Gains Trade

Smart Contract Security Audit

Prepared by: **Halborn**

Date of Engagement: **October 31st, 2022 - November 21st, 2022**

Visit: **Halborn.com**

DOCUMENT REVISION HISTORY

VERSION	MODIFICATION	DATE	AUTHOR
0.1	Document Creation	11/21/2022	Mariusz Maik
0.2	Draft Review	11/21/2022	Kubilay Onur Gungor
0.3	Draft Review	11/21/2022	Gabi Urrutia
0.4	Document Updates	12/09/2022	Roberto Reigada
0.5	Draft Review	12/09/2022	Roberto Reigada
0.6	Draft Review	12/09/2022	Piotr Cielas
0.7	Draft Review	12/09/2022	Gabi Urrutia
1.0	Remediation Plan	12/12/2022	Roberto Reigada
1.1	Remediation Plan Review	12/12/2022	Piotr Cielas
1.2	Remediation Plan Review	12/12/2022	Gabi Urrutia
1.3	Document Updates	12/15/2022	Roberto Reigada

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
Kubilay Onur Gungor	Halborn	Kubilay.Gungor@halborn.com
Mariusz Maik	Halborn	Mariusz.Maik@halborn.com
Roberto Reigada	Halborn	Roberto.Reigada@halborn.com
Piotr Cielas	Halborn	Piotr.Cielas@halborn.com

EXECUTIVE OVERVIEW

1.1 INTRODUCTION

Gains Trade is a decentralized trading platform for cryptocurrencies, stocks, and forex, fully on-chain.

Gains Trade engaged Halborn to conduct a security audit on their smart contracts beginning on October 31st, 2022 and ending on November 21st, 2022 . The security assessment was scoped to the smart contracts provided to the Halborn team.

1.2 AUDIT SUMMARY

The team at Halborn was provided three weeks for the engagement and assigned a full-time security engineer to audit the security of the smart contracts. 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 addressed and accepted by the Gains Trade team.

1.3 TEST APPROACH & METHODOLOGY

Halborn performed a combination of manual and automated security testing to balance efficiency, timeliness, practicality, and accuracy in regard to the scope of this audit. While manual testing is recommended to uncover flaws in logic, process, and implementation; automated testing techniques help enhance coverage of the bridge code and can quickly identify items

that do not follow security best practices. The following phases and associated tools were used throughout the term of the audit:

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

RISK METHODOLOGY:

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

RISK SCALE - LIKELIHOOD

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

RISK SCALE - IMPACT

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

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

CRITICAL	HIGH	MEDIUM	LOW	INFORMATIONAL

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

1.4 SCOPE

IN-SCOPE:

The security assessment was scoped to the following contracts:

- GToken.sol
- GTokenOpenPnlFeed.sol
- TWAPPriceGetter.sol
- GNSTokenBridge.sol
- NFTMintingBridge.sol

Commit ID: d6c7fe7f386b5678ee2597a17e866cee189f056e

Bridge commit ID: d70b3e5a47cf0f62bedd1363b7a10c2d9acc310e

OUT-OF-SCOPE:

Other smart contracts in the repository, external libraries and economical attacks.

Fixed Commit ID: a9a18643d0278886c79b56e0171b38a8a2945d08

Bridge fixed Commit ID: 2f709cdbd2cafbdac5bbb21d736502681c36ab4

2. ASSESSMENT SUMMARY & FINDINGS OVERVIEW

CRITICAL	HIGH	MEDIUM	LOW	INFORMATIONAL
1	0	1	1	5

LIKELIHOOD

IMPACT

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

EXECUTIVE OVERVIEW

SECURITY ANALYSIS	RISK LEVEL	REMEDIATION DATE
HAL-01 - BRIDGENFT CALL CAN BE FRONTRUN	Critical	SOLVED - 12/12/2022
HAL-02 - WRONG FEE SENT TO CHAINLINK NODES	Medium	RISK ACCEPTED
HAL-03 - THE TRANSFEROWNERSHIP PATTERN IS NOT FOLLOWED	Low	RISK ACCEPTED
HAL-04 - MISSING ZERO ADDRESS CHECKS	Low	RISK ACCEPTED
HAL-05 - GAS OVER-CONSUMPTION IN LOOPS	Informational	ACKNOWLEDGED
HAL-06 - REDUNDANT INITIALIZATION OF UINT VARIABLES TO 0	Informational	SOLVED - 12/12/2022
HAL-07 - USE OF REVERT STRINGS INSTEAD OF CUSTOM ERRORS	Informational	ACKNOWLEDGED
HAL-08 - MISSING/INCOMPLETE NATSPEC COMMENTS	Informational	ACKNOWLEDGED

FINDINGS & TECH DETAILS

3.1 (HAL-01) BRIDGENFT CALL CAN BE FRONTRUN - CRITICAL

Description:

In the NFTMintingBridge contract, the function bridgeNft() is used to bridge ERC721 tokens from one chain to another:

Listing 1: NFTMintingBridge.sol (Line 89)

```
74 /**
75  * bridgeNft
76  * @param dstChainId     The **LayerZero** destination chain ID.
77  * @param tokenId       The tokenId of the NFT
78  *
79  * Burns the NFT with ID <tokenID>. A cross-chain message is then
↳ sent to our counterpart on the destination chain to mint/release
↳ this NFT.
80  */
81 function bridgeNft(uint16 dstChainId, uint256 tokenId)
82     external
83     payable
84     whenNotPaused
85 {
86     require(msg.value != 0, "!fee"); // Make sure a native fee is
↳ supplied for the cross-chain message.
87
88     // Burn the NFT - This will only work if the bridge was
↳ approved by the token owner
89     nft.burn(tokenId);
90
91     bytes memory payloadBytes = _buildReleaseMesssage(msg.sender,
↳ tokenId);
92
93     // send LayerZero message
94     _lzSend(
95         dstChainId, // destination chainId
96         payloadBytes, // abi.encode()'ed bytes
97         payable(msg.sender), // refund address (LayerZero will
↳ refund any extra gas back to caller
98         address(0x0), // unused
99         bytes(""), // unused
```

```
100         msg.value // native fee amount
101     );
102
103     emit NftBurnReceived(dstChainId, msg.sender, tokenId);
104 }
```

The function burns the tokenId passed as parameter, but it does not check that the caller of the function actually owns that NFT. This opens up the following attack vector:

1. Alice owns NFT #1337.
2. Alice approves the NFTMintingBridge so then she can call bridgeNft() to bridge the NFT to a different chain.
3. Alice calls NFTMintingBridge.bridgeNft(2, 1337).
4. Bob is a malicious user who is monitoring the mempool for any NFTMintingBridge.bridgeNft() calls.
5. Bob detects Alice's call and front-runs her transaction by paying a higher gas fee. Bob's transaction is identical to Alice's, NFTMintingBridge.bridgeNft(2, 1337).
6. Bob's transaction is mined before Alice's. Alice's transaction reverts as the token was already burnt by Bob.
7. Bob gets the NFT minted in the new chain.

Risk Level:

Likelihood - 5
Impact - 5

Recommendation:

It is recommended to add the following require check in the bridgeNft() function:

Listing 2: NFTMintingBridge.sol (Line 87)

```
74 /**
75  * bridgeNft
```

```
76  * @param dstChainId     The **LayerZero** destination chain ID.
77  * @param tokenId        The tokenId of the NFT
78  *
79  * Burns the NFT with ID <tokenID>. A cross-chain message is then
↳ sent to our counterpart on the destination chain to mint/release
↳ this NFT.
80  */
81 function bridgeNft(uint16 dstChainId, uint256 tokenId)
82     external
83     payable
84     whenNotPaused
85 {
86     require(msg.value != 0, "!fee"); // Make sure a native fee is
↳ supplied for the cross-chain message.
87     require (nft.ownerOf(tokenId) == msg.sender, "!owner");
88     // Burn the NFT - This will only work if the bridge was
↳ approved by the token owner
89     nft.burn(tokenId);
90
91     bytes memory payloadBytes = _buildReleaseMesssage(msg.sender,
↳ tokenId);
92
93     // send LayerZero message
94     _lzSend(
95         dstChainId, // destination chainId
96         payloadBytes, // abi.encode()'ed bytes
97         payable(msg.sender), // refund address (LayerZero will
↳ refund any extra gas back to caller
98         address(0x0), // unused
99         bytes(""), // unused
100        msg.value // native fee amount
101    );
102
103    emit NftBurnReceived(dstChainId, msg.sender, tokenId);
104 }
```

Remediation Plan:

SOLVED: The Gains Trade team solved the issue in the Commit ID: a349e7438543d57b5845a6771d4e642a1a4868f4

Listing 3: NFTMintingBridge.sol (Line 89)

```
76 /**
77  * bridgeNft
78  * @param dstChainId     The **LayerZero** destination chain ID.
79  * @param tokenId       The tokenId of the NFT
80  *
81  * Burns the NFT with ID <tokenID>. A cross-chain message is then
↳ sent to our counterpart on the destination chain to mint/release
↳ this NFT.
82  */
83 function bridgeNft(uint16 dstChainId, uint256 tokenId)
84     external
85     payable
86     whenNotPaused
87 {
88     require(msg.value != 0, "!fee"); // Make sure a native fee is
↳ supplied for the cross-chain message.
89     require(nft.ownerOf(tokenId) == msg.sender, "!owner");
90
91     // Burn the NFT - This will only work if the bridge was
↳ approved by the token owner
92     nft.burn(tokenId);
93
94     bytes memory payloadBytes = _buildReleaseMesssage(msg.sender,
↳ tokenId);
95
96     // send LayerZero message
97     _lzSend(
98         dstChainId, // destination chainId
99         payloadBytes, // abi.encode()'ed bytes
100         payable(msg.sender), // refund address (LayerZero will
↳ refund any extra gas back to caller
101         address(0x0), // unused
102         bytes(""), // unused
103         msg.value // native fee amount
104     );
105
106     emit NftBurnReceived(dstChainId, msg.sender, tokenId);
107 }
```

3.2 (HAL-02) WRONG FEE SENT TO CHAINLINK NODES - MEDIUM

Description:

In the GTokenOpenPnlFeed contract, the makeOpenPnlRequest() function is used to send requests to a pool of Chainlink oracles:

Listing 4: GTokenOpenPnlFeed.sol (Lines 251-254)

```
243 // Create requests
244 function makeOpenPnlRequest() private{
245     Chainlink.Request memory linkRequest = buildChainlinkRequest(
246         job,
247         address(this),
248         this.fulfill.selector
249     );
250
251     uint linkFeePerNode = IERC20(chainlinkTokenAddress())
252         .balanceOf(address(this))
253         / LINK_FEE_BALANCE_DIVIDER
254         / oracles.length;
255
256     requests[++lastRequestId] = Request({
257         initiated: true,
258         active: true,
259         linkFeePerNode: linkFeePerNode
260     });
261
262     nextEpochValuesRequestCount++;
263     nextEpochValuesLastRequest = block.timestamp;
264
265     for(uint i; i < oracles.length; i ++){
266         requestIds[sendChainlinkRequestTo(
267             oracles[i],
268             linkRequest,
269             linkFeePerNode
270         )] = lastRequestId;
271     }
272
273     emit NextEpochValueRequested(
274         gToken.currentEpoch(),
```

```
275             lastRequestId,
276             job,
277             oracles.length,
278             linkFeePerNode
279         );
280 }
```

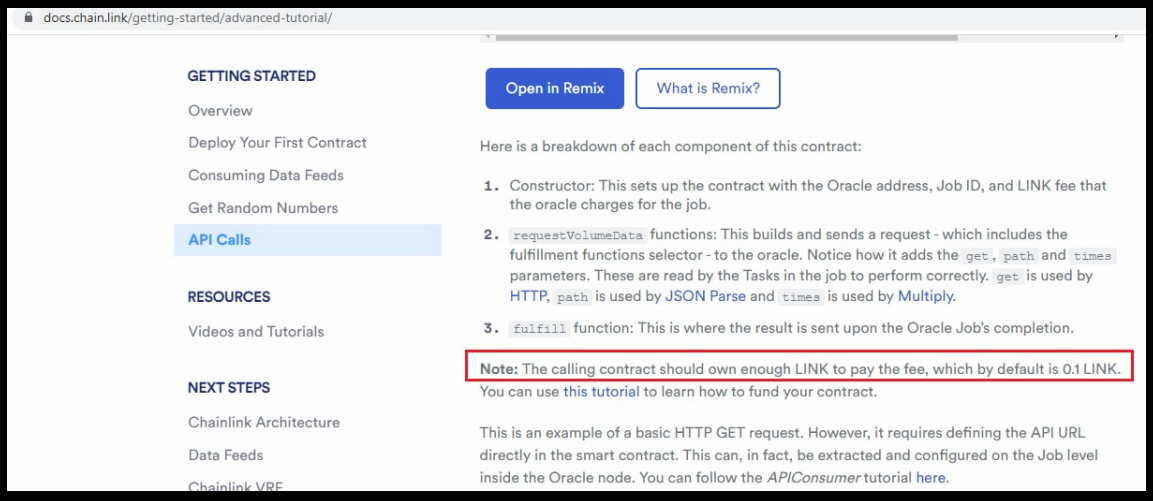
The linkFeePerNode variable is wrongly calculated, as it depends on the contract's balance. If the Link balance of the GTokenOpenPnlFeed contract is too high, more Link than needed would be sent to the Chainlink oracles as fees. In case that the Link balance of the GTokenOpenPnlFeed contract is not high enough, the contract will not send enough Link to the Chainlink oracles and the transactions will revert.

Risk Level:

Likelihood - 5
Impact - 1

Recommendation:

It is recommended to set a fixed Link fee based on the blockchain and the type of Chainlink oracle used. This Link fee is usually 0.1 Link:



Remediation Plan:

RISK ACCEPTED: The Gains Trade team accepted the risk and stated:
“We will send enough to the contract so that divided by 1000 it represents the amount we want, and then we will refill it every few months. This pattern allows everyone to send a link to the contract in a decentralized manner. It also means the transaction can never revert, unlike what the issue says because it can never run out of LINK.”

3.3 (HAL-03) THE TRANSFEROWNERSHIP PATTERN IS NOT FOLLOWED - INFORMATIONAL

Description:

The current ownership transfer process for the GToken contract inheriting from Ownable involves the current owner/timelock contract, calling the transferOwnership function.

If the nominated account is not a valid account, it is entirely possible that the owner may accidentally transfer ownership to an uncontrolled account, losing the access to all functions with the onlyOwner modifier.

Risk Level:

Likelihood - 1
Impact - 2

Recommendation:

It is recommended to implement a two-step process where the owner/timelock contract nominates an account and the nominated account needs to call an acceptOwnership() function for the transfer of the ownership to fully succeed. This ensures the nominated account is valid and active.

Remediation Plan:

RISK ACCEPTED: The Gains Trade team accepted the risk of this finding.

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

Description:

Contracts in-scope are missing address validation in some functions. It is possible to configure the zero address, which may cause issues during the contract execution.

Code Location:

The following function is not validating, that the given addresses in the newValues array are different from zero:

Listing 5: GTokenOpenPnlFeed.sol

```
173     function updateOracles(address[] memory newValues) external
 ↳ onlyOwner{
174         require(newValues.length >= minAnswers, "ARRAY_TOO_SMALL")
 ↳ ;
175
176         oracles = newValues;
177         emit OraclesUpdated(newValues);
178     }
```

The constructor is not validating that the address of the _token argument is different from zero:

Listing 6: TWAPPriceGetter.sol

```
16     constructor(IUniswapV3Pool _pool, address _token, uint32
 ↳ _twapInterval, uint _precision){
17         require(address(_pool) != address(0) && _twapInterval > 0
 ↳ && _precision > 0, "WRONG_TWAP_CONSTRUCTOR");
18
19         pool = _pool;
20         twapInterval = _twapInterval;
21         precision = _precision;
22
```

```
23          isGnsToken0InLp = pool.token0() == _token;
24      }
```

Risk Level:

Likelihood - 2
Impact - 2

Recommendation:

Consider adding a check to see if the address is different from the zero address.

Remediation Plan:

RISK ACCEPTED: The Gains Trade team accepted the risk of this finding.

3.5 (HAL-05) GAS OVER-CONSUMPTION IN LOOPS - INFORMATIONAL

Description:

In all the loops, the counter variable is incremented using i++. In loops, using ++i costs less gas per iteration than i++.

Code Location:

GTokenOpenPnlFeed.sol
- Line 197: for(uint i = 0; i < reqToResetCount; i++)
- Line 248: for(uint i = 0; i < oracles.length; i ++)
- Line 368: for(uint i = 0; i < array.length; i++)

Proof of Concept:

For example, based in the following test contract:

Listing 7: 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 }
```

The difference in the gas costs:

Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

It is recommended to use pre-incrementation instead of post-incrementation to update the value of an uint variable inside a loop to save some gas. This is not applicable outside of loops.

Remediation Plan:

ACKNOWLEDGED: The Gains Trade team acknowledged this finding.

3.6 (HAL-06) REDUNDANT INITIALIZATION OF UINT VARIABLES TO 0 - INFORMATIONAL

Description:

As the variable is a `uint`, it is already initialized to `0`. For example, `uint256 i = 0` reassigns the `0` to `i`, which wastes gas.

Code Location:

GTokenOpenPnlFeed.sol
- Line 197: `for(uint i = 0; i < reqToResetCount; i++)`
- Line 248: `for(uint i = 0; i < oracles.length; i ++`
- Line 368: `for(uint i = 0; i < array.length; i++)`

Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

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

Remediation Plan:

SOLVED: The Gains Trade team solved the issue by reducing the gas costs in the Commit ID: f1bcbb4da9efdd8a8be2d28a297cfc9b999d2e1a

3.7 (HAL-07) USE OF REVERT STRINGS INSTEAD OF CUSTOM ERRORS - INFORMATIONAL

Description:

Failed operations in the smart contracts in scope are reverted with accompanying messages selected from a set of hard-coded strings.

In the EVM, emitting a hard-coded string in an error message costs ~50 more gas than emitting a custom error. Additionally, hard-coded strings increase the gas required to deploy the contract.

Code Location:

GToken.sol
- Line 182:
`require(_contractAddresses.asset != address(0)...`
- Line 231:
`require(_msgSender()== manager, "ONLY_MANAGER");`
- Line 236:
`require(shareToAssetsPrice > 0, "PRICE_0");`
- Line 237:
`require(assetsOrShares > 0, "VALUE_0");`
- Line 242:
`require(maxDiscountP > 0, "NO_ACTIVE_DISCOUNT");`
- Line 243:
`require(lockDuration >= MIN_LOCK_DURATION, "BELOW_MIN_LOCK_DURATION");`
- Line 244:
`require(lockDuration <= MAX_LOCK_DURATION, "ABOVE_MAX_LOCK_DURATION");`
- Line 250:
`require(newOwner != address(0), "Ownable: new owner is the zero address");`
- Line 251:
`require(newOwner != manager && newOwner != admin, "WRONG_VALUE");`

- Line 256:
```
require(newValue != address(0), "ADDRESS_0");
```
- Line 257:
```
require(newValue != owner()&& newValue != admin, "WRONG_VALUE");
```
- Line 263:
```
require(newValue != address(0), "ADDRESS_0");
```
- Line 264:
```
require(newValue != owner()&& newValue != manager, "WRONG_VALUE");
```
- Line 270:
```
require(newValue != address(0), "ADDRESS_0");
```
- Line 276:
```
require(newValue.addr != address(0), "ADDRESS_0");
```
- Line 277:
```
require(newValue.signature.length > 0, "BYTES_0");
```
- Line 283:
```
require(newValue != address(0), "ADDRESS_0");
```
- Line 295:
```
require(newValue >= MIN_DAILY_ACC_PNL_DELTA, "BELOW_MIN");
```
- Line 301:
```
require(newValue[1] > newValue[0], "WRONG_VALUES");
```
- Line 307:
```
require(newValue <= MAX_SUPPLY_INCREASE_DAILY_P, "ABOVE_MAX");
```
- Line 313:
```
require(newValue <= MAX_LOSSES_BURN_P, "ABOVE_MAX");
```
- Line 319:
```
require(newValue <= MAX_GNS_SUPPLY_MINT_DAILY_P, "ABOVE_MAX");
```
- Line 325:
```
require(newValue <= MAX_DISCOUNT_P, "ABOVE_MAX_DISCOUNT");
```
- Line 331:
```
require(newValue >= 100 * PRECISION, "BELOW_MIN");
```
- Line 348:
```
require(success == true, "GNS_PRICE_CALL_FAILED");
```
- Line 351:
```
require(price > 0, "GNS_TOKEN_PRICE_0");
```
- Line 433:
```
require(totalSharesBeingWithdrawn(sender)<= balanceOf(sender)- amount,
"PENDING_WITHDRAWAL");
```

- Line 443:
`require(totalSharesBeingWithdrawn(from)<= balanceOf(from)- amount, "PENDING_WITHDRAWAL");`
- Line 499:
`require(assets <= maxDeposit(receiver), "ERC4626: deposit more than max");`
- Line 512:
`require(shares <= maxMint(receiver), "ERC4626: mint more than max");`
- Line 526:
`require(assets <= maxWithdraw(owner), "ERC4626: withdraw more than max");`
- Line 542:
`require(shares <= maxRedeem(owner), "ERC4626: redeem more than max");`
- Line 578:
`require(openTradesPnlFeed.nextEpochValuesRequestCount()== 0, "END_OF_EPOCH");`
- Line 582:
`require(sender == owner || allowance > 0 && allowance >= shares, "NOT_ALLOWED");`
- Line 584:
`require(totalSharesBeingWithdrawn(owner)+ shares <= balanceOf(owner), "MORE_THAN_BALANCE");`
- Line 593:
`require(shares <= withdrawRequests[owner][unlockEpoch], "MORE_THAN_WITHDRAW_AMOUNT");`
- Line 597:
`require(sender == owner || allowance > 0 && allowance >= shares, "NOT_ALLOWED");`
- Line 614:
`require(simulatedAssets <= maxDeposit(receiver), "DEPOSIT_MORE_THAN_MAX");`
- Line 630:
`require(shares <= maxMint(receiver), "MINT_MORE_THAN_MAX");`
- Line 651:
`require(assets > assetsDeposited, "NO_DISCOUNT");`
- Line 683:
`require(owner == sender...`

- Line 686:
`require(block.timestamp >= d.atTimestamp + d.lockDuration, "NOT_UNLOCKED");`
- Line 695:
`require(accPnlPerToken <= int(PRECISION), "NOT_ENOUGH_ASSETS");`
- Line 725:
`require(sender == pnlHandler, "ONLY_TRADING_PNL_HANDLER");`
- Line 734:
`require(accPnlPerToken <= int(PRECISION), "NOT_ENOUGH_ASSETS");`
- Line 738:
`require(dailyAccPnlDelta <= int(maxDailyAccPnlDelta), "MAX_DAILY_PNL");`
- Line 780:
`require(assets <= assetsToDeplete, "AMOUNT_TOO_BIG");`
- Line 801:
`require(accPnlPerTokenUsed > 0, "NOT_UNDER_COLLATERALIZED");`
- Line 804:
`require(assets <= uint(accPnlPerTokenUsed)* supply / PRECISION, "AMOUNT_TOO_BIG");`
- Line 814:
`require(dailyMintedGns <= maxGnsSupplyMintDailyP...`
- Line 839:
`require(sender == address(openTradesPnlFeed), "ONLY_PNL_FEED");`

GTokenOpenPnlFeed.sol
- Line 104:
`require(_linkToken != address(0)`
- Line 122:
`require(msg.sender == IOwnable(address(gToken)).owner(), "ONLY_OWNER");`
- Line 127:
`require(msg.sender == gToken.manager(), "ONLY_MANAGER");`
- Line 132:
`require(msg.sender == gToken.admin(), "ONLY_ADMIN");`
- Line 138:
`require(newValue >= MIN_REQUESTS_START, "BELOW_MIN");`
- Line 139:
`require(newValue <= MAX_REQUESTS_START, "ABOVE_MAX");`
- Line 145:
`require(newValue >= MIN_REQUESTS_EVERY, "BELOW_MIN");`

- Line 146:
`require(newValue <= MAX_REQUESTS_EVERY, "ABOVE_MAX");`
- Line 152:
`require(newValue >= MIN_REQUESTS_COUNT, "BELOW_MIN");`
- Line 153:
`require(newValue <= MAX_REQUESTS_COUNT, "ABOVE_MAX");`
- Line 169:
`require(newValue >= MIN_ANSWERS, "BELOW_MIN");`
- Line 170:
`require(newValue % 2 == 1, "EVEN");`
- Line 171:
`require(newValue <= oracles.length / 2, "ABOVE_MAX");`
- Line 177:
`require(_index < oracles.length, "INDEX_TOO_BIG");`
- Line 178:
`require(newValue != address(0), "VALUE_0");`
- Line 184:
`require(newValues.length >= minAnswers * 2, "ARRAY_TOO_SMALL");`
- Line 190:
`require(newValue != bytes32(0), "VALUE_0");`
- Line 198:
`require(reqToResetCount > 0, "NO_REQUEST_TO_RESET");`
- Line 218:
`require(block.timestamp - gToken.currentEpochStart()`

TWAPPriceGetter.sol
- Line 34:
`require(address(_uniV3Pool)!= address(0)...`
- Line 49:
`require(address(_uniV3Pool)!= address(0), "WRONG_VALUE");`
- Line 56:
`require(_twapInterval >= MIN_TWAP_PERIOD && _twapInterval <= MAX_TWAP_PERIOD, "WRONG_VALUE");`

GNSTokenBridge.sol
- Line 104:
`require(msg.value != 0, "!fee");`

NFTMintingBridge.sol
- Line 86:
`require(msg.value != 0, "!fee");`

Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

Custom errors are available since Solidity 0.8.4 version. Consider replacing all the revert strings with custom errors.

Remediation Plan:

ACKNOWLEDGED: The Gains Trade team acknowledged this finding.

3.8 (HAL-08) MISSING/INCOMPLETE NATSPEC COMMENTS - INFORMATIONAL

Description:

It was identified that the contracts are missing or have incomplete code documentation, which affects the understandability, auditability, and usability of the code.

Solidity contracts can use a special form of comments to provide rich documentation for functions, return variables, and more. This special form is named the Ethereum Natural Language Specification Format (*NatSpec**).

Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

Consider adding full **NatSpec** comments so that all functions have full code documentation for future use.

Remediation Plan:

ACKNOWLEDGED: The Gains Trade team acknowledged this finding.

AUTOMATED TESTING

4.1 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:

GToken.sol

