// HALBORN

Convergence Finance - Convergence Protocol

Smart Contract Security Assessment

Prepared by: **Halborn**

Date of Engagement: **April 24th, 2023 - June 9th, 2023**

Visit: **Halborn.com**

DOCUMENT REVISION HISTORY

VERSION	MODIFICATION	DATE	AUTHOR
0.1	Document Creation	02/09/2023	Manuel Garcia
0.2	Document Updates	06/09/2023	Manuel Garcia
0.3	Draft Version	06/09/2023	Manuel Garcia
0.4	Draft Review	06/12/2023	Grzegorz Trawinski
0.5	Draft Review	06/12/2023	Piotr Cielas
0.6	Draft Review	06/12/2023	Gabi Urrutia
1.0	Remediation Plan	07/06/2023	Manuel Garcia
1.1	Remediation Plan	07/28/2023	Manuel Garcia
1.2	Remediation Plan Review	07/28/2023	Piotr Cielas
1.3	Remediation Plan Review	07/30/2023	Gabi Urrutia

CONTACTS

CONTACT	COMPANY	EMAIL
Rob Behnke	Halborn	Rob.Behnke@halborn.com
Steven Walbroehl	Halborn	Steven.Walbroehl@halborn.com
Gabi Urrutia	Halborn	Gabi.Urrutia@halborn.com
Piotr Cielas	Halborn	Piotr.Cielas@halborn.com
Grzegorz Trawinski	Halborn	Grzegorz.Trawinski@halborn.com
Manuel Garcia	Halborn	Manuel.Diaz@halborn.com

EXECUTIVE OVERVIEW

1.1 INTRODUCTION

The Convergence protocol is a protocol aggregator that intends to allow users to stake their tokens from other protocols to generate yield.

Convergence Finance engaged Halborn to conduct a security assessment on their smart contracts beginning on April 24th, 2023 and ending on June 9th, 2023. The security assessment was scoped to the smart contracts provided in the Convergence-fi/contracts-audit GitHub repository. Commit hashes and further details can be found in the Scope section of this report.

1.2 ASSESSMENT SUMMARY

The team at Halborn was provided 7 weeks for the engagement and assigned 1 full-time security engineer to verify the security of the smart contracts in scope. The security engineer is a blockchain and smart contract security expert with advanced penetration testing and smart contract hacking skills, and deep knowledge of multiple blockchain protocols.

The purpose of the assessments is to:

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

In summary, Halborn identified some improvements to reduce the likelihood and impact of multiple risks, which have been mostly addressed by Convergence Finance . The main ones are the following:

- Increasing the invested stable variable value to prevent bypassing the merkle tree limit.
- Prevent revoked schedules from being used to release CVG.

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 assessment. While manual testing is recommended to uncover flaws in logic, process, and implementation; automated testing techniques help enhance coverage of the code and can quickly identify items that do not follow the security best practices. The following phases and associated tools were used during the assessment:

- 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 hot-spots or bugs. (MythX)
- Static Analysis of security for scoped contract, and imported functions. (Slither)
- Testnet deployment (Brownie, Remix IDE, Foundry)

2. RISK METHODOLOGY

Every vulnerability and issue observed by Halborn is ranked based on **two sets** of **Metrics** and a **Severity Coefficient**. This system is inspired by the industry standard Common Vulnerability Scoring System.

The two **Metric sets** are: **Exploitability** and **Impact**. **Exploitability** captures the ease and technical means by which vulnerabilities can be exploited and **Impact** describes the consequences of a successful exploit.

The **Severity Coefficients** is designed to further refine the accuracy of the ranking with two factors: **Reversibility** and **Scope**. These capture the impact of the vulnerability on the environment as well as the number of users and smart contracts affected.

The final score is a value between 0-10 rounded up to 1 decimal place and 10 corresponding to the highest security risk. This provides an objective and accurate rating of the severity of security vulnerabilities in smart contracts.

The system is designed to assist in identifying and prioritizing vulnerabilities based on their level of risk to address the most critical issues in a timely manner.

2.1 EXPLOITABILITY

Attack Origin (AO):

Captures whether the attack requires compromising a specific account.

Attack Cost (AC):

Captures the cost of exploiting the vulnerability incurred by the attacker relative to sending a single transaction on the relevant blockchain. Includes but is not limited to financial and computational cost.

