC721MinterBurnerPauser.sol

Report for contracts/ERC721MinterBurnerPauser.sol
https://dashboard.mythx.io/#/console/analyses/5df258b5-90dc-4d86-9183-2b97ec12d705

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

ERC721Safe.sol

Report for contracts/ERC721Safe.sol
https://dashboard.mythx.io/#/console/analyses/f9c240b8-ea5d-42f2-9eb2-63497be550f2

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

- The floating pragma is correctly flagged, although it is set to the 0.7.6 version in the `truffle-config.js` file.
- The reentrancies flagged by MythX are false positives.
- `block.number` is not used as a source of randomness in any of the smart contracts.
- No major issues were found by MythX.

THANK YOU FOR CHOOSING

// HALBORN