```
OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#94) shadows:
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#36)
ERC20Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#400) shadows:
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#36)
ERC4626Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#269) shadows:
        - ERC20Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#400)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#36)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variable-shadowing

MathUpgradeable.mulDiv(uint256,uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#102)
        -inverse = (3 * denominator) ^ 2 (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#117)
MathUpgradeable.mulDiv(uint256,uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#102)
        -inverse *= 2 - denominator * inverse (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#121)
MathUpgradeable.mulDiv(uint256,uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#102)
        -inverse *= 2 - denominator * inverse (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#122)
MathUpgradeable.mulDiv(uint256,uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#102)
        -inverse *= 2 - denominator * inverse (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#123)
MathUpgradeable.mulDiv(uint256,uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#102)
        -inverse *= 2 - denominator * inverse (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#124)
MathUpgradeable.mulDiv(uint256,uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#102)
        -inverse *= 2 - denominator * inverse (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#125)
MathUpgradeable.mulDiv(uint256,uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#102)
        -inverse *= 2 - denominator * inverse (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#126)
MathUpgradeable.mulDiv(uint256,uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) performs a multiplication on the result of a division:
        -prod0 = prod0 / twos (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#105)
        -result = prod0 * inverse (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#132)
GToken.lockDiscountP(uint256,uint256) (contracts/v6.3/GToken.sol#365-373) performs a multiplication on the result of a division:
        -(maxDiscountP * (maxDiscountThreshold - collatR) / (maxDiscountThreshold - 100 * PRECISION)) * lockDuration / MAX_LOCK_DURATION (contracts/v6.3/GToken.sol#366-372)
GToken.scaleVariables(uint256,uint256,bool) (contracts/v6.3/GToken.sol#553-574) performs a multiplication on the result of a division:
        -accPnlPerToken = accPnlPerToken * int256(supply) / (int256(supply + shares)) (contracts/v6.3/GToken.sol#557-562)
        -accPnlPerToken = accPnlPerToken * int256(supply) / (int256(supply - shares)) (contracts/v6.3/GToken.sol#557-562)
GToken.scaleVariables(uint256,uint256,bool) (contracts/v6.3/GToken.sol#553-574) performs a multiplication on the result of a division:
        -totalLiability += int256(shares) * totalLiability / int256(supply) * (int256(-1)) (contracts/v6.3/GToken.sol#565-568)
GToken.scaleVariables(uint256,uint256,bool) (contracts/v6.3/GToken.sol#553-574) performs a multiplication on the result of a division:
        -totalLiability += int256(shares) * totalLiability / int256(supply) * (int256(-1)) (contracts/v6.3/GToken.sol#565-568)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#divide-before-multiply

Reentrancy in GToken.refill(uint256) (contracts/v6.3/GToken.sol#800-831):
        External calls:
        - amountGns = assets * GNS_PRECISION / gnsTokenToAssetsPrice() (contracts/v6.3/GToken.sol#811)
                - (success,result) = gnsPriceProvider.addr.staticcall(gnsPriceProvider.signature) (contracts/v6.3/GToken.sol#344-346)
        - SafeERC20Upgradeable.safeTransferFrom(_assetIERC20(),sender,address(this),assets) (contracts/v6.3/GToken.sol#818)
        State variables written after the call(s):
        - accPnlPerTokenUsed += accPnlDelta (contracts/v6.3/GToken.sol#822)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-1

ERC4626Upgradeable.__ERC4626_init_unchained(IERC20Upgradeable).value (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#42) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

ERC4626Upgradeable.__ERC4626_init_unchained(IERC20Upgradeable) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#40-50) ignores return value by IERC20MetadataUpgradeable(address(asset_)).decimals() (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#42-46)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

GToken.initialize(string,string,GToken.ContractAddresses,uint256,uint256,uint256[2],uint256,uint256,uint256,uint256,uint256)._name (contracts/v6.3/GToken.sol#169) shadows:
        - ERC20Upgradeable._name (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#43) (state variable)
GToken.initialize(string,string,GToken.ContractAddresses,uint256,uint256,uint256[2],uint256,uint256,uint256,uint256,uint256)._symbol (contracts/v6.3/GToken.sol#170) shadows:
        - ERC20Upgradeable._symbol (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#44) (state variable)
GToken.totalSharesBeingWithdrawn(address).owner (contracts/v6.3/GToken.sol#576) shadows:
        - OwnableUpgradeable.owner() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#48-50) (function)
GToken.maxDeposit(address).owner (contracts/v6.3/GToken.sol#478) shadows:
        - OwnableUpgradeable.owner() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#48-50) (function)
GToken.maxRedeem(address).owner (contracts/v6.3/GToken.sol#482) shadows:
        - OwnableUpgradeable.owner() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#48-50) (function)
GToken.maxWithdraw(address).owner (contracts/v6.3/GToken.sol#490) shadows:
        - OwnableUpgradeable.owner() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#48-50) (function)
GToken.withdraw(uint256,address,address).owner (contracts/v6.3/GToken.sol#524) shadows:
        - OwnableUpgradeable.owner() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#48-50) (function)
GToken.redeem(uint256,address,address).owner (contracts/v6.3/GToken.sol#540) shadows:
        - OwnableUpgradeable.owner() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#48-50) (function)
GToken.makeWithdrawRequest(uint256,address).owner (contracts/v6.3/GToken.sol#577) shadows:
        - OwnableUpgradeable.owner() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#48-50) (function)
GToken.makeWithdrawRequest(uint256,address).allowance (contracts/v6.3/GToken.sol#581) shadows:
        - ERC20Upgradeable.allowance(address,address) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#127-129) (function)
        - IERC20Upgradeable.allowance(address,address) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/IERC20Upgradeable.sol#50) (function)
GToken.cancelWithdrawRequest(uint256,address,uint256).owner (contracts/v6.3/GToken.sol#592) shadows:
        - OwnableUpgradeable.owner() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#48-50) (function)
GToken.cancelWithdrawRequest(uint256,address,uint256).allowance (contracts/v6.3/GToken.sol#596) shadows:
        - ERC20Upgradeable.allowance(address,address) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#127-129) (function)
        - IERC20Upgradeable.allowance(address,address) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/IERC20Upgradeable.sol#50) (function)
GToken.unlockDeposit(uint256,address).owner (contracts/v6.3/GToken.sol#681) shadows:
        - OwnableUpgradeable.owner() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#48-50) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

Variable 'ERC4626Upgradeable.__ERC4626_init_unchained(IERC20Upgradeable).value (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#42)' in ERC4626Upgradeable.__ERC4626_init_unchained(IERC20Upgradeable) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#40-50) potentially used before declaration: decimals_ = value (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#43)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables
```

```
Reentrancy in ERC4626Upgradeable._deposit(address,address,uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#215-232):
        External calls:
        - SafeERC20Upgradeable.safeTransferFrom(_asset,caller,address(this),assets) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#228)
        State variables written after the call(s):
        - _mint(receiver,shares) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#229)
                - _balances[account] += amount (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#272)
        - _mint(receiver,shares) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#229)
                - _totalSupply += amount (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#269)
Reentrancy in GToken._executeDiscountAndLock(uint256,uint256,uint256,address) (contracts/v6.3/GToken.sol#644-675):
        External calls:
        - _deposit(sender,address(this),assetsDeposited,shares) (contracts/v6.3/GToken.sol#666)
                - SafeERC20Upgradeable.safeTransferFrom(_asset,caller,address(this),assets) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#228)
                - returndata = address(token).functionCall(data,SafeERC20: low-level call failed) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#110)
                - (success,returndata) = target.call{value: value}(data) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#135)
        External calls sending eth:
        - _deposit(sender,address(this),assetsDeposited,shares) (contracts/v6.3/GToken.sol#666)
                - (success,returndata) = target.call{value: value}(data) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#135)
        State variables written after the call(s):
        - totalDiscounts += assetsDiscount (contracts/v6.3/GToken.sol#668)
        - totalLockedDiscounts += assetsDiscount (contracts/v6.3/GToken.sol#669)
Reentrancy in GToken.deplete(uint256) (contracts/v6.3/GToken.sol#778-798):
        External calls:
        - amountGns = assets.mulDiv(GNS_PRECISION,gnsTokenToAssetsPrice(),MathUpgradeable.Rounding.Up) (contracts/v6.3/GToken.sol#783-787)
                - (success,result) = gnsPriceProvider.addr.staticcall(gnsPriceProvider.signature) (contracts/v6.3/GToken.sol#344-346)
        - IGnsToken(gnsToken).burn(sender,amountGns) (contracts/v6.3/GToken.sol#790)
        State variables written after the call(s):
        - totalDepleted += assets (contracts/v6.3/GToken.sol#792)
        - totalDepletedGns += amountGns (contracts/v6.3/GToken.sol#793)
Reentrancy in GToken.distributeReward(uint256) (contracts/v6.3/GToken.sol#711-720):
        External calls:
        - SafeERC20Upgradeable.safeTransferFrom(_assetIERC20(),sender,address(this),assets) (contracts/v6.3/GToken.sol#713)
        State variables written after the call(s):
        - accRewardsPerToken += assets * PRECISION / totalSupply() (contracts/v6.3/GToken.sol#715)
        - updateShareToAssetsPrice() (contracts/v6.3/GToken.sol#716)
                - shareToAssetsPrice = PRECISION - (uint256(accPnlPerTokenUsed)) + accRewardsPerToken (contracts/v6.3/GToken.sol#414-418)
                - shareToAssetsPrice = PRECISION - (uint256(0)) + accRewardsPerToken (contracts/v6.3/GToken.sol#414-418)
        - totalRewards += assets (contracts/v6.3/GToken.sol#718)
Reentrancy in GToken.receiveAssets(uint256,address) (contracts/v6.3/GToken.sol#751-776):
        External calls:
        - SafeERC20Upgradeable.safeTransferFrom(_assetIERC20(),sender,address(this),assets) (contracts/v6.3/GToken.sol#753)
        State variables written after the call(s):
        - accPnlPerToken -= accPnlDelta (contracts/v6.3/GToken.sol#764)
        - assetsToDeplete += depleteAmount (contracts/v6.3/GToken.sol#759)
        - tryResetDailyAccPnlDelta() (contracts/v6.3/GToken.sol#766)
                - dailyAccPnlDelta = 0 (contracts/v6.3/GToken.sol#395)
        - dailyAccPnlDelta -= accPnlDelta (contracts/v6.3/GToken.sol#767)
        - tryResetDailyAccPnlDelta() (contracts/v6.3/GToken.sol#766)
                - lastDailyAccPnlDeltaReset = block.timestamp (contracts/v6.3/GToken.sol#396)
        - totalClosedPnl -= int256(assetsLessDeplete) (contracts/v6.3/GToken.sol#770)
        - totalLiability -= int256(assetsLessDeplete) (contracts/v6.3/GToken.sol#769)
Reentrancy in GToken.receiveAssets(uint256,address) (contracts/v6.3/GToken.sol#751-776):
        External calls:
        - SafeERC20Upgradeable.safeTransferFrom(_assetIERC20(),sender,address(this),assets) (contracts/v6.3/GToken.sol#753)
        - tryNewOpenPnlRequestOrEpoch() (contracts/v6.3/GToken.sol#772)
                - (success) = address(openTradesPnlFeed).call(abi.encodeWithSignature(newOpenPnlRequestOrEpoch())) (contracts/v6.3/GToken.sol#404-406)
        State variables written after the call(s):
        - tryUpdateCurrentMaxSupply() (contracts/v6.3/GToken.sol#773)
                - currentMaxSupply = totalSupply() * (PRECISION * 100 + maxSupplyIncreaseDailyP) / (PRECISION * 100) (contracts/v6.3/GToken.sol#384-386)
        - tryUpdateCurrentMaxSupply() (contracts/v6.3/GToken.sol#773)
                - lastMaxSupplyUpdate = block.timestamp (contracts/v6.3/GToken.sol#387)
Reentrancy in GToken.refill(uint256) (contracts/v6.3/GToken.sol#800-831):
        External calls:
        - amountGns = assets * GNS_PRECISION / gnsTokenToAssetsPrice() (contracts/v6.3/GToken.sol#811)
                - (success,result) = gnsPriceProvider.addr.staticcall(gnsPriceProvider.signature) (contracts/v6.3/GToken.sol#344-346)
        State variables written after the call(s):
        - dailyMintedGns += amountGns (contracts/v6.3/GToken.sol#812)
Reentrancy in GToken.refill(uint256) (contracts/v6.3/GToken.sol#800-831):
        External calls:
        - amountGns = assets * GNS_PRECISION / gnsTokenToAssetsPrice() (contracts/v6.3/GToken.sol#811)
                - (success,result) = gnsPriceProvider.addr.staticcall(gnsPriceProvider.signature) (contracts/v6.3/GToken.sol#344-346)
        - SafeERC20Upgradeable.safeTransferFrom(_assetIERC20(),sender,address(this),assets) (contracts/v6.3/GToken.sol#818)
        State variables written after the call(s):
        - accPnlPerToken -= accPnlDelta (contracts/v6.3/GToken.sol#821)
        - updateShareToAssetsPrice() (contracts/v6.3/GToken.sol#823)
                - shareToAssetsPrice = PRECISION - (uint256(accPnlPerTokenUsed)) + accRewardsPerToken (contracts/v6.3/GToken.sol#414-418)
                - shareToAssetsPrice = PRECISION - (uint256(0)) + accRewardsPerToken (contracts/v6.3/GToken.sol#414-418)
        - totalRefilled += assets (contracts/v6.3/GToken.sol#825)
        - totalRefilledGns += amountGns (contracts/v6.3/GToken.sol#826)
Reentrancy in GToken.sendAssets(uint256,address) (contracts/v6.3/GToken.sol#723-749):
        External calls:
        - tryNewOpenPnlRequestOrEpoch() (contracts/v6.3/GToken.sol#743)
                - (success) = address(openTradesPnlFeed).call(abi.encodeWithSignature(newOpenPnlRequestOrEpoch())) (contracts/v6.3/GToken.sol#404-406)
        State variables written after the call(s):
        - tryUpdateCurrentMaxSupply() (contracts/v6.3/GToken.sol#744)
                - currentMaxSupply = totalSupply() * (PRECISION * 100 + maxSupplyIncreaseDailyP) / (PRECISION * 100) (contracts/v6.3/GToken.sol#384-386)
        - tryUpdateCurrentMaxSupply() (contracts/v6.3/GToken.sol#744)
                - lastMaxSupplyUpdate = block.timestamp (contracts/v6.3/GToken.sol#387)
Reentrancy in GToken.unlockDeposit(uint256,address) (contracts/v6.3/GToken.sol#677-708):
        External calls:
        - lockedDepositNft.burn(depositId) (contracts/v6.3/GToken.sol#697)
        State variables written after the call(s):
        - _transfer(address(this),receiver,d.shares) (contracts/v6.3/GToken.sol#705)
                - _balances[from] = fromBalance - amount (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#244)
                - _balances[to] += amount (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#247)
        - accPnlPerTokenUsed += accPnlDelta (contracts/v6.3/GToken.sol#699)
        - updateShareToAssetsPrice() (contracts/v6.3/GToken.sol#700)
                - shareToAssetsPrice = PRECISION - (uint256(accPnlPerTokenUsed)) + accRewardsPerToken (contracts/v6.3/GToken.sol#414-418)
                - shareToAssetsPrice = PRECISION - (uint256(0)) + accRewardsPerToken (contracts/v6.3/GToken.sol#414-418)
        - totalLiability += int256(d.assetsDiscount) (contracts/v6.3/GToken.sol#702)
        - totalLockedDiscounts -= d.assetsDiscount (contracts/v6.3/GToken.sol#703)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2

Reentrancy in ERC4626Upgradeable._deposit(address,address,uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#215-232):
        External calls:
        - SafeERC20Upgradeable.safeTransferFrom(_asset,caller,address(this),assets) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#228)
        Event emitted after the call(s):
        - Deposit(caller,receiver,assets,shares) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#231)
        - Transfer(address(0),account,amount) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#274)
                - _mint(receiver,shares) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#229)
Reentrancy in GToken._executeDiscountAndLock(uint256,uint256,uint256,address) (contracts/v6.3/GToken.sol#644-675):
        External calls:
        - _deposit(sender,address(this),assetsDeposited,shares) (contracts/v6.3/GToken.sol#666)
                - SafeERC20Upgradeable.safeTransferFrom(_asset,caller,address(this),assets) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#228)
                - returndata = address(token).functionCall(data,SafeERC20: low-level call failed) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#110)
                - (success,returndata) = target.call{value: value}(data) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#135)
        - lockedDepositNft.mint(receiver,depositId) (contracts/v6.3/GToken.sol#671)
        External calls sending eth:
        - _deposit(sender,address(this),assetsDeposited,shares) (contracts/v6.3/GToken.sol#666)
                - (success,returndata) = target.call{value: value}(data) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#135)
        Event emitted after the call(s):
        - DepositLocked(sender,d.owner,depositId,d) (contracts/v6.3/GToken.sol#673)
Reentrancy in ERC4626Upgradeable._withdraw(address,address,address,uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#237-258):
        External calls:
        - SafeERC20Upgradeable.safeTransfer(_asset,receiver,assets) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#255)
        Event emitted after the call(s):
        - Withdraw(caller,receiver,owner,assets,shares) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#257)
Reentrancy in GToken.deplete(uint256) (contracts/v6.3/GToken.sol#778-798):
        External calls:
        - amountGns = assets.mulDiv(GNS_PRECISION,gnsTokenToAssetsPrice(),MathUpgradeable.Rounding.Up) (contracts/v6.3/GToken.sol#783-787)
                - (success,result) = gnsPriceProvider.addr.staticcall(gnsPriceProvider.signature) (contracts/v6.3/GToken.sol#344-346)
        - IGnsToken(gnsToken).burn(sender,amountGns) (contracts/v6.3/GToken.sol#790)
        - SafeERC20Upgradeable.safeTransfer(_assetIERC20(),sender,assets) (contracts/v6.3/GToken.sol#795)
        Event emitted after the call(s):
        - Depleted(sender,assets,amountGns) (contracts/v6.3/GToken.sol#797)
Reentrancy in GToken.distributeReward(uint256) (contracts/v6.3/GToken.sol#711-720):
        External calls:
        - SafeERC20Upgradeable.safeTransferFrom(_assetIERC20(),sender,address(this),assets) (contracts/v6.3/GToken.sol#713)
        Event emitted after the call(s):
        - RewardDistributed(sender,assets) (contracts/v6.3/GToken.sol#719)
        - ShareToAssetsPriceUpdated(shareToAssetsPrice) (contracts/v6.3/GToken.sol#420)
                - updateShareToAssetsPrice() (contracts/v6.3/GToken.sol#716)
Reentrancy in GToken.receiveAssets(uint256,address) (contracts/v6.3/GToken.sol#751-776):
        External calls:
        - SafeERC20Upgradeable.safeTransferFrom(_assetIERC20(),sender,address(this),assets) (contracts/v6.3/GToken.sol#753)
        Event emitted after the call(s):
        - DailyAccPnlDeltaReset() (contracts/v6.3/GToken.sol#398)
                - tryResetDailyAccPnlDelta() (contracts/v6.3/GToken.sol#766)
Reentrancy in GToken.receiveAssets(uint256,address) (contracts/v6.3/GToken.sol#751-776):
        External calls:
        - SafeERC20Upgradeable.safeTransferFrom(_assetIERC20(),sender,address(this),assets) (contracts/v6.3/GToken.sol#753)
        - tryNewOpenPnlRequestOrEpoch() (contracts/v6.3/GToken.sol#772)
                - (success) = address(openTradesPnlFeed).call(abi.encodeWithSignature(newOpenPnlRequestOrEpoch())) (contracts/v6.3/GToken.sol#404-406)
        Event emitted after the call(s):
        - AssetsReceived(sender,user,assets,assetsLessDeplete) (contracts/v6.3/GToken.sol#775)
        - CurrentMaxSupplyUpdated(currentMaxSupply) (contracts/v6.3/GToken.sol#389)
                - tryUpdateCurrentMaxSupply() (contracts/v6.3/GToken.sol#773)
        - OpenTradesPnlFeedCallFailed() (contracts/v6.3/GToken.sol#408)
                - tryNewOpenPnlRequestOrEpoch() (contracts/v6.3/GToken.sol#772)
Reentrancy in GToken.refill(uint256) (contracts/v6.3/GToken.sol#800-831):
        External calls:
        - amountGns = assets * GNS_PRECISION / gnsTokenToAssetsPrice() (contracts/v6.3/GToken.sol#811)
                - (success,result) = gnsPriceProvider.addr.staticcall(gnsPriceProvider.signature) (contracts/v6.3/GToken.sol#344-346)
        - SafeERC20Upgradeable.safeTransferFrom(_assetIERC20(),sender,address(this),assets) (contracts/v6.3/GToken.sol#818)
        Event emitted after the call(s):
        - ShareToAssetsPriceUpdated(shareToAssetsPrice) (contracts/v6.3/GToken.sol#420)
                - updateShareToAssetsPrice() (contracts/v6.3/GToken.sol#823)
Reentrancy in GToken.refill(uint256) (contracts/v6.3/GToken.sol#800-831):
        External calls:
        - amountGns = assets * GNS_PRECISION / gnsTokenToAssetsPrice() (contracts/v6.3/GToken.sol#811)
                - (success,result) = gnsPriceProvider.addr.staticcall(gnsPriceProvider.signature) (contracts/v6.3/GToken.sol#344-346)
        - SafeERC20Upgradeable.safeTransferFrom(_assetIERC20(),sender,address(this),assets) (contracts/v6.3/GToken.sol#818)
        - IGnsToken(gnsToken).mint(sender,amountGns) (contracts/v6.3/GToken.sol#828)
        Event emitted after the call(s):
        - Refilled(sender,assets,amountGns) (contracts/v6.3/GToken.sol#830)
```