Attack Complexity (AX):

Describes the conditions beyond the attacker's control that must exist in order to exploit the vulnerability. Includes but is not limited to macro situation, available third-party liquidity and regulatory challenges.

Metrics:

Exploitability Metric (m_E)	Metric Value	Numerical Value
Attack Origin (AO)	Arbitrary (AO:A) Specific (AO:S)	1 0.2
Attack Cost (AC)	Low (AC:L) Medium (AC:M) High (AC:H)	1 0.67 0.33
Attack Complexity (AX)	Low (AX:L) Medium (AX:M) High (AX:H)	1 0.67 0.33

Exploitability E is calculated using the following formula:

$$E = \prod m_e$$

2.2 IMPACT

Confidentiality (C):

Measures the impact to the confidentiality of the information resources managed by the contract due to a successfully exploited vulnerability. Confidentiality refers to limiting access to authorized users only.

Integrity (I):

Measures the impact to integrity of a successfully exploited vulnerability. Integrity refers to the trustworthiness and veracity of data stored and/or processed on-chain. Integrity impact directly affecting Deposit or Yield records is excluded.

Availability (A):

Measures the impact to the availability of the impacted component resulting from a successfully exploited vulnerability. This metric refers to smart contract features and functionality, not state. Availability impact directly affecting Deposit or Yield is excluded.

Deposit (D):

Measures the impact to the deposits made to the contract by either users or owners.

Yield (Y):

Measures the impact to the yield generated by the contract for either users or owners.

Metrics:

Impact Metric (m_I)	Metric Value	Numerical Value
Confidentiality (C)	None (I:N)	0
	Low (I:L)	0.25
	Medium (I:M)	0.5
	High (I:H)	0.75
	Critical (I:C)	1
Integrity (I)	None (I:N)	0
	Low (I:L)	0.25
	Medium (I:M)	0.5
	High (I:H)	0.75
	Critical (I:C)	1
Availability (A)	None (A:N)	0
	Low (A:L)	0.25
	Medium (A:M)	0.5
	High (A:H)	0.75
	Critical	1
Deposit (D)	None (D:N)	0
	Low (D:L)	0.25
	Medium (D:M)	0.5
	High (D:H)	0.75
	Critical (D:C)	1
Yield (Y)	None (Y:N)	0
	Low (Y:L)	0.25
	Medium: (Y:M)	0.5
	High: (Y:H)	0.75
	Critical (Y:H)	1

Impact I is calculated using the following formula:

$$I = max(m_I) + \frac{\sum m_I - max(m_I)}{4}$$

2.3 SEVERITY COEFFICIENT

Reversibility (R):

Describes the share of the exploited vulnerability effects that can be reversed. For upgradeable contracts, assume the contract private key is available.

Scope (S):

Captures whether a vulnerability in one vulnerable contract impacts resources in other contracts.

Coefficient (C)	Coefficient Value	Numerical Value
Reversibility (r)	None (R:N) Partial (R:P) Full (R:F)	1 0.5 0.25
Scope (s)	Changed (S:C) Unchanged (S:U)	1.25 1

Severity Coefficient C is obtained by the following product:

$$C = rs$$

The Vulnerability Severity Score S is obtained by:

$$S = min(10, EIC * 10)$$

The score is rounded up to 1 decimal places.

Severity	Score Value Range
Critical	9 - 10
High	7 - 8.9
Medium	4.5 - 6.9
Low	2 - 4.4
Informational	0 - 1.9

2.4 SCOPE

Code repositories:

1. Convergence Protocol

- Repository: Convergence-fi/contracts-audit
- Commit ID: e1a661dc699393ec879bbf98ffb31d02753033ef - Remmediation Plan: 225258abc90206315302ade5c2d8701cb792cbf3
- Smart contracts in scope:
    1. BondCalculator.sol (contracts/Bond/BondCalculator.sol)
    2. BondDepository.sol (contracts/Bond/BondDepository.sol)
    3. BondLogo.sol (contracts/Bond/BondLogo.sol)
    4. BondPositionManager.sol (contracts/Bond/BondPositionManager.sol)
    5. GaugeController.vy (contracts/Locking/GaugeController.vy)
    6. LockingLogo.sol (contracts/Locking/LockingLogo.sol)
    7. LockingPositionDelegate.sol (contracts/Locking/LockingPositionDelegate.sol)
    8. LockingPositionManager.sol (contracts/Locking/LockingPositionManager.sol)
    9. veCVG.vy (contracts/Locking/veCVG.vy)
    10. VveCVGCalculator.sol (contracts/Locking/VveCVGCalculator.sol)
    11. CvgOracle.sol (contracts/Oracles/CvgOracle.sol)
    12. CvgV3Aggregator.sol (contracts/Oracles/CvgV3Aggregator.sol)
    13. SeedPresaleCvg.sol (contracts/PresaleVesting/SeedPresaleCvg.sol)
    14. VestingCvg.sol (contracts/PresaleVesting/VestingCvg.sol)
    15. WlPresaleCvg.sol (contracts/PresaleVesting/WlPresaleCvg.sol)
    16. CvgRewards.sol (contracts/Rewards/CvgRewards.sol)
    17. TAssetBlackHole.sol (contracts/Rewards/TAssetBlackHole.sol)
    18. YsDistributor.sol (contracts/Rewards/YsDistributor.sol)
    19. CvgTokeStaking.sol (contracts/Staking/CvgTokeStaking.sol)
    20. StakingLogo.sol (contracts/Staking/StakingLogo.sol)
    21. StakingViewer.sol (contracts/Staking/StakingViewer.sol)
    22. TAssetStaking.sol (contracts/Staking/TAssetStaking.sol)

23. TokeStaker.sol (contracts/Staking/TokeStaker.sol)
24. TokeStakingCommon.sol (contracts/Staking/TokeStakingCommon.sol)
25. Cvg.sol (contracts/Token/Cvg.sol)
26. CvgERC721TimeLocking.sol (contracts/Token/CvgERC721TimeLocking.sol)
27. CvgToke.sol (contracts/Token/CvgToke.sol)
28. CvgUtilities.sol (contracts/Token/CvgUtilities.sol)
29. SwapperFactory.sol (contracts/Token/SwapperFactory.sol)
30. CloneFactory.sol (contracts/CloneFactory.sol)
31. CvgControlTower.sol (contracts/CvgControlTower.sol)

Out-of-scope:

- third-party libraries and dependencies
- economic attacks
- manual interactions with other protocols
- attacks resulting from centralization risk

3. ASSESSMENT SUMMARY & FINDINGS OVERVIEW

CRITICAL	HIGH	MEDIUM	LOW	INFORMATIONAL
2	0	3	3	0

SECURITY ANALYSIS	RISK LEVEL	REMEDIATION DATE
(HAL-01) MERKLE TREE INVESTMENT LIMIT CAN BE BYPASSED	Critical (10)	SOLVED - 07/06/2023
(HAL-02) REVOKED SCHEDULES CAN BE USED TO RELEASE CVG	Critical (10)	SOLVED - 07/06/2023
(HAL-03) ORACLE RESPONSE NOT CHECKED FOR STALE PRICES	Medium (5.0)	SOLVED - 07/06/2023
(HAL-04) PRICE FEED AGGREGATOR NOT RETURNING ADDITIONAL PARAMETERS	Medium (5.0)	RISK ACCEPTED
(HAL-05) CENTRALIZATION RISK	Medium (5.0)	RISK ACCEPTED
(HAL-06) NFT TIME LOCKING MECHANISM CAN BE BYPASSED	Low (3.3)	SOLVED - 07/06/2023
(HAL-07) ROUNDING ERROR WHEN COMPUTING RELEASABLE AMOUNT	Low (2.5)	SOLVED - 07/28/2023
(HAL-08) VESTING SCHEDULES WITH AN AMOUNT LOWER THAN MAXIMUM SUPPLY REVERT	Low (2.0)	SOLVED - 07/06/2023

FINDINGS & TECH DETAILS

4.1 (HAL-01) MERKLE TREE INVESTMENT LIMIT CAN BE BYPASSED - CRITICAL(10)

Description:

The `WlPresaleCvg` contract allows users to buy CVG tokens if they are whitelisted in a Merkle tree. Currently, there are 3 Merkle trees, each one with a different investment limit:

- Small with a maximum of 800 * 10e18 CVG tokens.
- Medium with a maximum of 4,000 * 10e18 CVG tokens.
- Large with a maximum of 8,000 * 10e18 CVG tokens.

Users invest by sending the amount to invest, Merkle proof and the type of Merkle tree to the `investMint` function. This function checks that the amount is below the list type limit and mints a position NFT to the user.

Moreover, the `refillToken` function allows users to refill a position NFT as long as the new total amount does not exceed the Merkle type limit. However, although this function properly increases the `cvgRedeemable` variable, it does not properly increase the `stableInvested` amount used to determine whether an investment has exceeded the limit.