```
Reentrancy in GToken.sendAssets(uint256,address) (contracts/v6.3/GToken.sol#723-749):
        External calls:
        - tryNewOpenPnlRequestOrEpoch() (contracts/v6.3/GToken.sol#743)
            - (success) = address(openTradesPnlFeed).call(abi.encodeWithSignature(newOpenPnlRequestOrEpoch())) (contracts/v6.3/GToken.sol#404-406)
        Event emitted after the call(s):
        - CurrentMaxSupplyUpdated(currentMaxSupply) (contracts/v6.3/GToken.sol#389)
            - tryUpdateCurrentMaxSupply() (contracts/v6.3/GToken.sol#744)
Reentrancy in GToken.sendAssets(uint256,address) (contracts/v6.3/GToken.sol#723-749):
        External calls:
        - tryNewOpenPnlRequestOrEpoch() (contracts/v6.3/GToken.sol#743)
            - (success) = address(openTradesPnlFeed).call(abi.encodeWithSignature(newOpenPnlRequestOrEpoch())) (contracts/v6.3/GToken.sol#404-406)
        - SafeERC20Upgradeable.safeTransfer(_assetIERC20(),receiver,assets) (contracts/v6.3/GToken.sol#746)
        Event emitted after the call(s):
        - AssetsSent(sender,receiver,assets) (contracts/v6.3/GToken.sol#748)
Reentrancy in GToken.tryNewOpenPnlRequestOrEpoch() (contracts/v6.3/GToken.sol#402-410):
        External calls:
        - (success) = address(openTradesPnlFeed).call(abi.encodeWithSignature(newOpenPnlRequestOrEpoch())) (contracts/v6.3/GToken.sol#404-406)
        Event emitted after the call(s):
        - OpenTradesPnlFeedCallFailed() (contracts/v6.3/GToken.sol#408)
Reentrancy in GToken.unlockDeposit(uint256,address) (contracts/v6.3/GToken.sol#677-708):
        External calls:
        - lockedDepositNft.burn(depositId) (contracts/v6.3/GToken.sol#697)
        Event emitted after the call(s):
        - DepositUnlocked(sender,receiver,owner,depositId,d) (contracts/v6.3/GToken.sol#707)
        - ShareToAssetsPriceUpdated(shareToAssetsPrice) (contracts/v6.3/GToken.sol#420)
            - updateShareToAssetsPrice() (contracts/v6.3/GToken.sol#700)
        - Transfer(from,to,amount) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#250)
            - _transfer(address(this),receiver,d.shares) (contracts/v6.3/GToken.sol#705)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

GToken.tryUpdateCurrentMaxSupply() (contracts/v6.3/GToken.sol#382-391) uses timestamp for comparisons
        Dangerous comparisons:
        - block.timestamp - lastMaxSupplyUpdate >= 86400 (contracts/v6.3/GToken.sol#383)
GToken.tryResetDailyAccPnlDelta() (contracts/v6.3/GToken.sol#393-400) uses timestamp for comparisons
        Dangerous comparisons:
        - block.timestamp - lastDailyAccPnlDeltaReset >= 86400 (contracts/v6.3/GToken.sol#394)
GToken.unlockDeposit(uint256,address) (contracts/v6.3/GToken.sol#677-708) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(block.timestamp >= d.atTimestamp + d.lockDuration,NOT_UNLOCKED) (contracts/v6.3/GToken.sol#686)
GToken.refill(uint256) (contracts/v6.3/GToken.sol#800-831) uses timestamp for comparisons
        Dangerous comparisons:
        - block.timestamp - lastDailyMintedGnsReset >= 86400 (contracts/v6.3/GToken.sol#806)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

AddressUpgradeable._revert(bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#206-218) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#211-214)
MathUpgradeable.mulDiv(uint256,uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#55-135) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#66-70)
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#86-93)
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#100-109)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

GToken.gnsTokenToAssetsPrice() (contracts/v6.3/GToken.sol#363-352) compares to a boolean constant:
        -require(bool,string)(success == true,GNS_PRICE_CALL_FAILED) (contracts/v6.3/GToken.sol#348)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#boolean-equality

Different versions of Solidity is used:
        - Version used: ['^0.8.17', '^0.8.0', '^0.8.1', '^0.8.2']
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/interfaces/IERC4626Upgradeable.sol#4)
        - ^0.8.2 (node_modules/@openzeppelin/contracts-upgradeable/proxy/utils/Initializable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/IERC20Upgradeable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/extensions/IERC20MetadataUpgradeable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/extensions/draft-IERC20PermitUpgradeable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#4)
        - ^0.8.1 (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/IERC721.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/utils/introspection/IERC165.sol#4)
        - 0.8.17 (contracts/interfaces/IGToken.sol#2)
        - 0.8.17 (contracts/interfaces/IGnsToken.sol#2)
        - 0.8.17 (contracts/interfaces/INft.sol#4)
        - 0.8.17 (contracts/interfaces/IOpenTradesPnlFeed.sol#2)
        - 0.8.17 (contracts/v6.3/GToken.sol#12)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

AddressUpgradeable.functionCall(address,bytes) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#85-87) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#114-120) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#145-147) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#155-162) is never used and should be removed
AddressUpgradeable.sendValue(address,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#60-65) is never used and should be removed
AddressUpgradeable.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#194-204) is never used and should be removed
ContextUpgradeable._contextSuffixLength() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#18-19) is never used and should be removed
ContextUpgradeable.__Context_init() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#21-22) is never used and should be removed
ContextUpgradeable._msgData() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#27-29) is never used and should be removed
ERC4626Upgradeable._convertToAssets(uint256,MathUpgradeable.Rounding) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#194-198) is never used and should be removed
ERC4626Upgradeable._convertToShares(uint256,MathUpgradeable.Rounding) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#171-177) is never used and should be removed
ERC4626Upgradeable._initialConvertToAssets(uint256,MathUpgradeable.Rounding) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#205-210) is never used and should be removed
ERC4626Upgradeable._initialConvertToShares(uint256,MathUpgradeable.Rounding) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#184-189) is never used and should be removed
ERC4626Upgradeable._isVaultCollateralized() (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#260-262) is never used and should be removed
Initializable._disableInitializers() (node_modules/@openzeppelin/contracts-upgradeable/proxy/utils/Initializable.sol#144-150) is never used and should be removed
Initializable._getInitializedVersion() (node_modules/@openzeppelin/contracts-upgradeable/proxy/utils/Initializable.sol#155-157) is never used and should be removed
Initializable._isInitializing() (node_modules/@openzeppelin/contracts-upgradeable/proxy/utils/Initializable.sol#162-164) is never used and should be removed
MathUpgradeable.average(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#34-37) is never used and should be removed
MathUpgradeable.ceilDiv(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#45-48) is never used and should be removed
MathUpgradeable.log10(uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#258-290) is never used and should be removed
MathUpgradeable.log10(uint256,MathUpgradeable.Rounding) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#296-301) is never used and should be removed
MathUpgradeable.log2(uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#205-241) is never used and should be removed
MathUpgradeable.log2(uint256,MathUpgradeable.Rounding) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#247-252) is never used and should be removed
MathUpgradeable.log256(uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#309-333) is never used and should be removed
MathUpgradeable.log256(uint256,MathUpgradeable.Rounding) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#335-344) is never used and should be removed
MathUpgradeable.max(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#19-21) is never used and should be removed
MathUpgradeable.sqrt(uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#158-189) is never used and should be removed
MathUpgradeable.sqrt(uint256,MathUpgradeable.Rounding) (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#194-199) is never used and should be removed
OwnableUpgradeable._Ownable_init() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#29-31) is never used and should be removed
OwnableUpgradeable.__Ownable_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#33-35) is never used and should be removed
SafeERC20Upgradeable.safeApprove(IERC20Upgradeable,address,uint256) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#46-59) is never used and should be removed
SafeERC20Upgradeable.safeDecreaseAllowance(IERC20Upgradeable,address,uint256) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#70-81) is never used and should be removed
SafeERC20Upgradeable.safeIncreaseAllowance(IERC20Upgradeable,address,uint256) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#61-68) is never used and should be removed
SafeERC20Upgradeable.safePermit(IERC20PermitUpgradeable,address,address,uint256,uint8,bytes32,bytes32) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#83-97) is never used and should b
e removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/interfaces/IERC4626Upgradeable.sol#4) allows old versions
Pragma version^0.8.2 (node_modules/@openzeppelin/contracts-upgradeable/proxy/utils/Initializable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/IERC20Upgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/extensions/IERC20MetadataUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/extensions/draft-IERC20PermitUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/utils/SafeERC20Upgradeable.sol#4) allows old versions
Pragma version^0.8.1 (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/math/MathUpgradeable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/IERC721.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/introspection/IERC165.sol#4) allows old versions
Pragma version0.8.17 (contracts/interfaces/IGToken.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.17 (contracts/interfaces/IGnsToken.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.17 (contracts/interfaces/INft.sol#4) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.17 (contracts/interfaces/IOpenTradesPnlFeed.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.17 (contracts/v6.3/GToken.sol#12) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
solc-0.8.17 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in AddressUpgradeable.sendValue(address,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#60-65):
        - (success) = recipient.call{value: amount}() (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#63)
Low level call in AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#128-137):
        - (success,returndata) = target.call{value: value}(data) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#135)
Low level call in AddressUpgradeable.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#155-162):
        - (success,returndata) = target.staticcall(data) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#160)
Low level call in GToken.gnsTokenToAssetsPrice() (contracts/v6.3/GToken.sol#343-352):
        - (success,result) = gnsPriceProvider.addr.staticcall(gnsPriceProvider.signature) (contracts/v6.3/GToken.sol#344-346)
Low level call in GToken.tryNewOpenPnlRequestOrEpoch() (contracts/v6.3/GToken.sol#402-410):
        - (success) = address(openTradesPnlFeed).call(abi.encodeWithSignature(newOpenPnlRequestOrEpoch())) (contracts/v6.3/GToken.sol#404-406)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function OwnableUpgradeable._Ownable_init() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#29-31) is not in mixedCase
Function OwnableUpgradeable.__Ownable_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#33-35) is not in mixedCase
Variable OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#94) is not in mixedCase
Function ERC20Upgradeable.__ERC20_init(string,string) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#55-57) is not in mixedCase
Function ERC20Upgradeable.__ERC20_init_unchained(string,string) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#59-62) is not in mixedCase
Variable ERC20Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#400) is not in mixedCase
Function ERC4626Upgradeable.__ERC4626_init(IERC20Upgradeable) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#36-38) is not in mixedCase
Function ERC4626Upgradeable.__ERC4626_init_unchained(IERC20Upgradeable) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#40-50) is not in mixedCase
Variable ERC4626Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#268) is not in mixedCase
Function IERC20PermitUpgradeable.DOMAIN_SEPARATOR() (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/extensions/draft-IERC20PermitUpgradeable.sol#59) is not in mixedCase
Function ContextUpgradeable._Context_init() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#18-19) is not in mixedCase
Function ContextUpgradeable.__Context_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#21-22) is not in mixedCase
Variable ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#36) is not in mixedCase
Parameter GToken.initialize(string,string,GToken.ContractAddresses,uint256,uint256,uint256[2],uint256,uint256,uint256,uint256)._name (contracts/v6.3/GToken.sol#169) is not in mixedCase
Parameter GToken.initialize(string,string,GToken.ContractAddresses,uint256,uint256,uint256[2],uint256,uint256,uint256,uint256)._symbol (contracts/v6.3/GToken.sol#170) is not in mixedCase
Parameter GToken.initialize(string,string,GToken.ContractAddresses,uint256,uint256,uint256[2],uint256,uint256,uint256,uint256)._contractAddresses (contracts/v6.3/GToken.sol#171) is not in mixedCase
Parameter GToken.initialize(string,string,GToken.ContractAddresses,uint256,uint256,uint256[2],uint256,uint256,uint256,uint256)._maxAccOpenPnlDelta (contracts/v6.3/GToken.sol#172) is not in mixedCase
Parameter GToken.initialize(string,string,GToken.ContractAddresses,uint256,uint256,uint256[2],uint256,uint256,uint256,uint256)._maxDailyAccPnlDelta (contracts/v6.3/GToken.sol#173) is not in mixedCase
Parameter GToken.initialize(string,string,GToken.ContractAddresses,uint256,uint256,uint256[2],uint256,uint256,uint256,uint256)._withdrawLockThresholdsP (contracts/v6.3/GToken.sol#174) is not in mixedCase
```

```
Parameter GToken.initialize(string,string,GToken.ContractAddresses,uint256,uint256,uint256[2],uint256,uint256,uint256,uint256,uint256)._maxSupplyIncreaseDailyP (contracts/v6.3/GToken.sol#175) is not in mixedCase
Parameter GToken.initialize(string,string,GToken.ContractAddresses,uint256,uint256,uint256[2],uint256,uint256,uint256,uint256,uint256)._loseseBurnP (contracts/v6.3/GToken.sol#176) is not in mixedCase
Parameter GToken.initialize(string,string,GToken.ContractAddresses,uint256,uint256,uint256[2],uint256,uint256,uint256,uint256,uint256)._maxGnsSupplyMintDailyP (contracts/v6.3/GToken.sol#177) is not in mixedCase
Parameter GToken.initialize(string,string,GToken.ContractAddresses,uint256,uint256,uint256[2],uint256,uint256,uint256,uint256,uint256)._maxDiscountP (contracts/v6.3/GToken.sol#178) is not in mixedCase
Parameter GToken.initialize(string,string,GToken.ContractAddresses,uint256,uint256,uint256[2],uint256,uint256,uint256,uint256,uint256)._maxDiscountThreshold (contracts/v6.3/GToken.sol#179) is not in mixedCase
Variable GToken.WITHDRAW_EPOCHS_LOCKS (contracts/v6.3/GToken.sol#44) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#94) is never used in GToken (contracts/v6.3/GToken.sol#14-889)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

renounceOwnership() should be declared external:
        - OwnableUpgradeable.renounceOwnership() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#66-68)
transferOwnership(address) should be declared external:
        - GToken.transferOwnership(address) (contracts/v6.3/GToken.sol#249-253)
        - OwnableUpgradeable.transferOwnership(address) (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#74-77)
name() should be declared external:
        - ERC20Upgradeable.name() (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#67-69)
symbol() should be declared external:
        - ERC20Upgradeable.symbol() (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#75-77)
transfer(address,uint256) should be declared external:
        - ERC20Upgradeable.transfer(address,uint256) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#118-122)
        - GToken.transfer(address,uint256) (contracts/v6.3/GToken.sol#428-436)
approve(address,uint256) should be declared external:
        - ERC20Upgradeable.approve(address,uint256) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#141-145)
transferFrom(address,address,uint256) should be declared external:
        - ERC20Upgradeable.transferFrom(address,address,uint256) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#163-172)
        - GToken.transferFrom(address,address,uint256) (contracts/v6.3/GToken.sol#438-447)
increaseAllowance(address,uint256) should be declared external:
        - ERC20Upgradeable.increaseAllowance(address,uint256) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#186-190)
decreaseAllowance(address,uint256) should be declared external:
        - ERC20Upgradeable.decreaseAllowance(address,uint256) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/ERC20Upgradeable.sol#206-215)
convertToShares(uint256) should be declared external:
        - ERC4626Upgradeable.convertToShares(uint256) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#68-70)
convertToAssets(uint256) should be declared external:
        - ERC4626Upgradeable.convertToAssets(uint256) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#73-75)
deposit(uint256,address) should be declared external:
        - ERC4626Upgradeable.deposit(uint256,address) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#118-125)
        - GToken.deposit(uint256,address) (contracts/v6.3/GToken.sol#495-506)
mint(uint256,address) should be declared external:
        - ERC4626Upgradeable.mint(uint256,address) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#128-135)
        - GToken.mint(uint256,address) (contracts/v6.3/GToken.sol#508-519)
withdraw(uint256,address,address) should be declared external:
        - ERC4626Upgradeable.withdraw(uint256,address,address) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#138-149)
        - GToken.withdraw(uint256,address,address) (contracts/v6.3/GToken.sol#521-535)
redeem(uint256,address,address) should be declared external:
        - ERC4626Upgradeable.redeem(uint256,address,address) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/extensions/ERC4626Upgradeable.sol#152-163)
        - GToken.redeem(uint256,address,address) (contracts/v6.3/GToken.sol#537-551)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

GTokenOpenPnlFeed.sol

```
GTokenOpenPnlFeed.newOpenPnlRequestOrEpoch() (contracts/v6.3/GTokenOpenPnlFeed.sol#225-241) uses a dangerous strict equality:
        - firstRequest = nextEpochValuesLastRequest == 0 (contracts/v6.3/GTokenOpenPnlFeed.sol#226)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dangerous-strict-equalities

Reentrancy in GTokenOpenPnlFeed.startNewEpoch() (contracts/v6.3/GTokenOpenPnlFeed.sol#328-355):
        External calls:
        - finalNewEpochPositiveOpenPnl = gToken.updateAccPnlPerTokenUsed(currentEpochPositiveOpenPnl,uint256(newEpochOpenPnl)) (contracts/v6.3/GTokenOpenPnlFeed.sol#339-342)
        - finalNewEpochPositiveOpenPnl = gToken.updateAccPnlPerTokenUsed(currentEpochPositiveOpenPnl,0) (contracts/v6.3/GTokenOpenPnlFeed.sol#339-342)
        State variables written after the call(s):
        - delete nextEpochValues (contracts/v6.3/GTokenOpenPnlFeed.sol#354)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-1

GTokenOpenPnlFeed.average(int256[]).i (contracts/v6.3/GTokenOpenPnlFeed.sol#390) is a local variable never initialized
ChainlinkClient.buildOperatorRequest(bytes32,bytes4).req (node_modules/@chainlink/contracts/src/v0.8/ChainlinkClient.sol#67) is a local variable never initialized
ChainlinkClient.buildChainlinkRequest(bytes32,address,bytes4).req (node_modules/@chainlink/contracts/src/v0.8/ChainlinkClient.sol#52) is a local variable never initialized
GTokenOpenPnlFeed.resetNextEpochValueRequests().i (contracts/v6.3/GTokenOpenPnlFeed.sol#205) is a local variable never initialized
BufferChainlink.fromBytes(bytes).buf (node_modules/@chainlink/contracts/src/v0.8/vendor/BufferChainlink.sol#51) is a local variable never initialized
GTokenOpenPnlFeed.makeOpenPnlRequest().i (contracts/v6.3/GTokenOpenPnlFeed.sol#265) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

Chainlink.initialize(Chainlink.Request,bytes32,address,bytes4) (node_modules/@chainlink/contracts/src/v0.8/Chainlink.sol#33-44) ignores return value by BufferChainlink.init(self.buf,defaultBufferSize) (node_modules/@chainlink/contracts/src/v0.8/Chainlink.sol#39)
Chainlink.setBuffer(Chainlink.Request,bytes) (node_modules/@chainlink/contracts/src/v0.8/Chainlink.sol#52-55) ignores return value by BufferChainlink.init(self.buf,data.length) (node_modules/@chainlink/contracts/src/v0.8/Chainlink.sol#53)
Chainlink.setBuffer(Chainlink.Request,bytes) (node_modules/@chainlink/contracts/src/v0.8/Chainlink.sol#52-55) ignores return value by BufferChainlink.append(self.buf,data) (node_modules/@chainlink/contracts/src/v0.8/Chainlink.sol#54)
CBORChainlink.encodeFixedNumeric(BufferChainlink.buffer,uint8,uint64) (node_modules/@chainlink/contracts/src/v0.8/vendor/CBORChainlink.sol#21-37) ignores return value by buf.appendUint8(uint8((major << 5) | value)) (node_modules/@chainlink/contracts/src/v0.8/vendor/CBORChainlink.sol#23)
CBORChainlink.encodeFixedNumeric(BufferChainlink.buffer,uint8,uint64) (node_modules/@chainlink/contracts/src/v0.8/vendor/CBORChainlink.sol#21-37) ignores return value by buf.appendUint8(uint8((major << 5) | 24)) (node_modules/@chainlink/contracts/src/v0.8/vendor/CBORChainlink.sol#25)
CBORChainlink.encodeFixedNumeric(BufferChainlink.buffer,uint8,uint64) (node_modules/@chainlink/contracts/src/v0.8/vendor/CBORChainlink.sol#21-37) ignores return value by buf.appendInt(value,1) (node_modules/@chainlink/contracts/src/v0.8/vendor/CBORChainlink.sol#26)
CBORChainlink.encodeFixedNumeric(BufferChainlink.buffer,uint8,uint64) (node_modules/@chainlink/contracts/src/v0.8/vendor/CBORChainlink.sol#21-37) ignores return value by buf.appendUint8(uint8((major << 5) | 25)) (node_modules/@chainlink/contracts/src/v0.8/vendor/CBORChainlink.sol#28)
CBORChainlink.encodeFixedNumeric(BufferChainlink.buffer,uint8,uint64) (node_modules/@chainlink/contracts/src/v0.8/vendor/CBORChainlink.sol#21-37) ignores return value by buf.appendInt(value,2) (node_modules/@chainlink/contracts/src/v0.8/vendor/CBORChainlink.sol#29)
CBORChainlink.encodeFixedNumeric(BufferChainlink.buffer,uint8,uint64) (node_modules/@chainlink/contracts/src/v0.8/vendor/CBORChainlink.sol#21-37) ignores return value by buf.appendUint8(uint8((major << 5) | 26)) (node_modules/@chainlink/contracts/src/v0.8/vendor/CBORChainlink.sol#31)
CBORChainlink.encodeFixedNumeric(BufferChainlink.buffer,uint8,uint64) (node_modules/@chainlink/contracts/src/v0.8/vendor/CBORChainlink.sol#21-37) ignores return value by buf.appendInt(value,4) (node_modules/@chainlink/contracts/src/v0.8/vendor/CBORChainlink.sol#32)
CBORChainlink.encodeFixedNumeric(BufferChainlink.buffer,uint8,uint64) (node_modules/@chainlink/contracts/src/v0.8/vendor/CBORChainlink.sol#21-37) ignores return value by buf.appendUint8(uint8((major << 5) | 27)) (node_modules/@chainlink/contracts/src/v0.8/vendor/CBORChainlink.sol#34)
CBORChainlink.encodeFixedNumeric(BufferChainlink.buffer,uint8,uint64) (node_modules/@chainlink/contracts/src/v0.8/vendor/CBORChainlink.sol#21-37) ignores return value by buf.appendInt(value,8) (node_modules/@chainlink/contracts/src/v0.8/vendor/CBORChainlink.sol#35)
CBORChainlink.encodeIndefiniteLengthType(BufferChainlink.buffer,uint8) (node_modules/@chainlink/contracts/src/v0.8/vendor/CBORChainlink.sol#39-41) ignores return value by buf.appendUint8(uint8((major << 5) | 31)) (node_modules/@chainlink/contracts/src/v0.8/vendor/CBORChainlink.sol#40)
CBORChainlink.encodeBytes(BufferChainlink.buffer,bytes) (node_modules/@chainlink/contracts/src/v0.8/vendor/CBORChainlink.sol#63-66) ignores return value by buf.append(value) (node_modules/@chainlink/contracts/src/v0.8/vendor/CBORChainlink.sol#65)
CBORChainlink.encodeBigNum(BufferChainlink.buffer,uint256) (node_modules/@chainlink/contracts/src/v0.8/vendor/CBORChainlink.sol#68-71) ignores return value by buf.appendUint8(uint8((MAJOR_TYPE_TAG << 5) | TAG_TYPE_BIGNUM)) (node_modules/@chainlink/contracts/src/v0.8/vendor/CBORChainlink.sol#69)
CBORChainlink.encodeSignedBigNum(BufferChainlink.buffer,int256) (node_modules/@chainlink/contracts/src/v0.8/vendor/CBORChainlink.sol#73-76) ignores return value by buf.appendUint8(uint8((MAJOR_TYPE_TAG << 5) | TAG_TYPE_NEGATIVE_BIGNUM)) (node_modules/@chainlink/contracts/src/v0.8/vendor/CBORChainlink.sol#74)
CBORChainlink.encodeString(BufferChainlink.buffer,string) (node_modules/@chainlink/contracts/src/v0.8/vendor/CBORChainlink.sol#78-81) ignores return value by buf.append(bytes(value)) (node_modules/@chainlink/contracts/src/v0.8/vendor/CBORChainlink.sol#80)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

ENSInterface.setSubnodeOwner(bytes32,bytes32,address).owner (node_modules/@chainlink/contracts/src/v0.8/interfaces/ENSInterface.sol#20) shadows:
        - ENSInterface.owner(bytes32) (node_modules/@chainlink/contracts/src/v0.8/interfaces/ENSInterface.sol#29) (function)
ENSInterface.setResolver(bytes32,address).resolver (node_modules/@chainlink/contracts/src/v0.8/interfaces/ENSInterface.sol#23) shadows:
        - ENSInterface.resolver(bytes32) (node_modules/@chainlink/contracts/src/v0.8/interfaces/ENSInterface.sol#31) (function)
ENSInterface.setOwner(bytes32,address).owner (node_modules/@chainlink/contracts/src/v0.8/interfaces/ENSInterface.sol#25) shadows:
        - ENSInterface.owner(bytes32) (node_modules/@chainlink/contracts/src/v0.8/interfaces/ENSInterface.sol#29) (function)
ENSInterface.setTTL(bytes32,uint64).ttl (node_modules/@chainlink/contracts/src/v0.8/interfaces/ENSInterface.sol#27) shadows:
        - ENSInterface.ttl(bytes32) (node_modules/@chainlink/contracts/src/v0.8/interfaces/ENSInterface.sol#33) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

Reentrancy in GTokenOpenPnlFeed.forceNewEpoch() (contracts/v6.3/GTokenOpenPnlFeed.sol#217-222):
        External calls:
        - newEpoch = startNewEpoch() (contracts/v6.3/GTokenOpenPnlFeed.sol#220)
                - finalNewEpochPositiveOpenPnl = gToken.updateAccPnlPerTokenUsed(currentEpochPositiveOpenPnl,uint256(newEpochOpenPnl)) (contracts/v6.3/GTokenOpenPnlFeed.sol#339-342)
                - finalNewEpochPositiveOpenPnl = gToken.updateAccPnlPerTokenUsed(currentEpochPositiveOpenPnl,0) (contracts/v6.3/GTokenOpenPnlFeed.sol#339-342)
        Event emitted after the call(s):
        - NewEpochForced(newEpoch) (contracts/v6.3/GTokenOpenPnlFeed.sol#221)
Reentrancy in GTokenOpenPnlFeed.startNewEpoch() (contracts/v6.3/GTokenOpenPnlFeed.sol#328-355):
        External calls:
        - finalNewEpochPositiveOpenPnl = gToken.updateAccPnlPerTokenUsed(currentEpochPositiveOpenPnl,uint256(newEpochOpenPnl)) (contracts/v6.3/GTokenOpenPnlFeed.sol#339-342)
        - finalNewEpochPositiveOpenPnl = gToken.updateAccPnlPerTokenUsed(currentEpochPositiveOpenPnl,0) (contracts/v6.3/GTokenOpenPnlFeed.sol#339-342)
        Event emitted after the call(s):
        - NewEpoch(newEpoch,lastRequestId,nextEpochValues,newEpochOpenPnl,finalNewEpochPositiveOpenPnl) (contracts/v6.3/GTokenOpenPnlFeed.sol#346-352)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

GTokenOpenPnlFeed.forceNewEpoch() (contracts/v6.3/GTokenOpenPnlFeed.sol#217-222) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(block.timestamp - gToken.currentEpochStart() >= requestsStart + requestsEvery * requestsCount,TOO_EARLY) (contracts/v6.3/GTokenOpenPnlFeed.sol#218-219)
GTokenOpenPnlFeed.newOpenPnlRequestOrEpoch() (contracts/v6.3/GTokenOpenPnlFeed.sol#225-241) uses timestamp for comparisons
        Dangerous comparisons:
        - firstRequest = nextEpochValuesLastRequest == 0 (contracts/v6.3/GTokenOpenPnlFeed.sol#226)
        - firstRequest && block.timestamp - gToken.currentEpochStart() >= requestsStart (contracts/v6.3/GTokenOpenPnlFeed.sol#228-229)
        - ! firstRequest && block.timestamp - nextEpochValuesLastRequest >= requestsEvery (contracts/v6.3/GTokenOpenPnlFeed.sol#232-233)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

BufferChainlink.init(BufferChainlink.buffer,uint256) (node_modules/@chainlink/contracts/src/v0.8/vendor/BufferChainlink.sol#29-42) uses assembly
        - INLINE ASM (node_modules/@chainlink/contracts/src/v0.8/vendor/BufferChainlink.sol#35-40)
BufferChainlink.truncate(BufferChainlink.buffer) (node_modules/@chainlink/contracts/src/v0.8/vendor/BufferChainlink.sol#75-81) uses assembly
        - INLINE ASM (node_modules/@chainlink/contracts/src/v0.8/vendor/BufferChainlink.sol#76-79)
BufferChainlink.write(BufferChainlink.buffer,uint256,bytes,uint256) (node_modules/@chainlink/contracts/src/v0.8/vendor/BufferChainlink.sol#92-140) uses assembly
        - INLINE ASM (node_modules/@chainlink/contracts/src/v0.8/vendor/BufferChainlink.sol#106-118)
        - INLINE ASM (node_modules/@chainlink/contracts/src/v0.8/vendor/BufferChainlink.sol#122-124)
        - INLINE ASM (node_modules/@chainlink/contracts/src/v0.8/vendor/BufferChainlink.sol#132-136)
BufferChainlink.writeUint8(BufferChainlink.buffer,uint256,uint8) (node_modules/@chainlink/contracts/src/v0.8/vendor/BufferChainlink.sol#177-200) uses assembly
        - INLINE ASM (node_modules/@chainlink/contracts/src/v0.8/vendor/BufferChainlink.sol#186-198)
BufferChainlink.write(BufferChainlink.buffer,uint256,bytes32,uint256) (node_modules/@chainlink/contracts/src/v0.8/vendor/BufferChainlink.sol#222-249) uses assembly
        - INLINE ASM (node_modules/@chainlink/contracts/src/v0.8/vendor/BufferChainlink.sol#236-246)
BufferChainlink.writeInt(BufferChainlink.buffer,uint256,uint256,uint256) (node_modules/@chainlink/contracts/src/v0.8/vendor/BufferChainlink.sol#298-321) uses assembly
        - INLINE ASM (node_modules/@chainlink/contracts/src/v0.8/vendor/BufferChainlink.sol#309-319)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Different versions of Solidity is used:
        - Version used: ['^0.8.17', '>=0.4.19', '^0.8.0']
        - ^0.8.0 (node_modules/@chainlink/contracts/src/v0.8/Chainlink.sol#2)
        - ^0.8.0 (node_modules/@chainlink/contracts/src/v0.8/ChainlinkClient.sol#2)
        - ^0.8.0 (node_modules/@chainlink/contracts/src/v0.8/interfaces/ChainlinkRequestInterface.sol#2)
        - ^0.8.0 (node_modules/@chainlink/contracts/src/v0.8/interfaces/ENSInterface.sol#2)
        - ^0.8.0 (node_modules/@chainlink/contracts/src/v0.8/interfaces/LinkTokenInterface.sol#2)
        - ^0.8.0 (node_modules/@chainlink/contracts/src/v0.8/interfaces/OperatorInterface.sol#2)
        - ^0.8.0 (node_modules/@chainlink/contracts/src/v0.8/interfaces/OracleInterface.sol#2)
        - ^0.8.0 (node_modules/@chainlink/contracts/src/v0.8/interfaces/PointerInterface.sol#2)
        - ^0.8.0 (node_modules/@chainlink/contracts/src/v0.8/vendor/BufferChainlink.sol#2)
        - >=0.4.19 (node_modules/@chainlink/contracts/src/v0.8/vendor/CBORChainlink.sol#2)
        - ^0.8.0 (node_modules/@chainlink/contracts/src/v0.8/vendor/ENSResolver.sol#2)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4)
```

```
        - 0.8.17 (contracts/interfaces/IGToken.sol#2)
        - 0.8.17 (contracts/interfaces/IOpenTradesPnlFeed.sol#2)
        - 0.8.17 (contracts/interfaces/IOwnable.sol#2)
        - 0.8.17 (contracts/v6.3/GTokenOpenPnlFeed.sol#9)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

BufferChainlink.append(BufferChainlink.buffer,bytes) (node_modules/@chainlink/contracts/src/v0.8/vendor/BufferChainlink.sol#165-167) is never used and should be removed
BufferChainlink.append(BufferChainlink.buffer,bytes,uint256) (node_modules/@chainlink/contracts/src/v0.8/vendor/BufferChainlink.sol#150-156) is never used and should be removed
BufferChainlink.appendBytes20(BufferChainlink.buffer,bytes20) (node_modules/@chainlink/contracts/src/v0.8/vendor/BufferChainlink.sol#274-276) is never used and should be removed
BufferChainlink.appendBytes32(BufferChainlink.buffer,bytes32) (node_modules/@chainlink/contracts/src/v0.8/vendor/BufferChainlink.sol#285-287) is never used and should be removed
BufferChainlink.appendInt(BufferChainlink.buffer,uint256,uint256) (node_modules/@chainlink/contracts/src/v0.8/vendor/BufferChainlink.sol#330-336) is never used and should be removed
BufferChainlink.appendUint8(BufferChainlink.buffer,uint8) (node_modules/@chainlink/contracts/src/v0.8/vendor/BufferChainlink.sol#208-211) is never used and should be removed
BufferChainlink.fromBytes(bytes) (node_modules/@chainlink/contracts/src/v0.8/vendor/BufferChainlink.sol#50-55) is never used and should be removed
BufferChainlink.max(uint256,uint256) (node_modules/@chainlink/contracts/src/v0.8/vendor/BufferChainlink.sol#63-68) is never used and should be removed
BufferChainlink.resize(BufferChainlink.buffer,uint256) (node_modules/@chainlink/contracts/src/v0.8/vendor/BufferChainlink.sol#57-61) is never used and should be removed
BufferChainlink.truncate(BufferChainlink.buffer) (node_modules/@chainlink/contracts/src/v0.8/vendor/BufferChainlink.sol#75-81) is never used and should be removed
BufferChainlink.write(BufferChainlink.buffer,uint256,bytes,uint256) (node_modules/@chainlink/contracts/src/v0.8/vendor/BufferChainlink.sol#92-140) is never used and should be removed
BufferChainlink.write(BufferChainlink.buffer,uint256,bytes32,uint256) (node_modules/@chainlink/contracts/src/v0.8/vendor/BufferChainlink.sol#222-249) is never used and should be removed
BufferChainlink.writeBytes20(BufferChainlink.buffer,uint256,bytes20) (node_modules/@chainlink/contracts/src/v0.8/vendor/BufferChainlink.sol#259-265) is never used and should be removed
BufferChainlink.writeInt(BufferChainlink.buffer,uint256,uint256,uint256) (node_modules/@chainlink/contracts/src/v0.8/vendor/BufferChainlink.sol#298-321) is never used and should be removed
BufferChainlink.writeUint8(BufferChainlink.buffer,uint256,uint8) (node_modules/@chainlink/contracts/src/v0.8/vendor/BufferChainlink.sol#177-200) is never used and should be removed
CBORChainlink.encodeBigNum(BufferChainlink.buffer,uint256) (node_modules/@chainlink/contracts/src/v0.8/vendor/CBORChainlink.sol#68-71) is never used and should be removed
CBORChainlink.encodeBytes(BufferChainlink.buffer,bytes) (node_modules/@chainlink/contracts/src/v0.8/vendor/CBORChainlink.sol#63-66) is never used and should be removed
CBORChainlink.encodeFixedNumeric(BufferChainlink.buffer,uint8,uint64) (node_modules/@chainlink/contracts/src/v0.8/vendor/CBORChainlink.sol#21-37) is never used and should be removed
CBORChainlink.encodeIndefiniteLengthType(BufferChainlink.buffer,uint8) (node_modules/@chainlink/contracts/src/v0.8/vendor/CBORChainlink.sol#39-41) is never used and should be removed
CBORChainlink.encodeInt(BufferChainlink.buffer,int256) (node_modules/@chainlink/contracts/src/v0.8/vendor/CBORChainlink.sol#51-61) is never used and should be removed
CBORChainlink.encodeSignedBigNum(BufferChainlink.buffer,int256) (node_modules/@chainlink/contracts/src/v0.8/vendor/CBORChainlink.sol#73-76) is never used and should be removed
CBORChainlink.encodeString(BufferChainlink.buffer,string) (node_modules/@chainlink/contracts/src/v0.8/vendor/CBORChainlink.sol#78-81) is never used and should be removed
CBORChainlink.encodeUInt(BufferChainlink.buffer,uint256) (node_modules/@chainlink/contracts/src/v0.8/vendor/CBORChainlink.sol#43-49) is never used and should be removed
CBORChainlink.endSequence(BufferChainlink.buffer) (node_modules/@chainlink/contracts/src/v0.8/vendor/CBORChainlink.sol#91-93) is never used and should be removed
CBORChainlink.startArray(BufferChainlink.buffer) (node_modules/@chainlink/contracts/src/v0.8/vendor/CBORChainlink.sol#83-85) is never used and should be removed
CBORChainlink.startMap(BufferChainlink.buffer) (node_modules/@chainlink/contracts/src/v0.8/vendor/CBORChainlink.sol#87-89) is never used and should be removed
Chainlink.add(Chainlink.Request,string,string) (node_modules/@chainlink/contracts/src/v0.8/Chainlink.sol#63-70) is never used and should be removed
Chainlink.addBytes(Chainlink.Request,string,bytes) (node_modules/@chainlink/contracts/src/v0.8/Chainlink.sol#78-85) is never used and should be removed
Chainlink.addInt(Chainlink.Request,string,int256) (node_modules/@chainlink/contracts/src/v0.8/Chainlink.sol#93-100) is never used and should be removed
Chainlink.addStringArray(Chainlink.Request,string,string[]) (node_modules/@chainlink/contracts/src/v0.8/Chainlink.sol#123-134) is never used and should be removed
Chainlink.addUint(Chainlink.Request,string,uint256) (node_modules/@chainlink/contracts/src/v0.8/Chainlink.sol#109-115) is never used and should be removed
Chainlink.setBuffer(Chainlink.Request,bytes) (node_modules/@chainlink/contracts/src/v0.8/Chainlink.sol#52-55) is never used and should be removed
ChainlinkClient.addChainlinkExternalRequest(address,bytes32) (node_modules/@chainlink/contracts/src/v0.8/ChainlinkClient.sol#253-255) is never used and should be removed
ChainlinkClient.buildOperatorRequest(bytes32,bytes4) (node_modules/@chainlink/contracts/src/v0.8/ChainlinkClient.sol#62-68) is never used and should be removed
ChainlinkClient.cancelChainlinkRequest(bytes32,uint256,bytes4,uint256) (node_modules/@chainlink/contracts/src/v0.8/ChainlinkClient.sol#186-196) is never used and should be removed
ChainlinkClient.chainlinkOracleAddress() (node_modules/@chainlink/contracts/src/v0.8/ChainlinkClient.sol#243-245) is never used and should be removed
ChainlinkClient.getNextRequestCount() (node_modules/@chainlink/contracts/src/v0.8/ChainlinkClient.sol#203-205) is never used and should be removed
ChainlinkClient.sendChainlinkRequest(Chainlink.Request,uint256) (node_modules/@chainlink/contracts/src/v0.8/ChainlinkClient.sol#78-80) is never used and should be removed
ChainlinkClient.sendOperatorRequest(Chainlink.Request,uint256) (node_modules/@chainlink/contracts/src/v0.8/ChainlinkClient.sol#121-123) is never used and should be removed
ChainlinkClient.sendOperatorRequestTo(address,Chainlink.Request,uint256) (node_modules/@chainlink/contracts/src/v0.8/ChainlinkClient.sol#136-154) is never used and should be removed
ChainlinkClient.setChainlinkOracle(address) (node_modules/@chainlink/contracts/src/v0.8/ChainlinkClient.sol#211-213) is never used and should be removed
ChainlinkClient.setPublicChainlinkToken() (node_modules/@chainlink/contracts/src/v0.8/ChainlinkClient.sol#227-229) is never used and should be removed
ChainlinkClient.updateChainlinkOracleWithENS() (node_modules/@chainlink/contracts/src/v0.8/ChainlinkClient.sol#276-280) is never used and should be removed
ChainlinkClient.useChainlinkWithENS(address,bytes32) (node_modules/@chainlink/contracts/src/v0.8/ChainlinkClient.sol#263-270) is never used and should be removed
ChainlinkClient.validateChainlinkCallback(bytes32) (node_modules/@chainlink/contracts/src/v0.8/ChainlinkClient.sol#287-293) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version^0.8.0 (node_modules/@chainlink/contracts/src/v0.8/Chainlink.sol#2) allows old versions
Pragma version^0.8.0 (node_modules/@chainlink/contracts/src/v0.8/ChainlinkClient.sol#2) allows old versions
Pragma version^0.8.0 (node_modules/@chainlink/contracts/src/v0.8/interfaces/ChainlinkRequestInterface.sol#2) allows old versions
Pragma version^0.8.0 (node_modules/@chainlink/contracts/src/v0.8/interfaces/ENSInterface.sol#2) allows old versions
Pragma version^0.8.0 (node_modules/@chainlink/contracts/src/v0.8/interfaces/LinkTokenInterface.sol#2) allows old versions
Pragma version^0.8.0 (node_modules/@chainlink/contracts/src/v0.8/interfaces/OperatorInterface.sol#2) allows old versions
Pragma version^0.8.0 (node_modules/@chainlink/contracts/src/v0.8/interfaces/OracleInterface.sol#2) allows old versions
Pragma version^0.8.0 (node_modules/@chainlink/contracts/src/v0.8/interfaces/PointerInterface.sol#2) allows old versions
Pragma version^0.8.0 (node_modules/@chainlink/contracts/src/v0.8/vendor/BufferChainlink.sol#2) allows old versions
Pragma version^0.4.19 (node_modules/@chainlink/contracts/src/v0.8/vendor/CBORChainlink.sol#2) allows old versions
Pragma version^0.8.0 (node_modules/@chainlink/contracts/src/v0.8/vendor/ENSResolver.sol#2) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4) allows old versions
Pragma version0.8.17 (contracts/interfaces/IGToken.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.17 (contracts/interfaces/IOpenTradesPnlFeed.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.17 (contracts/interfaces/IOwnable.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.17 (contracts/v6.3/GTokenOpenPnlFeed.sol#9) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
solc-0.8.17 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Constant Chainlink.defaultBufferSize (node_modules/@chainlink/contracts/src/v0.8/Chainlink.sol#12) is not in UPPER_CASE_WITH_UNDERSCORES
Variable ChainlinkClient._ens (node_modules/@chainlink/contracts/src/v0.8/ChainlinkClient.sol#29) is not in mixedCase
Variable ChainlinkClient._ensNode (node_modules/@chainlink/contracts/src/v0.8/ChainlinkClient.sol#30) is not in mixedCase
Variable ChainlinkClient._link (node_modules/@chainlink/contracts/src/v0.8/ChainlinkClient.sol#31) is not in mixedCase
Variable ChainlinkClient._oracle (node_modules/@chainlink/contracts/src/v0.8/ChainlinkClient.sol#32) is not in mixedCase
Variable ChainlinkClient._requestCount (node_modules/@chainlink/contracts/src/v0.8/ChainlinkClient.sol#33) is not in mixedCase
Variable ChainlinkClient._pendingRequests (node_modules/@chainlink/contracts/src/v0.8/ChainlinkClient.sol#34) is not in mixedCase
Struct BufferChainlink.buffer (node_modules/@chainlink/contracts/src/v0.8/vendor/BufferChainlink.sol#18-21) is not in CapWords
Parameter GTokenOpenPnlFeed.updateOracle(uint256,address)._index (contracts/v6.3/GTokenOpenPnlFeed.sol#176) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Variable GTokenOpenPnlFeed.MIN_ANSWERS (contracts/v6.3/GTokenOpenPnlFeed.sol#16) is too similar to GTokenOpenPnlFeed.constructor(address,IGToken,address[],bytes32,uint256)._minAnswers (contracts/v6.3/GTokenOpenPnlFeed.sol#102)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-are-too-similar

CBORChainlink.encodeInt(BufferChainlink.buffer,int256) (node_modules/@chainlink/contracts/src/v0.8/vendor/CBORChainlink.sol#51-61) uses literals with too many digits:
        - value < -0x10000000000000000 (node_modules/@chainlink/contracts/src/v0.8/vendor/CBORChainlink.sol#52)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits

ChainlinkClient.LINK_DIVISIBILITY (node_modules/@chainlink/contracts/src/v0.8/ChainlinkClient.sol#20) is never used in GTokenOpenPnlFeed (contracts/v6.3/GTokenOpenPnlFeed.sol#11-397)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

addr(bytes32) should be declared external:
        - ENSResolver.addr(bytes32) (node_modules/@chainlink/contracts/src/v0.8/vendor/ENSResolver.sol#5)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

TWAPPriceGetter.sol

```
FullMath.mulDiv(uint256,uint256,uint256) (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#14-108) performs a multiplication on the result of a division:
        -denominator = denominator / twos (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#68)
        -inv = (3 * denominator) ^ 2 (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#88)
FullMath.mulDiv(uint256,uint256,uint256) (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#14-108) performs a multiplication on the result of a division:
        -denominator = denominator / twos (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#68)
        -inv *= 2 - denominator * inv (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#92)
FullMath.mulDiv(uint256,uint256,uint256) (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#14-108) performs a multiplication on the result of a division:
        -denominator = denominator / twos (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#68)
        -inv *= 2 - denominator * inv (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#93)
FullMath.mulDiv(uint256,uint256,uint256) (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#14-108) performs a multiplication on the result of a division:
        -denominator = denominator / twos (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#68)
        -inv *= 2 - denominator * inv (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#94)
FullMath.mulDiv(uint256,uint256,uint256) (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#14-108) performs a multiplication on the result of a division:
        -denominator = denominator / twos (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#68)
        -inv *= 2 - denominator * inv (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#95)
FullMath.mulDiv(uint256,uint256,uint256) (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#14-108) performs a multiplication on the result of a division:
        -denominator = denominator / twos (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#68)
        -inv *= 2 - denominator * inv (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#96)
FullMath.mulDiv(uint256,uint256,uint256) (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#14-108) performs a multiplication on the result of a division:
        -denominator = denominator / twos (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#68)
        -inv *= 2 - denominator * inv (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#97)
FullMath.mulDiv(uint256,uint256,uint256) (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#14-108) performs a multiplication on the result of a division:
        -prod0 = prod0 / twos (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#73)
        -result = prod0 * inv (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#105)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#divide-before-multiply

IUniswapV3PoolState.positions(bytes32).liquidity (node_modules/@uniswap/v3-core/contracts/interfaces/pool/IUniswapV3PoolState.sol#93) shadows:
        - IUniswapV3PoolState.liquidity() (node_modules/@uniswap/v3-core/contracts/interfaces/pool/IUniswapV3PoolState.sol#49) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

FullMath.mulDiv(uint256,uint256,uint256) (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#14-108) uses assembly
        - INLINE ASM (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#27-31)
        - INLINE ASM (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#36-38)
        - INLINE ASM (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#53-55)
        - INLINE ASM (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#57-60)
        - INLINE ASM (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#67-69)
        - INLINE ASM (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#72-74)
        - INLINE ASM (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#79-80)
TickMath.getTickAtSqrtRatio(uint160) (node_modules/@uniswap/v3-core/contracts/libraries/TickMath.sol#68-213) uses assembly
        - INLINE ASM (node_modules/@uniswap/v3-core/contracts/libraries/TickMath.sol#77-81)
        - INLINE ASM (node_modules/@uniswap/v3-core/contracts/libraries/TickMath.sol#82-86)
        - INLINE ASM (node_modules/@uniswap/v3-core/contracts/libraries/TickMath.sol#87-91)
        - INLINE ASM (node_modules/@uniswap/v3-core/contracts/libraries/TickMath.sol#92-96)
        - INLINE ASM (node_modules/@uniswap/v3-core/contracts/libraries/TickMath.sol#97-101)
        - INLINE ASM (node_modules/@uniswap/v3-core/contracts/libraries/TickMath.sol#102-106)
        - INLINE ASM (node_modules/@uniswap/v3-core/contracts/libraries/TickMath.sol#107-111)
        - INLINE ASM (node_modules/@uniswap/v3-core/contracts/libraries/TickMath.sol#112-115)
        - INLINE ASM (node_modules/@uniswap/v3-core/contracts/libraries/TickMath.sol#122-127)
        - INLINE ASM (node_modules/@uniswap/v3-core/contracts/libraries/TickMath.sol#128-133)
        - INLINE ASM (node_modules/@uniswap/v3-core/contracts/libraries/TickMath.sol#134-139)
        - INLINE ASM (node_modules/@uniswap/v3-core/contracts/libraries/TickMath.sol#140-145)
        - INLINE ASM (node_modules/@uniswap/v3-core/contracts/libraries/TickMath.sol#146-151)
        - INLINE ASM (node_modules/@uniswap/v3-core/contracts/libraries/TickMath.sol#152-157)
        - INLINE ASM (node_modules/@uniswap/v3-core/contracts/libraries/TickMath.sol#158-163)
        - INLINE ASM (node_modules/@uniswap/v3-core/contracts/libraries/TickMath.sol#164-169)
        - INLINE ASM (node_modules/@uniswap/v3-core/contracts/libraries/TickMath.sol#170-175)
        - INLINE ASM (node_modules/@uniswap/v3-core/contracts/libraries/TickMath.sol#176-181)
        - INLINE ASM (node_modules/@uniswap/v3-core/contracts/libraries/TickMath.sol#182-187)
        - INLINE ASM (node_modules/@uniswap/v3-core/contracts/libraries/TickMath.sol#188-193)
        - INLINE ASM (node_modules/@uniswap/v3-core/contracts/libraries/TickMath.sol#194-199)
        - INLINE ASM (node_modules/@uniswap/v3-core/contracts/libraries/TickMath.sol#200-204)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
```

```
Different versions of Solidity is used:
        - Version used: ['^0.8.17', '>=0.4.0', '>=0.5.0', '^0.8.0']
        - >=0.5.0 (node_modules/@uniswap/v3-core/contracts/interfaces/IUniswapV3Pool.sol#2)
        - >=0.5.0 (node_modules/@uniswap/v3-core/contracts/interfaces/pool/IUniswapV3PoolActions.sol#2)
        - >=0.5.0 (node_modules/@uniswap/v3-core/contracts/interfaces/pool/IUniswapV3PoolDerivedState.sol#2)
        - >=0.5.0 (node_modules/@uniswap/v3-core/contracts/interfaces/pool/IUniswapV3PoolErrors.sol#2)
        - >=0.5.0 (node_modules/@uniswap/v3-core/contracts/interfaces/pool/IUniswapV3PoolEvents.sol#2)
        - >=0.5.0 (node_modules/@uniswap/v3-core/contracts/interfaces/pool/IUniswapV3PoolImmutables.sol#2)
        - >=0.5.0 (node_modules/@uniswap/v3-core/contracts/interfaces/pool/IUniswapV3PoolOwnerActions.sol#2)
        - >=0.5.0 (node_modules/@uniswap/v3-core/contracts/interfaces/pool/IUniswapV3PoolState.sol#2)
        - >=0.4.0 (node_modules/@uniswap/v3-core/contracts/libraries/FixedPoint96.sol#2)
        - ^0.8.0 (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#2)
        - ^0.8.0 (node_modules/@uniswap/v3-core/contracts/libraries/TickMath.sol#2)
        - 0.8.17 (contracts/v6.3/TWAPPriceGetter.sol#7)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

FullMath.mulDivRoundingUp(uint256,uint256,uint256) (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#115-127) is never used and should be removed
TWAPPriceGetter._updateTwapInterval(uint32) (contracts/v6.3/TWAPPriceGetter.sol#55-59) is never used and should be removed
TWAPPriceGetter._updateUniV3Pool(IUniswapV3Pool) (contracts/v6.3/TWAPPriceGetter.sol#48-53) is never used and should be removed
TickMath.getTickAtSqrtRatio(uint160) (node_modules/@uniswap/v3-core/contracts/libraries/TickMath.sol#68-213) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version>=0.5.0 (node_modules/@uniswap/v3-core/contracts/interfaces/IUniswapV3Pool.sol#2) allows old versions
Pragma version>=0.5.0 (node_modules/@uniswap/v3-core/contracts/interfaces/pool/IUniswapV3PoolActions.sol#2) allows old versions
Pragma version>=0.5.0 (node_modules/@uniswap/v3-core/contracts/interfaces/pool/IUniswapV3PoolDerivedState.sol#2) allows old versions
Pragma version>=0.5.0 (node_modules/@uniswap/v3-core/contracts/interfaces/pool/IUniswapV3PoolErrors.sol#2) allows old versions
Pragma version>=0.5.0 (node_modules/@uniswap/v3-core/contracts/interfaces/pool/IUniswapV3PoolEvents.sol#2) allows old versions
Pragma version>=0.5.0 (node_modules/@uniswap/v3-core/contracts/interfaces/pool/IUniswapV3PoolImmutables.sol#2) allows old versions
Pragma version>=0.5.0 (node_modules/@uniswap/v3-core/contracts/interfaces/pool/IUniswapV3PoolOwnerActions.sol#2) allows old versions
Pragma version>=0.5.0 (node_modules/@uniswap/v3-core/contracts/interfaces/pool/IUniswapV3PoolState.sol#2) allows old versions
Pragma version>=0.4.0 (node_modules/@uniswap/v3-core/contracts/libraries/FixedPoint96.sol#2) allows old versions
Pragma version^0.8.0 (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#2) allows old versions
Pragma version^0.8.0 (node_modules/@uniswap/v3-core/contracts/libraries/TickMath.sol#2) allows old versions
Pragma version0.8.17 (contracts/v6.3/TWAPPriceGetter.sol#7) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
solc-0.8.17 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Variable IUniswapV3PoolActions.collect(address,int24,int24,uint128,uint128).amount0Requested (node_modules/@uniswap/v3-core/contracts/interfaces/pool/IUniswapV3PoolActions.sol#47) is too similar to IUniswapV3PoolActions.collect(address,int24,int24,uint128,uint128).amount1Requested (node_modules/@uniswap/v3-core/contracts/interfaces/pool/IUniswapV3PoolActions.sol#48)
Variable IUniswapV3PoolOwnerActions.collectProtocol(address,uint128,uint128).amount0Requested (node_modules/@uniswap/v3-core/contracts/interfaces/pool/IUniswapV3PoolOwnerActions.sol#20) is too similar to IUniswapV3PoolOwnerActions.collectProtocol(address,uint128,uint128).amount1Requested (node_modules/@uniswap/v3-core/contracts/interfaces/pool/IUniswapV3PoolOwnerActions.sol#21)
Variable IUniswapV3PoolActions.collect(address,int24,int24,uint128,uint128).amount0Requested (node_modules/@uniswap/v3-core/contracts/interfaces/pool/IUniswapV3PoolActions.sol#47) is too similar to IUniswapV3PoolOwnerActions.collectProtocol(address,uint128,uint128).amount1Requested (node_modules/@uniswap/v3-core/contracts/interfaces/pool/IUniswapV3PoolOwnerActions.sol#21)
Variable IUniswapV3PoolOwnerActions.collectProtocol(address,uint128,uint128).amount0Requested (node_modules/@uniswap/v3-core/contracts/interfaces/pool/IUniswapV3PoolOwnerActions.sol#20) is too similar to IUniswapV3PoolActions.collect(address,int24,int24,uint128,uint128).amount1Requested (node_modules/@uniswap/v3-core/contracts/interfaces/pool/IUniswapV3PoolActions.sol#48)
Variable IUniswapV3PoolState.positions(bytes32).feeGrowthInside0LastX128 (node_modules/@uniswap/v3-core/contracts/interfaces/pool/IUniswapV3PoolState.sol#94) is too similar to IUniswapV3PoolState.positions(bytes32).feeGrowthInside1LastX128 (node_modules/@uniswap/v3-core/contracts/interfaces/pool/IUniswapV3PoolState.sol#95)
Variable IUniswapV3PoolState.ticks(int24).feeGrowthOutside0X128 (node_modules/@uniswap/v3-core/contracts/interfaces/pool/IUniswapV3PoolState.sol#71) is too similar to IUniswapV3PoolState.ticks(int24).feeGrowthOutside1X128 (node_modules/@uniswap/v3-core/contracts/interfaces/pool/IUniswapV3PoolState.sol#72)
Variable IUniswapV3PoolOwnerActions.setFeeProtocol(uint8,uint8).feeProtocol0 (node_modules/@uniswap/v3-core/contracts/interfaces/pool/IUniswapV3PoolOwnerActions.sol#10) is too similar to IUniswapV3PoolOwnerActions.setFeeProtocol(uint8,uint8).feeProtocol1 (node_modules/@uniswap/v3-core/contracts/interfaces/pool/IUniswapV3PoolOwnerActions.sol#10)
Variable IUniswapV3PoolState.positions(bytes32).tokensOwed0 (node_modules/@uniswap/v3-core/contracts/interfaces/pool/IUniswapV3PoolState.sol#96) is too similar to IUniswapV3PoolState.positions(bytes32).tokensOwed1 (node_modules/@uniswap/v3-core/contracts/interfaces/pool/IUniswapV3PoolState.sol#97)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-are-too-similar

FixedPoint96.slitherConstructorConstantVariables() (node_modules/@uniswap/v3-core/contracts/libraries/FixedPoint96.sol#7-10) uses literals with too many digits:
        - Q96 = 0x1000000000000000000000000 (node_modules/@uniswap/v3-core/contracts/libraries/FixedPoint96.sol#9)
TickMath.getSqrtRatioAtTick(int24) (node_modules/@uniswap/v3-core/contracts/libraries/TickMath.sol#26-61) uses literals with too many digits:
        - ratio = 0x100000000000000000000000000000000 (node_modules/@uniswap/v3-core/contracts/libraries/TickMath.sol#31-33)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits

tokenPriceDai() should be declared external:
        - TWAPPriceGetter.tokenPriceDai() (contracts/v6.3/TWAPPriceGetter.sol#62-79)



Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

GNSTokenBridge.sol

```
Math.mulDiv(uint256,uint256,uint256) (../../this-arbitrum-v6.3-d6c7fe7f386b5678ee2597a17e866cee189f056e/node_modules/@openzeppelin/contracts/utils/math/Math.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (../../this-arbitrum-v6.3-d6c7fe7f386b5678ee2597a17e866cee189f056e/node_modules/@openzeppelin/contracts/utils/math/Math.sol#102)
        -inverse = (3 * denominator) ^ 2 (../../this-arbitrum-v6.3-d6c7fe7f386b5678ee2597a17e866cee189f056e/node_modules/@openzeppelin/contracts/utils/math/Math.sol#117)
Math.mulDiv(uint256,uint256,uint256) (../../this-arbitrum-v6.3-d6c7fe7f386b5678ee2597a17e866cee189f056e/node_modules/@openzeppelin/contracts/utils/math/Math.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (../../this-arbitrum-v6.3-d6c7fe7f386b5678ee2597a17e866cee189f056e/node_modules/@openzeppelin/contracts/utils/math/Math.sol#102)
        -inverse *= 2 - denominator * inverse (../../this-arbitrum-v6.3-d6c7fe7f386b5678ee2597a17e866cee189f056e/node_modules/@openzeppelin/contracts/utils/math/Math.sol#121)
Math.mulDiv(uint256,uint256,uint256) (../../this-arbitrum-v6.3-d6c7fe7f386b5678ee2597a17e866cee189f056e/node_modules/@openzeppelin/contracts/utils/math/Math.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (../../this-arbitrum-v6.3-d6c7fe7f386b5678ee2597a17e866cee189f056e/node_modules/@openzeppelin/contracts/utils/math/Math.sol#102)
        -inverse *= 2 - denominator * inverse (../../this-arbitrum-v6.3-d6c7fe7f386b5678ee2597a17e866cee189f056e/node_modules/@openzeppelin/contracts/utils/math/Math.sol#122)
Math.mulDiv(uint256,uint256,uint256) (../../this-arbitrum-v6.3-d6c7fe7f386b5678ee2597a17e866cee189f056e/node_modules/@openzeppelin/contracts/utils/math/Math.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (../../this-arbitrum-v6.3-d6c7fe7f386b5678ee2597a17e866cee189f056e/node_modules/@openzeppelin/contracts/utils/math/Math.sol#102)
        -inverse *= 2 - denominator * inverse (../../this-arbitrum-v6.3-d6c7fe7f386b5678ee2597a17e866cee189f056e/node_modules/@openzeppelin/contracts/utils/math/Math.sol#123)
Math.mulDiv(uint256,uint256,uint256) (../../this-arbitrum-v6.3-d6c7fe7f386b5678ee2597a17e866cee189f056e/node_modules/@openzeppelin/contracts/utils/math/Math.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (../../this-arbitrum-v6.3-d6c7fe7f386b5678ee2597a17e866cee189f056e/node_modules/@openzeppelin/contracts/utils/math/Math.sol#102)
        -inverse *= 2 - denominator * inverse (../../this-arbitrum-v6.3-d6c7fe7f386b5678ee2597a17e866cee189f056e/node_modules/@openzeppelin/contracts/utils/math/Math.sol#124)
Math.mulDiv(uint256,uint256,uint256) (../../this-arbitrum-v6.3-d6c7fe7f386b5678ee2597a17e866cee189f056e/node_modules/@openzeppelin/contracts/utils/math/Math.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (../../this-arbitrum-v6.3-d6c7fe7f386b5678ee2597a17e866cee189f056e/node_modules/@openzeppelin/contracts/utils/math/Math.sol#102)
        -inverse *= 2 - denominator * inverse (../../this-arbitrum-v6.3-d6c7fe7f386b5678ee2597a17e866cee189f056e/node_modules/@openzeppelin/contracts/utils/math/Math.sol#125)
Math.mulDiv(uint256,uint256,uint256) (../../this-arbitrum-v6.3-d6c7fe7f386b5678ee2597a17e866cee189f056e/node_modules/@openzeppelin/contracts/utils/math/Math.sol#55-135) performs a multiplication on the result of a division:
        -denominator = denominator / twos (../../this-arbitrum-v6.3-d6c7fe7f386b5678ee2597a17e866cee189f056e/node_modules/@openzeppelin/contracts/utils/math/Math.sol#102)
        -inverse *= 2 - denominator * inverse (../../this-arbitrum-v6.3-d6c7fe7f386b5678ee2597a17e866cee189f056e/node_modules/@openzeppelin/contracts/utils/math/Math.sol#126)
Math.mulDiv(uint256,uint256,uint256) (../../this-arbitrum-v6.3-d6c7fe7f386b5678ee2597a17e866cee189f056e/node_modules/@openzeppelin/contracts/utils/math/Math.sol#55-135) performs a multiplication on the result of a division:
        -prod0 = prod0 / twos (../../this-arbitrum-v6.3-d6c7fe7f386b5678ee2597a17e866cee189f056e/node_modules/@openzeppelin/contracts/utils/math/Math.sol#105)
        -result = prod0 * inverse (../../this-arbitrum-v6.3-d6c7fe7f386b5678ee2597a17e866cee189f056e/node_modules/@openzeppelin/contracts/utils/math/Math.sol#132)
BytesLib.concatStorage(bytes,bytes) (util/BytesLib.sol#91-226) performs a multiplication on the result of a division:
        -sstore(uint256,uint256)(_preBytes,fslot_concatStorage_asm_0 + mload(uint256)(_postBytes + 0x20) / 0x100 ** 32 - mlength_concatStorage_asm_0 * 0x100 ** 32 - newlength_concatStorage_asm_0 + mlength_concatStorage_asm_0 * 2) (util/BytesLib.sol#115-140)
BytesLib.concatStorage(bytes,bytes) (util/BytesLib.sol#91-226) performs a multiplication on the result of a division:
        -sstore(uint256,uint256)(sc_concatStorage_asm_0,mload(uint256)(mc_concatStorage_asm_0) / mask_concatStorage_asm_0 * mask_concatStorage_asm_0) (util/BytesLib.sol#189)
BytesLib.concatStorage(bytes,bytes) (util/BytesLib.sol#91-226) performs a multiplication on the result of a division:
        -sstore(uint256,uint256)(sc_concatStorage_asm_0,mload(uint256)(mc_concatStorage_asm_0) / mask_concatStorage_asm_0 * mask_concatStorage_asm_0) (util/BytesLib.sol#223)
BytesLib.equalStorage(bytes,bytes) (util/BytesLib.sol#439-509) performs a multiplication on the result of a division:
        -fslot_equalStorage_asm_0 = fslot_equalStorage_asm_0 / 0x100 * 0x100 (util/BytesLib.sol#466)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#divide-before-multiply

ERC20MintRateLimiter.executePendingClaim(address) (contracts/abstract/ERC20MintRateLimiter.sol#105-142) uses a dangerous strict equality:
        - remainingClaimAmount == 0 (contracts/abstract/ERC20MintRateLimiter.sol#125)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dangerous-strict-equalities

GNSTokenBridge._buildMintMessage(address,uint256).message (contracts/GNSTokenBridge.sol#69) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

GNSTokenBridge.constructor(address,address,uint256,uint256).token (contracts/GNSTokenBridge.sol#42) shadows:
        - ERC20MintRateLimiter.token (contracts/abstract/ERC20MintRateLimiter.sol#52) (state variable)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

LzApp.setPrecrime(address)._precrime (contracts/LzApp.sol#105) lacks a zero-check on :
        - precrime = _precrime (contracts/LzApp.sol#106)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation

Variable 'BytesLib.concatStorage(bytes,bytes).sc_concatStorage_asm_0 (util/BytesLib.sol#147)' in BytesLib.concatStorage(bytes,bytes) (util/BytesLib.sol#91-226) potentially used before declaration: sc_concatStorage_asm_0 = keccak256(uint256,uint256)(0x0,0x20) + slength_concatStorage_asm_0 / 32 (util/BytesLib.sol#195)
Variable 'BytesLib.concatStorage(bytes,bytes).submod_concatStorage_asm_0 (util/BytesLib.sol#161)' in BytesLib.concatStorage(bytes,bytes) (util/BytesLib.sol#91-226) potentially used before declaration: submod_concatStorage_asm_0 = 32 - slengthmod_concatStorage_asm_0 (util/BytesLib.sol#204)
Variable 'BytesLib.concatStorage(bytes,bytes).mc_concatStorage_asm_0 (util/BytesLib.sol#162)' in BytesLib.concatStorage(bytes,bytes) (util/BytesLib.sol#91-226) potentially used before declaration: mc_concatStorage_asm_0 = _postBytes + submod_concatStorage_asm_0 (util/BytesLib.sol#205)
Variable 'BytesLib.concatStorage(bytes,bytes).submod_concatStorage_asm_0 (util/BytesLib.sol#161)' in BytesLib.concatStorage(bytes,bytes) (util/BytesLib.sol#91-226) potentially used before declaration: mc_concatStorage_asm_0 = _postBytes + submod_concatStorage_asm_0 (util/BytesLib.sol#205)
Variable 'BytesLib.concatStorage(bytes,bytes).end_concatStorage_asm_0 (util/BytesLib.sol#163)' in BytesLib.concatStorage(bytes,bytes) (util/BytesLib.sol#91-226) potentially used before declaration: end_concatStorage_asm_0 = _postBytes + mlength_concatStorage_asm_0 (util/BytesLib.sol#206)
Variable 'BytesLib.concatStorage(bytes,bytes).mask_concatStorage_asm_0 (util/BytesLib.sol#164)' in BytesLib.concatStorage(bytes,bytes) (util/BytesLib.sol#91-226) potentially used before declaration: mask_concatStorage_asm_0 = 0x100 ** submod_concatStorage_asm_0 - 1 (util/BytesLib.sol#207)
Variable 'BytesLib.concatStorage(bytes,bytes).submod_concatStorage_asm_0 (util/BytesLib.sol#161)' in BytesLib.concatStorage(bytes,bytes) (util/BytesLib.sol#91-226) potentially used before declaration: mask_concatStorage_asm_0 = 0x100 ** submod_concatStorage_asm_0 - 1 (util/BytesLib.sol#207)
Variable 'BytesLib.concatStorage(bytes,bytes).mc_concatStorage_asm_0 (util/BytesLib.sol#162)' in BytesLib.concatStorage(bytes,bytes) (util/BytesLib.sol#91-226) potentially used before declaration: sstore(uint256,uint256)(sc_concatStorage_asm_0,sload(uint256)(sc_concatStorage_asm_0) + mload(uint256)(mc_concatStorage_asm_0) & mask_concatStorage_asm_0) (util/BytesLib.sol#209)
Variable 'BytesLib.concatStorage(bytes,bytes).mask_concatStorage_asm_0 (util/BytesLib.sol#164)' in BytesLib.concatStorage(bytes,bytes) (util/BytesLib.sol#91-226) potentially used before declaration: sstore(uint256,uint256)(sc_concatStorage_asm_0,sload(uint256)(sc_concatStorage_asm_0) + mload(uint256)(mc_concatStorage_asm_0) & mask_concatStorage_asm_0) (util/BytesLib.sol#209)
Variable 'BytesLib.concatStorage(bytes,bytes).mc_concatStorage_asm_0 (util/BytesLib.sol#147)' in BytesLib.concatStorage(bytes,bytes) (util/BytesLib.sol#91-226) potentially used before declaration: sstore(uint256,uint256)(sc_concatStorage_asm_0,sload(uint256)(sc_concatStorage_asm_0) + mload(uint256)(mc_concatStorage_asm_0) & mask_concatStorage_asm_0) (util/BytesLib.sol#209)
Variable 'BytesLib.concatStorage(bytes,bytes).sc_concatStorage_asm_0 (util/BytesLib.sol#147)' in BytesLib.concatStorage(bytes,bytes) (util/BytesLib.sol#91-226) potentially used before declaration: sc_concatStorage_asm_0 = sc_concatStorage_asm_0 + 1 (util/BytesLib.sol#212)
Variable 'BytesLib.concatStorage(bytes,bytes).mc_concatStorage_asm_0 (util/BytesLib.sol#162)' in BytesLib.concatStorage(bytes,bytes) (util/BytesLib.sol#91-226) potentially used before declaration: mc_concatStorage_asm_0 = mc_concatStorage_asm_0 + 0x20 (util/BytesLib.sol#213)
Variable 'BytesLib.concatStorage(bytes,bytes).end_concatStorage_asm_0 (util/BytesLib.sol#163)' in BytesLib.concatStorage(bytes,bytes) (util/BytesLib.sol#91-226) potentially used before declaration: mc_concatStorage_asm_0 < end_concatStorage_asm_0 (util/BytesLib.sol#214)
Variable 'BytesLib.concatStorage(bytes,bytes).mc_concatStorage_asm_0 (util/BytesLib.sol#162)' in BytesLib.concatStorage(bytes,bytes) (util/BytesLib.sol#91-226) potentially used before declaration: mc_concatStorage_asm_0 < end_concatStorage_asm_0 (util/BytesLib.sol#214)
Variable 'BytesLib.concatStorage(bytes,bytes).end_concatStorage_asm_0 (util/BytesLib.sol#163)' in BytesLib.concatStorage(bytes,bytes) (util/BytesLib.sol#91-226) potentially used before declaration: mask_concatStorage_asm_0 = 0x100 ** mc_concatStorage_asm_0 - end_concatStorage_asm_0 (util/BytesLib.sol#221)
Variable 'BytesLib.concatStorage(bytes,bytes).mc_concatStorage_asm_0 (util/BytesLib.sol#162)' in BytesLib.concatStorage(bytes,bytes) (util/BytesLib.sol#91-226) potentially used before declaration: mask_concatStorage_asm_0 = 0x100 ** mc_concatStorage_asm_0 - end_concatStorage_asm_0 (util/BytesLib.sol#221)
Variable 'BytesLib.concatStorage(bytes,bytes).mask_concatStorage_asm_0 (util/BytesLib.sol#164)' in BytesLib.concatStorage(bytes,bytes) (util/BytesLib.sol#91-226) potentially used before declaration: mask_concatStorage_asm_0 = 0x100 ** mc_concatStorage_asm_0 - end_concatStorage_asm_0 (util/BytesLib.sol#221)
Variable 'BytesLib.concatStorage(bytes,bytes).mc_concatStorage_asm_0 (util/BytesLib.sol#162)' in BytesLib.concatStorage(bytes,bytes) (util/BytesLib.sol#91-226) potentially used before declaration: sstore(uint256,uint256)(sc_concatStorage_asm_0,sload(uint256)(mc_concatStorage_asm_0) / mask_concatStorage_asm_0 * mask_concatStorage_asm_0) (util/BytesLib.sol#223)
Variable 'BytesLib.concatStorage(bytes,bytes).mask_concatStorage_asm_0 (util/BytesLib.sol#164)' in BytesLib.concatStorage(bytes,bytes) (util/BytesLib.sol#91-226) potentially used before declaration: sstore(uint256,uint256)(sc_concatStorage_asm_0,sload(uint256)(mc_concatStorage_asm_0) / mask_concatStorage_asm_0 * mask_concatStorage_asm_0) (util/BytesLib.sol#223)
Variable 'BytesLib.concatStorage(bytes,bytes).sc_concatStorage_asm_0 (util/BytesLib.sol#147)' in BytesLib.concatStorage(bytes,bytes) (util/BytesLib.sol#91-226) potentially used before declaration: sstore(uint256,uint256)(sc_concatStorage_asm_0,sload(uint256)(mc_concatStorage_asm_0) / mask_concatStorage_asm_0 * mask_concatStorage_asm_0) (util/BytesLib.sol#223)
Variable 'BytesLib.concatStorage(bytes,bytes).mc_concatStorage_asm_0 (util/BytesLib.sol#162)' in BytesLib.concatStorage(bytes,bytes) (util/BytesLib.sol#91-226) potentially used before declaration: sstore(uint256,uint256)(sc_concatStorage_asm_0,sload(uint256)(mc_concatStorage_asm_0)) (util/BytesLib.sol#218)
Variable 'BytesLib.concatStorage(bytes,bytes).sc_concatStorage_asm_0 (util/BytesLib.sol#147)' in BytesLib.concatStorage(bytes,bytes) (util/BytesLib.sol#91-226) potentially used before declaration: sstore(uint256,uint256)(sc_concatStorage_asm_0,sload(uint256)(mc_concatStorage_asm_0)) (util/BytesLib.sol#218)
Variable 'BytesLib.concatStorage(bytes,bytes).sc_concatStorage_asm_0 (util/BytesLib.sol#147)' in BytesLib.concatStorage(bytes,bytes) (util/BytesLib.sol#91-226) potentially used before declaration: sc_concatStorage_asm_0 = sc_concatStorage_asm_0 + 1 (util/BytesLib.sol#215)
Variable 'BytesLib.concatStorage(bytes,bytes).mc_concatStorage_asm_0 (util/BytesLib.sol#162)' in BytesLib.concatStorage(bytes,bytes) (util/BytesLib.sol#91-226) potentially used before declaration: mc_concatStorage_asm_0 = mc_concatStorage_asm_0 + 0x20 (util/BytesLib.sol#216)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables
```

```
Reentrancy in GNSTokenBridge._nonblockingLzReceive(uint16,bytes,uint64,bytes) (contracts/GNSTokenBridge.sol#131-141):
        External calls:
        - tryMint(message.recipient,message.amount) (contracts/GNSTokenBridge.sol#139)
            - token.mint(to,amount) (contracts/abstract/ERC20MintRateLimiter.sol#96)
        Event emitted after the call(s):
        - MintReceived(message.recipient,message.amount) (contracts/GNSTokenBridge.sol#140)
Reentrancy in GNSTokenBridge.burnToken(uint16,uint256) (contracts/GNSTokenBridge.sol#99-120):
        External calls:
        - tryBurn(_msgSender(),amount) (contracts/GNSTokenBridge.sol#106)
            - token.burn(from,amount) (contracts/abstract/ERC20MintRateLimiter.sol#70)
        - _lzSend(dstChainId,payloadBytes,address(_msgSender()),address(0x0),bytes(),msg.value) (contracts/GNSTokenBridge.sol#110-117)
            - lzEndpoint.send{value: _nativeFee}(_dstChainId,trustedRemote,_payload,_refundAddress,_zroPaymentAddress,_adapterParams) (contracts/LzApp.sol#48)
        External calls sending eth:
        - _lzSend(dstChainId,payloadBytes,address(_msgSender()),address(0x0),bytes(),msg.value) (contracts/GNSTokenBridge.sol#110-117)
            - lzEndpoint.send{value: _nativeFee}(_dstChainId,trustedRemote,_payload,_refundAddress,_zroPaymentAddress,_adapterParams) (contracts/LzApp.sol#48)
        Event emitted after the call(s):
        - BurnReceived(dstChainId,_msgSender(),amount) (contracts/GNSTokenBridge.sol#119)
Reentrancy in ERC20MintRateLimiter.executePendingClaim(address) (contracts/abstract/ERC20MintRateLimiter.sol#105-142):
        External calls:
        - token.mint(receiver,claimAmount) (contracts/abstract/ERC20MintRateLimiter.sol#139)
        Event emitted after the call(s):
        - Claimed(receiver,claimAmount) (contracts/abstract/ERC20MintRateLimiter.sol#141)
Reentrancy in ERC20MintRateLimiter.tryBurn(address,uint256) (contracts/abstract/ERC20MintRateLimiter.sol#64-72):
        External calls:
        - token.burn(from,amount) (contracts/abstract/ERC20MintRateLimiter.sol#70)
        Event emitted after the call(s):
        - Burn(from,amount) (contracts/abstract/ERC20MintRateLimiter.sol#71)
Reentrancy in ERC20MintRateLimiter.tryMint(address,uint256) (contracts/abstract/ERC20MintRateLimiter.sol#77-100):
        External calls:
        - token.mint(to,amount) (contracts/abstract/ERC20MintRateLimiter.sol#96)
        Event emitted after the call(s):
        - Mint(to,amount) (contracts/abstract/ERC20MintRateLimiter.sol#98)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

ERC20MintRateLimiter.tryBurn(address,uint256) (contracts/abstract/ERC20MintRateLimiter.sol#64-72) uses timestamp for comparisons
        Dangerous comparisons:
        - countAfterDeduction > 0 (contracts/abstract/ERC20MintRateLimiter.sol#68)
ERC20MintRateLimiter.executePendingClaim(address) (contracts/abstract/ERC20MintRateLimiter.sol#105-142) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(pendingClaim.amount > 0,NO_PENDING_CLAIM) (contracts/abstract/ERC20MintRateLimiter.sol#107)
        - require(bool,string)(pendingClaim.claimTimestamp <= block.timestamp,CLAIM_TIME_NOT_REACHED) (contracts/abstract/ERC20MintRateLimiter.sol#108-111)
        - remainingClaimAmount == 0 (contracts/abstract/ERC20MintRateLimiter.sol#125)
        - epochLimit >= currentEpochCount (contracts/abstract/ERC20MintRateLimiter.sol#115-120)
EpochBasedLimiter.tryUpdateEpoch() (contracts/abstract/EpochBasedLimiter.sol#79-93) uses timestamp for comparisons
        Dangerous comparisons:
        - currentEpoch == 0 || block.timestamp >= currentEpochEnd (contracts/abstract/EpochBasedLimiter.sol#81)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

Math.mulDiv(uint256,uint256,uint256) (../../this-arbitrum-v6.3-d6c7fe7f386b5678ee2597a17e866cee189f056e/node_modules/@openzeppelin/contracts/utils/math/Math.sol#55-135) uses assembly
        - INLINE ASM (../../this-arbitrum-v6.3-d6c7fe7f386b5678ee2597a17e866cee189f056e/node_modules/@openzeppelin/contracts/utils/math/Math.sol#66-70)
        - INLINE ASM (../../this-arbitrum-v6.3-d6c7fe7f386b5678ee2597a17e866cee189f056e/node_modules/@openzeppelin/contracts/utils/math/Math.sol#86-93)
        - INLINE ASM (../../this-arbitrum-v6.3-d6c7fe7f386b5678ee2597a17e866cee189f056e/node_modules/@openzeppelin/contracts/utils/math/Math.sol#100-109)
LzApp._getGasLimit(bytes) (contracts/LzApp.sol#55-63) uses assembly
        - INLINE ASM (contracts/LzApp.sol#60-62)
BytesLib.concat(bytes,bytes) (util/BytesLib.sol#13-89) uses assembly
        - INLINE ASM (util/BytesLib.sol#23-86)
BytesLib.concatStorage(bytes,bytes) (util/BytesLib.sol#91-226) uses assembly
        - INLINE ASM (util/BytesLib.sol#92-225)
BytesLib.slice(bytes,uint256,uint256) (util/BytesLib.sol#228-295) uses assembly
        - INLINE ASM (util/BytesLib.sol#242-292)
BytesLib.toAddress(bytes,uint256) (util/BytesLib.sol#297-306) uses assembly
        - INLINE ASM (util/BytesLib.sol#301-303)
BytesLib.toUint8(bytes,uint256) (util/BytesLib.sol#308-317) uses assembly

        - INLINE ASM (util/BytesLib.sol#312-314)
BytesLib.toUint16(bytes,uint256) (util/BytesLib.sol#319-328) uses assembly
        - INLINE ASM (util/BytesLib.sol#323-325)
BytesLib.toUint32(bytes,uint256) (util/BytesLib.sol#330-339) uses assembly
        - INLINE ASM (util/BytesLib.sol#334-336)
BytesLib.toUint64(bytes,uint256) (util/BytesLib.sol#341-350) uses assembly
        - INLINE ASM (util/BytesLib.sol#345-347)
BytesLib.toUint96(bytes,uint256) (util/BytesLib.sol#352-361) uses assembly
        - INLINE ASM (util/BytesLib.sol#356-358)
BytesLib.toUint128(bytes,uint256) (util/BytesLib.sol#363-372) uses assembly
        - INLINE ASM (util/BytesLib.sol#367-369)
BytesLib.toUint256(bytes,uint256) (util/BytesLib.sol#374-383) uses assembly
        - INLINE ASM (util/BytesLib.sol#377-380)
BytesLib.toBytes32(bytes,uint256) (util/BytesLib.sol#385-394) uses assembly
        - INLINE ASM (util/BytesLib.sol#389-391)
BytesLib.equal(bytes,bytes) (util/BytesLib.sol#396-437) uses assembly
        - INLINE ASM (util/BytesLib.sol#399-434)
BytesLib.equalStorage(bytes,bytes) (util/BytesLib.sol#439-509) uses assembly
        - INLINE ASM (util/BytesLib.sol#449-506)
ExcessivelySafeCall.excessivelySafeCall(address,uint256,uint16,bytes) (util/ExcessivelySafeCall.sol#23-58) uses assembly
        - INLINE ASM (util/ExcessivelySafeCall.sol#37-56)
ExcessivelySafeCall.excessivelySafeStaticCall(address,uint256,uint16,bytes) (util/ExcessivelySafeCall.sol#75-109) uses assembly
        - INLINE ASM (util/ExcessivelySafeCall.sol#85-107)
ExcessivelySafeCall.swapSelector(bytes4,bytes) (util/ExcessivelySafeCall.sol#120-135) uses assembly
        - INLINE ASM (util/ExcessivelySafeCall.sol#126-134)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Different versions of Solidity is used:
        - Version used: ['0.8.17', '>=0.5.0', '>=0.7.6', '>=0.8.0<0.9.0', '^0.8.0']
        - ^0.8.0 (../../this-arbitrum-v6.3-d6c7fe7f386b5678ee2597a17e866cee189f056e/node_modules/@openzeppelin/contracts/access/Ownable.sol#4)
        - ^0.8.0 (../../this-arbitrum-v6.3-d6c7fe7f386b5678ee2597a17e866cee189f056e/node_modules/@openzeppelin/contracts/security/Pausable.sol#4)
        - ^0.8.0 (../../this-arbitrum-v6.3-d6c7fe7f386b5678ee2597a17e866cee189f056e/node_modules/@openzeppelin/contracts/utils/Context.sol#4)
        - ^0.8.0 (../../this-arbitrum-v6.3-d6c7fe7f386b5678ee2597a17e866cee189f056e/node_modules/@openzeppelin/contracts/utils/math/Math.sol#4)
        - 0.8.17 (contracts/GNSTokenBridge.sol#2)
        - ^0.8.0 (contracts/LzApp.sol#3)
        - ^0.8.0 (contracts/NonblockingLzApp.sol#3)
        - 0.8.17 (contracts/abstract/ERC20MintRateLimiter.sol#2)
        - 0.8.17 (contracts/abstract/EpochBasedLimiter.sol#2)
        - 0.8.17 (contracts/interface/IBlacklist.sol#2)
        - 0.8.17 (contracts/interface/IERC20MintRateLimiter.sol#2)
        - 0.8.17 (contracts/interface/IEpochBasedLimiter.sol#2)
        - >=0.5.0 (interfaces/ILayerZeroEndpoint.sol#3)
        - >=0.5.0 (interfaces/ILayerZeroReceiver.sol#3)
        - >=0.5.0 (interfaces/ILayerZeroUserApplicationConfig.sol#3)
        - >=0.8.0<0.9.0 (util/BytesLib.sol#5)
        - >=0.7.6 (util/ExcessivelySafeCall.sol#2)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

BytesLib.concat(bytes,bytes) (util/BytesLib.sol#13-89) is never used and should be removed
BytesLib.concatStorage(bytes,bytes) (util/BytesLib.sol#91-226) is never used and should be removed
BytesLib.equal(bytes,bytes) (util/BytesLib.sol#396-437) is never used and should be removed
BytesLib.equalStorage(bytes,bytes) (util/BytesLib.sol#439-509) is never used and should be removed
BytesLib.toAddress(bytes,uint256) (util/BytesLib.sol#297-306) is never used and should be removed
BytesLib.toBytes32(bytes,uint256) (util/BytesLib.sol#385-394) is never used and should be removed
BytesLib.toUint128(bytes,uint256) (util/BytesLib.sol#363-372) is never used and should be removed
BytesLib.toUint16(bytes,uint256) (util/BytesLib.sol#319-328) is never used and should be removed
BytesLib.toUint256(bytes,uint256) (util/BytesLib.sol#374-383) is never used and should be removed
BytesLib.toUint32(bytes,uint256) (util/BytesLib.sol#330-339) is never used and should be removed
BytesLib.toUint64(bytes,uint256) (util/BytesLib.sol#341-350) is never used and should be removed
BytesLib.toUint8(bytes,uint256) (util/BytesLib.sol#308-317) is never used and should be removed
BytesLib.toUint96(bytes,uint256) (util/BytesLib.sol#352-361) is never used and should be removed
Context._msgData() (../../this-arbitrum-v6.3-d6c7fe7f386b5678ee2597a17e866cee189f056e/node_modules/@openzeppelin/contracts/utils/Context.sol#21-23) is never used and should be removed
ExcessivelySafeCall.excessivelySafeStaticCall(address,uint256,uint16,bytes) (util/ExcessivelySafeCall.sol#75-109) is never used and should be removed
ExcessivelySafeCall.swapSelector(bytes4,bytes) (util/ExcessivelySafeCall.sol#120-135) is never used and should be removed
LzApp._blockingLzReceive(uint16,bytes,uint64,bytes) (contracts/LzApp.sol#83) is never used and should be removed
LzApp._checkGasLimit(uint16,uint16,bytes,uint256) (contracts/LzApp.sol#51-56) is never used and should be removed
LzApp._getGasLimit(bytes) (contracts/LzApp.sol#58-63) is never used and should be removed
Math.average(uint256,uint256) (../../this-arbitrum-v6.3-d6c7fe7f386b5678ee2597a17e866cee189f056e/node_modules/@openzeppelin/contracts/utils/math/Math.sol#34-37) is never used and should be removed
Math.ceilDiv(uint256,uint256) (../../this-arbitrum-v6.3-d6c7fe7f386b5678ee2597a17e866cee189f056e/node_modules/@openzeppelin/contracts/utils/math/Math.sol#45-48) is never used and should be removed
Math.max(uint256,uint256) (../../this-arbitrum-v6.3-d6c7fe7f386b5678ee2597a17e866cee189f056e/node_modules/@openzeppelin/contracts/utils/math/Math.sol#19-21) is never used and should be removed
Math.mulDiv(uint256,uint256,uint256) (../../this-arbitrum-v6.3-d6c7fe7f386b5678ee2597a17e866cee189f056e/node_modules/@openzeppelin/contracts/utils/math/Math.sol#55-135) is never used and should be removed
Math.mulDiv(uint256,uint256,uint256,Math.Rounding) (../../this-arbitrum-v6.3-d6c7fe7f386b5678ee2597a17e866cee189f056e/node_modules/@openzeppelin/contracts/utils/math/Math.sol#140-151) is never used and should be removed
Math.sqrt(uint256) (../../this-arbitrum-v6.3-d6c7fe7f386b5678ee2597a17e866cee189f056e/node_modules/@openzeppelin/contracts/utils/math/Math.sol#158-214) is never used and should be removed
Math.sqrt(uint256,Math.Rounding) (../../this-arbitrum-v6.3-d6c7fe7f386b5678ee2597a17e866cee189f056e/node_modules/@openzeppelin/contracts/utils/math/Math.sol#219-225) is never used and should be removed
NonblockingLzApp._nonblockingLzReceive(uint16,bytes,uint64,bytes) (contracts/NonblockingLzApp.sol#40) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version^0.8.0 (../../this-arbitrum-v6.3-d6c7fe7f386b5678ee2597a17e866cee189f056e/node_modules/@openzeppelin/contracts/access/Ownable.sol#4) allows old versions
Pragma version^0.8.0 (../../this-arbitrum-v6.3-d6c7fe7f386b5678ee2597a17e866cee189f056e/node_modules/@openzeppelin/contracts/security/Pausable.sol#4) allows old versions
Pragma version^0.8.0 (../../this-arbitrum-v6.3-d6c7fe7f386b5678ee2597a17e866cee189f056e/node_modules/@openzeppelin/contracts/utils/Context.sol#4) allows old versions
Pragma version^0.8.0 (../../this-arbitrum-v6.3-d6c7fe7f386b5678ee2597a17e866cee189f056e/node_modules/@openzeppelin/contracts/utils/math/Math.sol#4) allows old versions
Pragma version0.8.17 (contracts/GNSTokenBridge.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version^0.8.0 (contracts/LzApp.sol#3) allows old versions
Pragma version^0.8.0 (contracts/NonblockingLzApp.sol#3) allows old versions
Pragma version0.8.17 (contracts/abstract/ERC20MintRateLimiter.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.17 (contracts/abstract/EpochBasedLimiter.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.17 (contracts/interface/IBlacklist.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.17 (contracts/interface/IERC20MintRateLimiter.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.17 (contracts/interface/IEpochBasedLimiter.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version>=0.5.0 (interfaces/ILayerZeroEndpoint.sol#3) allows old versions
Pragma version>=0.5.0 (interfaces/ILayerZeroReceiver.sol#3) allows old versions
Pragma version>=0.5.0 (interfaces/ILayerZeroUserApplicationConfig.sol#3) allows old versions
Pragma version>=0.8.0<0.9.0 (util/BytesLib.sol#5) is too complex
solc-0.8.17 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Parameter GNSTokenBridge.estimateNativeFee(uint16,address,uint256)._dstChainId (contracts/GNSTokenBridge.sol#80) is not in mixedCase
Parameter GNSTokenBridge.estimateNativeFee(uint16,address,uint256)._amount (contracts/GNSTokenBridge.sol#82) is not in mixedCase
Parameter LzApp.lzReceive(uint16,bytes,uint64,bytes)._srcChainId (contracts/LzApp.sol#31) is not in mixedCase
Parameter LzApp.lzReceive(uint16,bytes,uint64,bytes)._srcAddress (contracts/LzApp.sol#31) is not in mixedCase
Parameter LzApp.lzReceive(uint16,bytes,uint64,bytes)._nonce (contracts/LzApp.sol#31) is not in mixedCase
Parameter LzApp.lzReceive(uint16,bytes,uint64,bytes)._payload (contracts/LzApp.sol#31) is not in mixedCase
Parameter LzApp.getConfig(uint16,uint16,address,uint256)._version (contracts/LzApp.sol#66) is not in mixedCase
Parameter LzApp.getConfig(uint16,uint16,address,uint256)._chainId (contracts/LzApp.sol#66) is not in mixedCase
Parameter LzApp.getConfig(uint16,uint16,address,uint256)._configType (contracts/LzApp.sol#66) is not in mixedCase
Parameter LzApp.setConfig(uint16,uint16,uint256,bytes)._version (contracts/LzApp.sol#71) is not in mixedCase
Parameter LzApp.setConfig(uint16,uint16,uint256,bytes)._chainId (contracts/LzApp.sol#71) is not in mixedCase
Parameter LzApp.setConfig(uint16,uint16,uint256,bytes)._configType (contracts/LzApp.sol#71) is not in mixedCase
Parameter LzApp.setConfig(uint16,uint16,uint256,bytes)._config (contracts/LzApp.sol#71) is not in mixedCase
Parameter LzApp.setSendVersion(uint16)._version (contracts/LzApp.sol#75) is not in mixedCase
Parameter LzApp.setReceiveVersion(uint16)._version (contracts/LzApp.sol#79) is not in mixedCase
Parameter LzApp.forceResumeReceive(uint16,bytes)._srcChainId (contracts/LzApp.sol#83) is not in mixedCase
Parameter LzApp.forceResumeReceive(uint16,bytes)._srcAddress (contracts/LzApp.sol#83) is not in mixedCase
Parameter LzApp.setTrustedRemote(uint16,bytes)._srcChainId (contracts/LzApp.sol#89) is not in mixedCase
Parameter LzApp.setTrustedRemote(uint16,bytes)._path (contracts/LzApp.sol#89) is not in mixedCase
```

```
Parameter LzApp.setTrustedRemoteAddress(uint16,bytes)._remoteChainId (contracts/LzApp.sol#94) is not in mixedCase
Parameter LzApp.setTrustedRemoteAddress(uint16,bytes)._remoteAddress (contracts/LzApp.sol#94) is not in mixedCase
Parameter LzApp.getTrustedRemoteAddress(uint16)._remoteChainId (contracts/LzApp.sol#99) is not in mixedCase
Parameter LzApp.setPrecrime(address)._precrime (contracts/LzApp.sol#105) is not in mixedCase
Parameter LzApp.setMinDstGas(uint16,uint16,uint256)._dstChainId (contracts/LzApp.sol#110) is not in mixedCase
Parameter LzApp.setMinDstGas(uint16,uint16,uint256)._packetType (contracts/LzApp.sol#110) is not in mixedCase
Parameter LzApp.setMinDstGas(uint16,uint16,uint256)._minGas (contracts/LzApp.sol#110) is not in mixedCase
Parameter LzApp.isTrustedRemote(uint16,bytes)._srcChainId (contracts/LzApp.sol#117) is not in mixedCase
Parameter LzApp.isTrustedRemote(uint16,bytes)._srcAddress (contracts/LzApp.sol#117) is not in mixedCase
Parameter NonblockingLzApp.nonblockingLzReceive(uint16,bytes,uint64,bytes)._srcChainId (contracts/NonblockingLzApp.sol#33) is not in mixedCase
Parameter NonblockingLzApp.nonblockingLzReceive(uint16,bytes,uint64,bytes)._srcAddress (contracts/NonblockingLzApp.sol#33) is not in mixedCase
Parameter NonblockingLzApp.nonblockingLzReceive(uint16,bytes,uint64,bytes)._nonce (contracts/NonblockingLzApp.sol#33) is not in mixedCase
Parameter NonblockingLzApp.nonblockingLzReceive(uint16,bytes,uint64,bytes)._payload (contracts/NonblockingLzApp.sol#33) is not in mixedCase
Parameter NonblockingLzApp.retryMessage(uint16,bytes,uint64,bytes)._srcChainId (contracts/NonblockingLzApp.sol#42) is not in mixedCase
Parameter NonblockingLzApp.retryMessage(uint16,bytes,uint64,bytes)._srcAddress (contracts/NonblockingLzApp.sol#42) is not in mixedCase
Parameter NonblockingLzApp.retryMessage(uint16,bytes,uint64,bytes)._nonce (contracts/NonblockingLzApp.sol#42) is not in mixedCase
Parameter NonblockingLzApp.retryMessage(uint16,bytes,uint64,bytes)._payload (contracts/NonblockingLzApp.sol#42) is not in mixedCase
Parameter ERC20MintRateLimiter.blacklistReceiver(address,bool)._blacklist (contracts/abstract/ERC20MintRateLimiter.sol#147) is not in mixedCase
Parameter EpochBasedLimiter.updateEpochLimit(uint256)._epochLimit (contracts/abstract/EpochBasedLimiter.sol#52) is not in mixedCase
Parameter EpochBasedLimiter.updateEpochDuration(uint256)._epochDuration (contracts/abstract/EpochBasedLimiter.sol#65) is not in mixedCase
Variable EpochBasedLimiter.MIN_EPOCH_LIMIT (contracts/abstract/EpochBasedLimiter.sol#13) is not in mixedCase
Variable EpochBasedLimiter.MAX_EPOCH_LIMIT (contracts/abstract/EpochBasedLimiter.sol#14) is not in mixedCase
Parameter BytesLib.concat(bytes,bytes)._preBytes (util/BytesLib.sol#14) is not in mixedCase
Parameter BytesLib.concat(bytes,bytes)._postBytes (util/BytesLib.sol#15) is not in mixedCase
Parameter BytesLib.concatStorage(bytes,bytes)._preBytes (util/BytesLib.sol#91) is not in mixedCase
Parameter BytesLib.concatStorage(bytes,bytes)._postBytes (util/BytesLib.sol#91) is not in mixedCase
Parameter BytesLib.slice(bytes,uint256,uint256)._bytes (util/BytesLib.sol#229) is not in mixedCase
Parameter BytesLib.slice(bytes,uint256,uint256)._start (util/BytesLib.sol#230) is not in mixedCase
Parameter BytesLib.slice(bytes,uint256,uint256)._length (util/BytesLib.sol#231) is not in mixedCase
Parameter BytesLib.toAddress(bytes,uint256)._bytes (util/BytesLib.sol#297) is not in mixedCase
Parameter BytesLib.toAddress(bytes,uint256)._start (util/BytesLib.sol#297) is not in mixedCase
Parameter BytesLib.toUint8(bytes,uint256)._bytes (util/BytesLib.sol#308) is not in mixedCase
Parameter BytesLib.toUint8(bytes,uint256)._start (util/BytesLib.sol#308) is not in mixedCase
Parameter BytesLib.toUint16(bytes,uint256)._bytes (util/BytesLib.sol#319) is not in mixedCase
Parameter BytesLib.toUint16(bytes,uint256)._start (util/BytesLib.sol#319) is not in mixedCase
Parameter BytesLib.toUint32(bytes,uint256)._bytes (util/BytesLib.sol#330) is not in mixedCase
Parameter BytesLib.toUint32(bytes,uint256)._start (util/BytesLib.sol#330) is not in mixedCase
Parameter BytesLib.toUint64(bytes,uint256)._bytes (util/BytesLib.sol#341) is not in mixedCase
Parameter BytesLib.toUint64(bytes,uint256)._start (util/BytesLib.sol#341) is not in mixedCase
Parameter BytesLib.toUint96(bytes,uint256)._bytes (util/BytesLib.sol#352) is not in mixedCase
Parameter BytesLib.toUint96(bytes,uint256)._start (util/BytesLib.sol#352) is not in mixedCase
Parameter BytesLib.toUint128(bytes,uint256)._bytes (util/BytesLib.sol#363) is not in mixedCase
Parameter BytesLib.toUint128(bytes,uint256)._start (util/BytesLib.sol#363) is not in mixedCase
Parameter BytesLib.toUint256(bytes,uint256)._bytes (util/BytesLib.sol#374) is not in mixedCase
Parameter BytesLib.toUint256(bytes,uint256)._start (util/BytesLib.sol#374) is not in mixedCase
Parameter BytesLib.toBytes32(bytes,uint256)._bytes (util/BytesLib.sol#385) is not in mixedCase
Parameter BytesLib.toBytes32(bytes,uint256)._start (util/BytesLib.sol#385) is not in mixedCase
Parameter BytesLib.equal(bytes,bytes)._preBytes (util/BytesLib.sol#396) is not in mixedCase
Parameter BytesLib.equal(bytes,bytes)._postBytes (util/BytesLib.sol#396) is not in mixedCase
Parameter BytesLib.equalStorage(bytes,bytes)._preBytes (util/BytesLib.sol#440) is not in mixedCase
Parameter BytesLib.equalStorage(bytes,bytes)._postBytes (util/BytesLib.sol#441) is not in mixedCase
Parameter ExcessivelySafeCall.excessivelySafeCall(address,uint256,uint16,bytes)._target (util/ExcessivelySafeCall.sol#24) is not in mixedCase
Parameter ExcessivelySafeCall.excessivelySafeCall(address,uint256,uint16,bytes)._gas (util/ExcessivelySafeCall.sol#25) is not in mixedCase
Parameter ExcessivelySafeCall.excessivelySafeCall(address,uint256,uint16,bytes)._maxCopy (util/ExcessivelySafeCall.sol#26) is not in mixedCase
Parameter ExcessivelySafeCall.excessivelySafeCall(address,uint256,uint16,bytes)._calldata (util/ExcessivelySafeCall.sol#27) is not in mixedCase
Parameter ExcessivelySafeCall.excessivelySafeStaticCall(address,uint256,uint16,bytes)._target (util/ExcessivelySafeCall.sol#76) is not in mixedCase
Parameter ExcessivelySafeCall.excessivelySafeStaticCall(address,uint256,uint16,bytes)._gas (util/ExcessivelySafeCall.sol#77) is not in mixedCase
Parameter ExcessivelySafeCall.excessivelySafeStaticCall(address,uint256,uint16,bytes)._maxCopy (util/ExcessivelySafeCall.sol#78) is not in mixedCase
Parameter ExcessivelySafeCall.excessivelySafeStaticCall(address,uint256,uint16,bytes)._calldata (util/ExcessivelySafeCall.sol#79) is not in mixedCase
Parameter ExcessivelySafeCall.swapSelector(bytes4,bytes)._newSelector (util/ExcessivelySafeCall.sol#120) is not in mixedCase
Parameter ExcessivelySafeCall.swapSelector(bytes4,bytes)._buf (util/ExcessivelySafeCall.sol#120) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions


BytesLib.toAddress(bytes,uint256) (util/BytesLib.sol#297-306) uses literals with too many digits:
        - tempAddress = mload(uint256)(_bytes + 0x20 + _start) / 0x1000000000000000000000000 (util/BytesLib.sol#302)
ExcessivelySafeCall.slitherConstructorConstantVariable() (util/ExcessivelySafeCall.sol#4-137) uses literals with too many digits:
        - LOW_28_MASK = 0x00000000ffffffffffffffffffffffffffffffffffffffffffffffffffffffff (util/ExcessivelySafeCall.sol#5-6)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits

renounceOwnership() should be declared external:
        - Ownable.renounceOwnership() (../../this-arbitrum-v6.3-d6c7fe7f386b5678ee2597a17e866cee189f056e/node_modules/@openzeppelin/contracts/access/Ownable.sol#61-63)
transferOwnership(address) should be declared external:
        - Ownable.transferOwnership(address) (../../this-arbitrum-v6.3-d6c7fe7f386b5678ee2597a17e866cee189f056e/node_modules/@openzeppelin/contracts/access/Ownable.sol#69-72)
estimateNativeFee(uint16,address,uint256) should be declared external:
        - GNSTokenBridge.estimateNativeFee(uint16,address,uint256) (contracts/GNSTokenBridge.sol#79-92)
lzReceive(uint16,bytes,uint64,bytes) should be declared external:
        - LzApp.lzReceive(uint16,bytes,uint64,bytes) (contracts/LzApp.sol#31-40)
nonblockingLzReceive(uint16,bytes,uint64,bytes) should be declared external:
        - NonblockingLzApp.nonblockingLzReceive(uint16,bytes,uint64,bytes) (contracts/NonblockingLzApp.sol#33-37)
retryMessage(uint16,bytes,uint64,bytes) should be declared external:
        - NonblockingLzApp.retryMessage(uint16,bytes,uint64,bytes) (contracts/NonblockingLzApp.sol#42-52)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

NFTMintingBridge.sol

```
BytesLib.concatStorage(bytes,bytes) (util/BytesLib.sol#91-226) performs a multiplication on the result of a division:
        -sstore(uint256,uint256)(_preBytes,fslot_concatStorage_asm_0 + mload(uint256)(_postBytes + 0x20) / 0x100 ** 32 - mlength_concatStorage_asm_0 * 0x100 ** 32 - newlength_concatStorage_asm_0 + mlength_concatStorage_asm_0 * 2) (util/BytesLib.sol#115-140)
BytesLib.concatStorage(bytes,bytes) (util/BytesLib.sol#91-226) performs a multiplication on the result of a division:
        -sstore(uint256,uint256)(sc_concatStorage_asm_0,mload(uint256)(mc_concatStorage_asm_0) / mask_concatStorage_asm_0 * mask_concatStorage_asm_0) (util/BytesLib.sol#189)
BytesLib.concatStorage(bytes,bytes) (util/BytesLib.sol#91-226) performs a multiplication on the result of a division:
        -sstore(uint256,uint256)(sc_concatStorage_asm_0,mload(uint256)(mc_concatStorage_asm_0) / mask_concatStorage_asm_0 * mask_concatStorage_asm_0) (util/BytesLib.sol#223)
BytesLib.equalStorage(bytes,bytes) (util/BytesLib.sol#439-509) performs a multiplication on the result of a division:
        -fslot_mask_asm_0 = fslot_equalStorage_asm_0 / 0x100 * 0x100 (util/BytesLib.sol#466)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#divide-before-multiply

NFTMintingBridge._buildReleaseMessage(address,uint256).message (contracts/NFTMintingBridge.sol#49) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

LzApp.setPrecrime(address)._precrime (contracts/LzApp.sol#105) lacks a zero-check on :
        - precrime = _precrime (contracts/LzApp.sol#106)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation

Variable 'BytesLib.concatStorage(bytes,bytes).sc_concatStorage_asm_0 (util/BytesLib.sol#147)' in BytesLib.concatStorage(bytes,bytes) (util/BytesLib.sol#91-226) potentially used before declaration: sc_concatStorage_asm_0 = keccak256(uint256,uint256)(0x0,0x20) + slength_concatStorage_asm_0 / 32 (util/BytesLib.sol#195)
Variable 'BytesLib.concatStorage(bytes,bytes).submod_concatStorage_asm_0 (util/BytesLib.sol#161)' in BytesLib.concatStorage(bytes,bytes) (util/BytesLib.sol#91-226) potentially used before declaration: submod_concatStorage_asm_0 = 32 - slengthmod_concatStorage_asm_0 (util/BytesLib.sol#204)
Variable 'BytesLib.concatStorage(bytes,bytes).submod_concatStorage_asm_0 (util/BytesLib.sol#161)' in BytesLib.concatStorage(bytes,bytes) (util/BytesLib.sol#91-226) potentially used before declaration: mc_concatStorage_asm_0 = _postBytes + submod_concatStorage_asm_0 (util/BytesLib.sol#205)
Variable 'BytesLib.concatStorage(bytes,bytes).mc_concatStorage_asm_0 (util/BytesLib.sol#162)' in BytesLib.concatStorage(bytes,bytes) (util/BytesLib.sol#91-226) potentially used before declaration: mc_concatStorage_asm_0 = _postBytes + submod_concatStorage_asm_0 (util/BytesLib.sol#205)
Variable 'BytesLib.concatStorage(bytes,bytes).end_concatStorage_asm_0 (util/BytesLib.sol#163)' in BytesLib.concatStorage(bytes,bytes) (util/BytesLib.sol#91-226) potentially used before declaration: end_concatStorage_asm_0 = _postBytes + mlength_concatStorage_asm_0 (util/BytesLib.sol#206)
Variable 'BytesLib.concatStorage(bytes,bytes).submod_concatStorage_asm_0 (util/BytesLib.sol#161)' in BytesLib.concatStorage(bytes,bytes) (util/BytesLib.sol#91-226) potentially used before declaration: mask_concatStorage_asm_0 = 0x100 ** submod_concatStorage_asm_0 - 1 (util/BytesLib.sol#207)
Variable 'BytesLib.concatStorage(bytes,bytes).mask_concatStorage_asm_0 (util/BytesLib.sol#164)' in BytesLib.concatStorage(bytes,bytes) (util/BytesLib.sol#91-226) potentially used before declaration: mask_concatStorage_asm_0 = 0x100 ** submod_concatStorage_asm_0 - 1 (util/BytesLib.sol#207)
Variable 'BytesLib.concatStorage(bytes,bytes).mask_concatStorage_asm_0 (util/BytesLib.sol#164)' in BytesLib.concatStorage(bytes,bytes) (util/BytesLib.sol#91-226) potentially used before declaration: sstore(uint256,uint256)(sc_concatStorage_asm_0,mload(uint256)(mc_concatStorage_asm_0) + mload(uint256)(mc_concatStorage_asm_0) & mask_concatStorage_asm_0) (util/BytesLib.sol#209)
Variable 'BytesLib.concatStorage(bytes,bytes).sc_concatStorage_asm_0 (util/BytesLib.sol#147)' in BytesLib.concatStorage(bytes,bytes) (util/BytesLib.sol#91-226) potentially used before declaration: sstore(uint256,uint256)(sc_concatStorage_asm_0,mload(uint256)(sc_concatStorage_asm_0) + mload(uint256)(mc_concatStorage_asm_0) & mask_concatStorage_asm_0)) (util/BytesLib.sol#209)
Variable 'BytesLib.concatStorage(bytes,bytes).mc_concatStorage_asm_0 (util/BytesLib.sol#162)' in BytesLib.concatStorage(bytes,bytes) (util/BytesLib.sol#91-226) potentially used before declaration: sstore(uint256,uint256)(sc_concatStorage_asm_0,mload(uint256)(sc_concatStorage_asm_0) + mload(uint256)(mc_concatStorage_asm_0) & mask_concatStorage_asm_0)) (util/BytesLib.sol#209)
Variable 'BytesLib.concatStorage(bytes,bytes).sc_concatStorage_asm_0 (util/BytesLib.sol#147)' in BytesLib.concatStorage(bytes,bytes) (util/BytesLib.sol#91-226) potentially used before declaration: sc_concatStorage_asm_0 = sc_concatStorage_asm_0 + 1 (util/BytesLib.sol#212)
Variable 'BytesLib.concatStorage(bytes,bytes).mc_concatStorage_asm_0 (util/BytesLib.sol#162)' in BytesLib.concatStorage(bytes,bytes) (util/BytesLib.sol#91-226) potentially used before declaration: mc_concatStorage_asm_0 = mc_concatStorage_asm_0 + 0x20 (util/BytesLib.sol#213)
Variable 'BytesLib.concatStorage(bytes,bytes).end_concatStorage_asm_0 (util/BytesLib.sol#163)' in BytesLib.concatStorage(bytes,bytes) (util/BytesLib.sol#91-226) potentially used before declaration: mc_concatStorage_asm_0 < end_concatStorage_asm_0 (util/BytesLib.sol#214)
Variable 'BytesLib.concatStorage(bytes,bytes).mc_concatStorage_asm_0 (util/BytesLib.sol#162)' in BytesLib.concatStorage(bytes,bytes) (util/BytesLib.sol#91-226) potentially used before declaration: mc_concatStorage_asm_0 < end_concatStorage_asm_0 (util/BytesLib.sol#214)
Variable 'BytesLib.concatStorage(bytes,bytes).mask_concatStorage_asm_0 (util/BytesLib.sol#164)' in BytesLib.concatStorage(bytes,bytes) (util/BytesLib.sol#91-226) potentially used before declaration: mask_concatStorage_asm_0 = 0x100 ** mc_concatStorage_asm_0 - end_concatStorage_asm_0 (util/BytesLib.sol#221)
Variable 'BytesLib.concatStorage(bytes,bytes).end_concatStorage_asm_0 (util/BytesLib.sol#163)' in BytesLib.concatStorage(bytes,bytes) (util/BytesLib.sol#91-226) potentially used before declaration: mask_concatStorage_asm_0 = 0x100 ** mc_concatStorage_asm_0 - end_concatStorage_asm_0 (util/BytesLib.sol#221)
Variable 'BytesLib.concatStorage(bytes,bytes).mc_concatStorage_asm_0 (util/BytesLib.sol#162)' in BytesLib.concatStorage(bytes,bytes) (util/BytesLib.sol#91-226) potentially used before declaration: mask_concatStorage_asm_0 = 0x100 ** mc_concatStorage_asm_0 - end_concatStorage_asm_0 (util/BytesLib.sol#221)
Variable 'BytesLib.concatStorage(bytes,bytes).mask_concatStorage_asm_0 (util/BytesLib.sol#164)' in BytesLib.concatStorage(bytes,bytes) (util/BytesLib.sol#91-226) potentially used before declaration: sstore(uint256,uint256)(sc_concatStorage_asm_0,mload(uint256)(mc_concatStorage_asm_0) / mask_concatStorage_asm_0 * mask_concatStorage_asm_0) (util/BytesLib.sol#223)
Variable 'BytesLib.concatStorage(bytes,bytes).sc_concatStorage_asm_0 (util/BytesLib.sol#147)' in BytesLib.concatStorage(bytes,bytes) (util/BytesLib.sol#91-226) potentially used before declaration: sstore(uint256,uint256)(sc_concatStorage_asm_0,mload(uint256)(mc_concatStorage_asm_0) / mask_concatStorage_asm_0 * mask_concatStorage_asm_0)) (util/BytesLib.sol#223)
Variable 'BytesLib.concatStorage(bytes,bytes).mc_concatStorage_asm_0 (util/BytesLib.sol#162)' in BytesLib.concatStorage(bytes,bytes) (util/BytesLib.sol#91-226) potentially used before declaration: sstore(uint256,uint256)(sc_concatStorage_asm_0,mload(uint256)(mc_concatStorage_asm_0) / mask_concatStorage_asm_0 * mask_concatStorage_asm_0)) (util/BytesLib.sol#223)
Variable 'BytesLib.concatStorage(bytes,bytes).sc_concatStorage_asm_0 (util/BytesLib.sol#147)' in BytesLib.concatStorage(bytes,bytes) (util/BytesLib.sol#91-226) potentially used before declaration: sstore(uint256,uint256)(sc_concatStorage_asm_0,mload(uint256)(sc_concatStorage_asm_0) + mload(uint256)(mc_concatStorage_asm_0) & mask_concatStorage_asm_0)) (util/BytesLib.sol#218)
Variable 'BytesLib.concatStorage(bytes,bytes).mc_concatStorage_asm_0 (util/BytesLib.sol#162)' in BytesLib.concatStorage(bytes,bytes) (util/BytesLib.sol#91-226) potentially used before declaration: sstore(uint256,uint256)(sc_concatStorage_asm_0,mload(uint256)(mc_concatStorage_asm_0)) (util/BytesLib.sol#218)
Variable 'BytesLib.concatStorage(bytes,bytes).sc_concatStorage_asm_0 (util/BytesLib.sol#147)' in BytesLib.concatStorage(bytes,bytes) (util/BytesLib.sol#91-226) potentially used before declaration: sc_concatStorage_asm_0 = sc_concatStorage_asm_0 + 1 (util/BytesLib.sol#215)
Variable 'BytesLib.concatStorage(bytes,bytes).mc_concatStorage_asm_0 (util/BytesLib.sol#162)' in BytesLib.concatStorage(bytes,bytes) (util/BytesLib.sol#91-226) potentially used before declaration: mc_concatStorage_asm_0 = mc_concatStorage_asm_0 + 0x20 (util/BytesLib.sol#216)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables

Reentrancy in NFTMintingBridge._nonblockingLzReceive(uint16,bytes,uint64,bytes) (contracts/NFTMintingBridge.sol#115-130):
        External calls:
        - nft.mint(message.recipient,message.tokenId) (contracts/NFTMintingBridge.sol#127)
        Event emitted after the call(s):
        - NFTMintReceived(message.recipient,message.tokenId) (contracts/NFTMintingBridge.sol#129)
Reentrancy in NFTMintingBridge.bridgeNft(uint16,uint256) (contracts/NFTMintingBridge.sol#81-104):
        External calls:
        - nft.burn(tokenId) (contracts/NFTMintingBridge.sol#89)
        - _lzSend(dstChainId,payloadBytes,address(msg.sender),address(0x0),bytes(),msg.value) (contracts/NFTMintingBridge.sol#94-101)
            - lzEndpoint.send{value: _nativeFee}(_dstChainId,trustedRemote,_payload,_refundAddress,_zroPaymentAddress,_adapterParams) (contracts/LzApp.sol#48)
        External calls sending eth:
        - _lzSend(dstChainId,payloadBytes,address(msg.sender),address(0x0),bytes(),msg.value) (contracts/NFTMintingBridge.sol#94-101)
            - lzEndpoint.send{value: _nativeFee}(_dstChainId,trustedRemote,_payload,_refundAddress,_zroPaymentAddress,_adapterParams) (contracts/LzApp.sol#48)
        Event emitted after the call(s):
        - NftBurnReceived(dstChainId,msg.sender,tokenId) (contracts/NFTMintingBridge.sol#103)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3
```

```
LzApp._getGasLimit(bytes) (contracts/LzApp.sol#58-63) uses assembly
        - INLINE ASM (contracts/LzApp.sol#60-62)
BytesLib.concat(bytes,bytes) (util/BytesLib.sol#13-89) uses assembly
        - INLINE ASM (util/BytesLib.sol#23-86)
BytesLib.concatStorage(bytes,bytes) (util/BytesLib.sol#91-226) uses assembly
        - INLINE ASM (util/BytesLib.sol#92-225)
BytesLib.slice(bytes,uint256,uint256) (util/BytesLib.sol#228-295) uses assembly
        - INLINE ASM (util/BytesLib.sol#242-292)
BytesLib.toAddress(bytes,uint256) (util/BytesLib.sol#297-306) uses assembly
        - INLINE ASM (util/BytesLib.sol#301-303)
BytesLib.toUint8(bytes,uint256) (util/BytesLib.sol#308-317) uses assembly
        - INLINE ASM (util/BytesLib.sol#312-314)
BytesLib.toUint16(bytes,uint256) (util/BytesLib.sol#319-328) uses assembly
        - INLINE ASM (util/BytesLib.sol#323-325)
BytesLib.toUint32(bytes,uint256) (util/BytesLib.sol#330-339) uses assembly
        - INLINE ASM (util/BytesLib.sol#334-336)
BytesLib.toUint64(bytes,uint256) (util/BytesLib.sol#341-350) uses assembly
        - INLINE ASM (util/BytesLib.sol#345-347)
BytesLib.toUint96(bytes,uint256) (util/BytesLib.sol#352-361) uses assembly
        - INLINE ASM (util/BytesLib.sol#356-358)
BytesLib.toUint128(bytes,uint256) (util/BytesLib.sol#363-372) uses assembly
        - INLINE ASM (util/BytesLib.sol#367-369)
BytesLib.toUint256(bytes,uint256) (util/BytesLib.sol#374-383) uses assembly
        - INLINE ASM (util/BytesLib.sol#378-380)
BytesLib.toBytes32(bytes,uint256) (util/BytesLib.sol#385-394) uses assembly
        - INLINE ASM (util/BytesLib.sol#389-391)
BytesLib.equal(bytes,bytes) (util/BytesLib.sol#396-437) uses assembly
        - INLINE ASM (util/BytesLib.sol#399-434)
BytesLib.equalStorage(bytes,bytes) (util/BytesLib.sol#439-509) uses assembly
        - INLINE ASM (util/BytesLib.sol#449-506)
ExcessivelySafeCall.excessivelySafeCall(address,uint256,uint16,bytes) (util/ExcessivelySafeCall.sol#23-58) uses assembly
        - INLINE ASM (util/ExcessivelySafeCall.sol#37-56)
ExcessivelySafeCall.excessivelySafeStaticCall(address,uint256,uint16,bytes) (util/ExcessivelySafeCall.sol#75-109) uses assembly
        - INLINE ASM (util/ExcessivelySafeCall.sol#89-107)
ExcessivelySafeCall.swapSelector(bytes4,bytes) (util/ExcessivelySafeCall.sol#120-135) uses assembly
        - INLINE ASM (util/ExcessivelySafeCall.sol#126-134)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Different versions of Solidity is used:
        - Version used: ['0.8.17', '>=0.5.0', '>=0.7.6', '>=0.8.0<0.9.0', '^0.8.0']
        - ^0.8.0 (../../this-arbitrum-v6.3-d6c7fe7f386b5678ee2597a17e866cee189f056e/node_modules/@openzeppelin/contracts/access/Ownable.sol#4)
        - ^0.8.0 (../../this-arbitrum-v6.3-d6c7fe7f386b5678ee2597a17e866cee189f056e/node_modules/@openzeppelin/contracts/security/Pausable.sol#4)
        - ^0.8.0 (../../this-arbitrum-v6.3-d6c7fe7f386b5678ee2597a17e866cee189f056e/node_modules/@openzeppelin/contracts/utils/Context.sol#4)
        - ^0.8.0 (contracts/LzApp.sol#3)
        - 0.8.17 (contracts/NFTMintingBridge.sol#2)
        - ^0.8.0 (contracts/NonblockingLzApp.sol#3)
        - >=0.5.0 (interfaces/ILayerZeroEndpoint.sol#3)
        - >=0.5.0 (interfaces/ILayerZeroReceiver.sol#3)
        - >=0.5.0 (interfaces/ILayerZeroUserApplicationConfig.sol#3)
        - >=0.8.0<0.9.0 (util/BytesLib.sol#9)
        - >=0.7.6 (util/ExcessivelySafeCall.sol#2)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

BytesLib.concat(bytes,bytes) (util/BytesLib.sol#13-89) is never used and should be removed
BytesLib.concatStorage(bytes,bytes) (util/BytesLib.sol#91-226) is never used and should be removed
BytesLib.equal(bytes,bytes) (util/BytesLib.sol#396-437) is never used and should be removed
BytesLib.equalStorage(bytes,bytes) (util/BytesLib.sol#439-509) is never used and should be removed
BytesLib.toAddress(bytes,uint256) (util/BytesLib.sol#297-306) is never used and should be removed
BytesLib.toBytes32(bytes,uint256) (util/BytesLib.sol#385-394) is never used and should be removed
BytesLib.toUint128(bytes,uint256) (util/BytesLib.sol#363-372) is never used and should be removed
BytesLib.toUint16(bytes,uint256) (util/BytesLib.sol#319-328) is never used and should be removed
BytesLib.toUint256(bytes,uint256) (util/BytesLib.sol#374-383) is never used and should be removed
BytesLib.toUint32(bytes,uint256) (util/BytesLib.sol#330-339) is never used and should be removed
BytesLib.toUint64(bytes,uint256) (util/BytesLib.sol#341-350) is never used and should be removed
BytesLib.toUint8(bytes,uint256) (util/BytesLib.sol#308-317) is never used and should be removed
BytesLib.toUint96(bytes,uint256) (util/BytesLib.sol#352-361) is never used and should be removed
Context._msgData() (../../this-arbitrum-v6.3-d6c7fe7f386b5678ee2597a17e866cee189f056e/node_modules/@openzeppelin/contracts/utils/Context.sol#21-23) is never used and should be removed
ExcessivelySafeCall.excessivelySafeStaticCall(address,uint256,uint16,bytes) (util/ExcessivelySafeCall.sol#75-109) is never used and should be removed
ExcessivelySafeCall.swapSelector(bytes4,bytes) (util/ExcessivelySafeCall.sol#120-135) is never used and should be removed
LzApp._blockingLzReceive(uint16,bytes,uint64,bytes) (contracts/LzApp.sol#43) is never used and should be removed
LzApp._checkGasLimit(uint16,uint16,bytes,uint256) (contracts/LzApp.sol#51-56) is never used and should be removed
LzApp._getGasLimit(bytes) (contracts/LzApp.sol#58-63) is never used and should be removed
NonblockingLzApp._nonblockingLzReceive(uint16,bytes,uint64,bytes) (contracts/NonblockingLzApp.sol#40) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version^0.8.0 (../../this-arbitrum-v6.3-d6c7fe7f386b5678ee2597a17e866cee189f056e/node_modules/@openzeppelin/contracts/access/Ownable.sol#4) allows old versions
Pragma version^0.8.0 (../../this-arbitrum-v6.3-d6c7fe7f386b5678ee2597a17e866cee189f056e/node_modules/@openzeppelin/contracts/security/Pausable.sol#4) allows old versions
Pragma version^0.8.0 (../../this-arbitrum-v6.3-d6c7fe7f386b5678ee2597a17e866cee189f056e/node_modules/@openzeppelin/contracts/utils/Context.sol#4) allows old versions
Pragma version^0.8.0 (contracts/LzApp.sol#3) allows old versions
Pragma version0.8.17 (contracts/NFTMintingBridge.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version^0.8.0 (contracts/NonblockingLzApp.sol#3) allows old versions
Pragma version>=0.5.0 (interfaces/ILayerZeroEndpoint.sol#3) allows old versions
Pragma version>=0.5.0 (interfaces/ILayerZeroReceiver.sol#3) allows old versions
Pragma version>=0.5.0 (interfaces/ILayerZeroUserApplicationConfig.sol#3) allows old versions
Pragma version>=0.8.0<0.9.0 (util/BytesLib.sol#9) is too complex
solc-0.8.17 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Parameter LzApp.lzReceive(uint16,bytes,uint64,bytes)._srcChainId (contracts/LzApp.sol#31) is not in mixedCase
Parameter LzApp.lzReceive(uint16,bytes,uint64,bytes)._srcAddress (contracts/LzApp.sol#31) is not in mixedCase
Parameter LzApp.lzReceive(uint16,bytes,uint64,bytes)._nonce (contracts/LzApp.sol#31) is not in mixedCase
Parameter LzApp.lzReceive(uint16,bytes,uint64,bytes)._payload (contracts/LzApp.sol#31) is not in mixedCase
Parameter LzApp.getConfig(uint16,uint16,address,uint256)._version (contracts/LzApp.sol#66) is not in mixedCase
Parameter LzApp.getConfig(uint16,uint16,address,uint256)._chainId (contracts/LzApp.sol#66) is not in mixedCase
Parameter LzApp.getConfig(uint16,uint16,address,uint256)._configType (contracts/LzApp.sol#66) is not in mixedCase
Parameter LzApp.setConfig(uint16,uint16,uint256,bytes)._version (contracts/LzApp.sol#71) is not in mixedCase
Parameter LzApp.setConfig(uint16,uint16,uint256,bytes)._chainId (contracts/LzApp.sol#71) is not in mixedCase
Parameter LzApp.setConfig(uint16,uint16,uint256,bytes)._configType (contracts/LzApp.sol#71) is not in mixedCase
Parameter LzApp.setConfig(uint16,uint16,uint256,bytes)._config (contracts/LzApp.sol#71) is not in mixedCase
Parameter LzApp.setSendVersion(uint16)._version (contracts/LzApp.sol#75) is not in mixedCase
Parameter LzApp.setReceiveVersion(uint16)._version (contracts/LzApp.sol#79) is not in mixedCase
Parameter LzApp.forceResumeReceive(uint16,bytes)._srcChainId (contracts/LzApp.sol#83) is not in mixedCase
Parameter LzApp.forceResumeReceive(uint16,bytes)._srcAddress (contracts/LzApp.sol#83) is not in mixedCase
Parameter LzApp.setTrustedRemote(uint16,bytes)._srcChainId (contracts/LzApp.sol#89) is not in mixedCase
Parameter LzApp.setTrustedRemote(uint16,bytes)._path (contracts/LzApp.sol#89) is not in mixedCase
Parameter LzApp.setTrustedRemoteAddress(uint16,bytes)._remoteChainId (contracts/LzApp.sol#94) is not in mixedCase
Parameter LzApp.setTrustedRemoteAddress(uint16,bytes)._remoteAddress (contracts/LzApp.sol#94) is not in mixedCase
Parameter LzApp.getTrustedRemoteAddress(uint16)._remoteChainId (contracts/LzApp.sol#99) is not in mixedCase
Parameter LzApp.setPrecrime(address)._precrime (contracts/LzApp.sol#105) is not in mixedCase
Parameter LzApp.setMinDstGas(uint16,uint16,uint256)._dstChainId (contracts/LzApp.sol#110) is not in mixedCase
Parameter LzApp.setMinDstGas(uint16,uint16,uint256)._packetType (contracts/LzApp.sol#110) is not in mixedCase
Parameter LzApp.setMinDstGas(uint16,uint16,uint256)._minGas (contracts/LzApp.sol#110) is not in mixedCase
Parameter LzApp.isTrustedRemote(uint16,bytes)._srcChainId (contracts/LzApp.sol#117) is not in mixedCase
Parameter LzApp.isTrustedRemote(uint16,bytes)._srcAddress (contracts/LzApp.sol#117) is not in mixedCase
Parameter NFTMintingBridge.estimateNativeFee(uint16,address,uint256)._dstChainId (contracts/NFTMintingBridge.sol#60) is not in mixedCase
Parameter NonblockingLzApp.nonblockingLzReceive(uint16,bytes,uint64,bytes)._srcChainId (contracts/NonblockingLzApp.sol#33) is not in mixedCase
Parameter NonblockingLzApp.nonblockingLzReceive(uint16,bytes,uint64,bytes)._srcAddress (contracts/NonblockingLzApp.sol#33) is not in mixedCase
Parameter NonblockingLzApp.nonblockingLzReceive(uint16,bytes,uint64,bytes)._nonce (contracts/NonblockingLzApp.sol#33) is not in mixedCase
Parameter NonblockingLzApp.nonblockingLzReceive(uint16,bytes,uint64,bytes)._payload (contracts/NonblockingLzApp.sol#33) is not in mixedCase
Parameter NonblockingLzApp.retryMessage(uint16,bytes,uint64,bytes)._srcChainId (contracts/NonblockingLzApp.sol#42) is not in mixedCase
Parameter NonblockingLzApp.retryMessage(uint16,bytes,uint64,bytes)._srcAddress (contracts/NonblockingLzApp.sol#42) is not in mixedCase
Parameter NonblockingLzApp.retryMessage(uint16,bytes,uint64,bytes)._nonce (contracts/NonblockingLzApp.sol#42) is not in mixedCase
Parameter NonblockingLzApp.retryMessage(uint16,bytes,uint64,bytes)._payload (contracts/NonblockingLzApp.sol#42) is not in mixedCase
Parameter BytesLib.concat(bytes,bytes)._preBytes (util/BytesLib.sol#14) is not in mixedCase
Parameter BytesLib.concat(bytes,bytes)._postBytes (util/BytesLib.sol#15) is not in mixedCase
Parameter BytesLib.concatStorage(bytes,bytes)._preBytes (util/BytesLib.sol#91) is not in mixedCase
Parameter BytesLib.concatStorage(bytes,bytes)._postBytes (util/BytesLib.sol#91) is not in mixedCase
Parameter BytesLib.slice(bytes,uint256,uint256)._bytes (util/BytesLib.sol#229) is not in mixedCase
Parameter BytesLib.slice(bytes,uint256,uint256)._start (util/BytesLib.sol#230) is not in mixedCase
Parameter BytesLib.slice(bytes,uint256,uint256)._length (util/BytesLib.sol#231) is not in mixedCase
Parameter BytesLib.toAddress(bytes,uint256)._bytes (util/BytesLib.sol#297) is not in mixedCase
Parameter BytesLib.toAddress(bytes,uint256)._start (util/BytesLib.sol#297) is not in mixedCase
Parameter BytesLib.toUint8(bytes,uint256)._bytes (util/BytesLib.sol#308) is not in mixedCase
Parameter BytesLib.toUint8(bytes,uint256)._start (util/BytesLib.sol#308) is not in mixedCase
Parameter BytesLib.toUint16(bytes,uint256)._bytes (util/BytesLib.sol#319) is not in mixedCase
Parameter BytesLib.toUint16(bytes,uint256)._start (util/BytesLib.sol#319) is not in mixedCase
Parameter BytesLib.toUint32(bytes,uint256)._bytes (util/BytesLib.sol#330) is not in mixedCase
Parameter BytesLib.toUint32(bytes,uint256)._start (util/BytesLib.sol#330) is not in mixedCase
Parameter BytesLib.toUint64(bytes,uint256)._bytes (util/BytesLib.sol#341) is not in mixedCase
Parameter BytesLib.toUint64(bytes,uint256)._start (util/BytesLib.sol#341) is not in mixedCase
Parameter BytesLib.toUint96(bytes,uint256)._bytes (util/BytesLib.sol#352) is not in mixedCase
Parameter BytesLib.toUint96(bytes,uint256)._start (util/BytesLib.sol#352) is not in mixedCase
Parameter BytesLib.toUint128(bytes,uint256)._bytes (util/BytesLib.sol#363) is not in mixedCase
Parameter BytesLib.toUint128(bytes,uint256)._start (util/BytesLib.sol#363) is not in mixedCase
Parameter BytesLib.toUint256(bytes,uint256)._bytes (util/BytesLib.sol#374) is not in mixedCase
Parameter BytesLib.toUint256(bytes,uint256)._start (util/BytesLib.sol#374) is not in mixedCase
Parameter BytesLib.toBytes32(bytes,uint256)._bytes (util/BytesLib.sol#385) is not in mixedCase
Parameter BytesLib.toBytes32(bytes,uint256)._start (util/BytesLib.sol#385) is not in mixedCase
Parameter BytesLib.equal(bytes,bytes)._preBytes (util/BytesLib.sol#396) is not in mixedCase
Parameter BytesLib.equal(bytes,bytes)._postBytes (util/BytesLib.sol#396) is not in mixedCase
Parameter BytesLib.equalStorage(bytes,bytes)._preBytes (util/BytesLib.sol#440) is not in mixedCase
Parameter BytesLib.equalStorage(bytes,bytes)._postBytes (util/BytesLib.sol#441) is not in mixedCase
Parameter ExcessivelySafeCall.excessivelySafeCall(address,uint256,uint16,bytes)._target (util/ExcessivelySafeCall.sol#24) is not in mixedCase
Parameter ExcessivelySafeCall.excessivelySafeCall(address,uint256,uint16,bytes)._gas (util/ExcessivelySafeCall.sol#25) is not in mixedCase
Parameter ExcessivelySafeCall.excessivelySafeCall(address,uint256,uint16,bytes)._maxCopy (util/ExcessivelySafeCall.sol#26) is not in mixedCase
Parameter ExcessivelySafeCall.excessivelySafeCall(address,uint256,uint16,bytes)._calldata (util/ExcessivelySafeCall.sol#27) is not in mixedCase
Parameter ExcessivelySafeCall.excessivelySafeStaticCall(address,uint256,uint16,bytes)._target (util/ExcessivelySafeCall.sol#76) is not in mixedCase
Parameter ExcessivelySafeCall.excessivelySafeStaticCall(address,uint256,uint16,bytes)._gas (util/ExcessivelySafeCall.sol#77) is not in mixedCase
Parameter ExcessivelySafeCall.excessivelySafeStaticCall(address,uint256,uint16,bytes)._maxCopy (util/ExcessivelySafeCall.sol#78) is not in mixedCase
Parameter ExcessivelySafeCall.excessivelySafeStaticCall(address,uint256,uint16,bytes)._calldata (util/ExcessivelySafeCall.sol#79) is not in mixedCase
Parameter ExcessivelySafeCall.swapSelector(bytes4,bytes)._newSelector (util/ExcessivelySafeCall.sol#120) is not in mixedCase
Parameter ExcessivelySafeCall.swapSelector(bytes4,bytes)._buf (util/ExcessivelySafeCall.sol#120) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

BytesLib.toAddress(bytes,uint256) (util/BytesLib.sol#297-306) uses literals with too many digits:
        - tempAddress = mload(uint256)(_bytes + 0x20 + _start) / 0x1000000000000000000000000 (util/BytesLib.sol#302)
ExcessivelySafeCall.slitherConstructorConstantVariables() (util/ExcessivelySafeCall.sol#4-137) uses literals with too many digits:
        - LOW_28_MASK = 0x00000000ffffffffffffffffffffffffffffffffffffffffffffffffffffffff (util/ExcessivelySafeCall.sol#5-6)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits
```

```
renounceOwnership() should be declared external:
        - Ownable.renounceOwnership() (../../this-arbitrum-v6.3-d6c7fe7f386b5678ee2597a17e866cee189f056e/node_modules/@openzeppelin/contracts/access/Ownable.sol#61-63)
transferOwnership(address) should be declared external:
        - Ownable.transferOwnership(address) (../../this-arbitrum-v6.3-d6c7fe7f386b5678ee2597a17e866cee189f056e/node_modules/@openzeppelin/contracts/access/Ownable.sol#69-72)
lzReceive(uint16,bytes,uint64,bytes) should be declared external:
        - LzApp.lzReceive(uint16,bytes,uint64,bytes) (contracts/LzApp.sol#31-40)
estimateNativeFee(uint16,address,uint256) should be declared external:
        - NFTMintingBridge.estimateNativeFee(uint16,address,uint256) (contracts/NFTMintingBridge.sol#59-72)
nonblockingLzReceive(uint16,bytes,uint64,bytes) should be declared external:
        - NonblockingLzApp.nonblockingLzReceive(uint16,bytes,uint64,bytes) (contracts/NonblockingLzApp.sol#33-37)
retryMessage(uint16,bytes,uint64,bytes) should be declared external:
        - NonblockingLzApp.retryMessage(uint16,bytes,uint64,bytes) (contracts/NonblockingLzApp.sol#42-52)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

- The majority of the issues identified are related to reentrancy, low level calls, multiplication on the result of a division or timestamp usage for comparison.
- Block.timestamp should not be used because it can be manipulated by miners.
- All the reentrancies flagged are false positives.
- Several informational issues related to solidity naming convention were identified.
- No major issues were found by Slither.

4.2 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:

GToken.sol

Line	SWC Title	Severity	Short Description
40	(SWC-108) State Variable Default Visibility	Low	State variable visibility is not set.
352	(SWC-116) Timestamp Dependence	Low	A control flow decision is made based on The block.timestamp environment variable.
363	(SWC-116) Timestamp Dependence	Low	A control flow decision is made based on The block.timestamp environment variable.

GTokenOpenPnlFeed.sol

Line	SWC Title	Severity	Short Description
30	(SWC-110) Assert Violation	Unknown	Public state variable with array type causing reacheable exception by default.
35	(SWC-110) Assert Violation	Unknown	Public state variable with array type causing reacheable exception by default.
42	(SWC-110) Assert Violation	Unknown	Public state variable with array type causing reacheable exception by default.
106	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "%" discovered
162	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "%" discovered
171	(SWC-110) Assert Violation	Unknown	Out of bounds array access
197	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
198	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
212	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
216	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
234	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
239	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
245	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
248	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
250	(SWC-110) Assert Violation	Unknown	Out of bounds array access
337	(SWC-110) Assert Violation	Unknown	Out of bounds array access
344	(SWC-110) Assert Violation	Unknown	Out of bounds array access
346	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
346	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
347	(SWC-110) Assert Violation	Unknown	Out of bounds array access
348	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
354	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
360	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "%" discovered
361	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
361	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
361	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
361	(SWC-110) Assert Violation	Unknown	Out of bounds array access
361	(SWC-101) Integer Overflow and Underflow	Unknown	Compiler-rewritable "<uint> - 1" discovered
362	(SWC-110) Assert Violation	Unknown	Out of bounds array access
362	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
368	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
369	(SWC-110) Assert Violation	Unknown	Out of bounds array access
369	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
372	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered

TWAPPriceGetter.sol

Line	SWC Title	Severity	Short Description
10	(SWC-108) State Variable Default Visibility	Low	State variable visibility is not set.

GNSTokenBridge.sol

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

NFTMintingBridge.sol

No issues found by MythX.

- The majority of the issues identified are related to arithmetic operations, using of block.timestamp, variable visibility and out of bounds array access.
- Integer Overflows and Underflows flagged by MythX are false positives, as those contracts are using Solidity ^0.8.0 version. After the Solidity version 0.8.0 Arithmetic operations revert to underflow and overflow by default.
- No major issues were found by MythX.

THANK YOU FOR CHOOSING

// HALBORN