This allows a malicious user to invest the minimum amount required in order to create a position NFT and later call `refillToken` multiple times, allowing them to retrieve the whole CVG Tokens, not only bypassing their Merkle tree limits but also leaving other users in the Merkle trees without any tokens.

Code Location:

Listing 1: contracts/PresaleVesting/WlPresaleCvg.sol

```
232 function refillToken(
233     uint256 _tokenId,
234     uint256 _amount,
235     bool _isDai
236 ) external {
237     require(ownerOf(_tokenId) == msg.sender, "NOT_OWNED");
238 
239     IERC20 token = _isDai ? Dai : Frax;
240 
241     uint256 _vestingType = presaleInfos[_tokenId].vestingType;
242     uint256 cvgAmount = (_amount * NUMERATOR) / PRICE_WL;
243 
244     wlParams[_vestingType].cvgRedeemable += cvgAmount;
245 
246     require(
247         _amount + presaleInfos[_tokenId].stableInvested <=
248             wlParams[_vestingType].maxInvest,
249         "TOO_MUCH_Q_WL"
250     );
251 
252     /// @dev update the presales info for this address, only
 ↳ change cvgAmount
253     presaleInfos[_tokenId].cvgAmount += cvgAmount;
254 
255     /// @dev Update available supply
256     supply -= cvgAmount;
257 
258     /// @dev Transfer
259     token.transferFrom(msg.sender, address(this), _amount);
260 }
```

BVSS:

AO:A/AC:L/AX:L/C:N/I:C/A:C/D:N/Y:C/R:N/S:U (10)

Proof of Concept:

1. A malicious user calls `investMint()` to create a new position.
2. Malicious users call `refillToken()` multiple times, bypassing the Merkle tree limit and ending with all CVG Token supply.
3. Now, legitimate users can no longer receive their CVG tokens.

```
Running 1 test for test/Presale/WlPresaleCvg.t.sol:WlPresaleCvgTest
[FAIL. Reason: Assertion failed.] testFail_HAL01_WlPresale_POC_Bypass_WL_Type_Limit_Refill() (gas: 457591)
Logs:
  Users calls investMint() for a medium size list with 1_000 * 1e18 stable, NFT ID 1. Max investment: 4_000 * 1e18
  NFT ID 1 CVG Amount:  4545  * 1e18
  Calling refillToken with 3_000 * 1e18.
  NFT ID 1 CVG Amount:  18181  * 1e18
  Calling refillToken with 3_000 * 1e18.
  NFT ID 1 CVG Amount:  31818  * 1e18
  Calling refillToken with 3_000 * 1e18.
  NFT ID 1 CVG Amount:  45454  * 1e18

Test result: FAILED. 0 passed; 1 failed; finished in 3.76ms
```

Recommendation:

It is recommended to increase the `stableInvested` variable in the `refillToken` function.

Remediation Plan:

SOLVED: The Convergence Finance team fixed the issue by increasing the `stableInvested` amount when calling `refillToken()` in commit `20414f9`.

4.2 (HAL-02) REVOKED SCHEDULES CAN BE USED TO RELEASE CVG - CRITICAL(10)

Description:

In the VestingCvg contract, the revokeVestingSchedule function allows revoking an existing vesting schedule. This function reduces the vestingSchedulesTotalAmount by the amount of CVG pending release.

However, when releasing the CVG with any of the available functions (releaseSeed, releaseWl or releaseTeamOrDao) the last vesting schedule is retrieved without checking the revoked variable. Even if a new schedule is introduced, if a user calls any of the release functions while the current schedule is revoked, the user can release CVG tokens with the revoked schedule and the released amount is deducted from the vestingSchedulesTotalAmount, causing an underflow for other legitimate user when they are trying to release their tokens.

Code Location:

Listing 2: contracts/PresaleVesting/VestingCvg.sol

```
238 function releaseSeed(uint256 _tokenId) external onlyOwnerOfSeed(
 ↳ _tokenId) {
239     (
240         uint256 amountToRelease,
241         ,
242         ,
243         uint256 vestingScheduleId
244     ) = _computeReleaseAmount(_tokenId, true);
245     require(amountToRelease > 0, "NOT_RELEASABLE"); //@audit Not
 ↳ checking if revoked.
246 
247     //update totalReleased & amountReleasedId &
 ↳ vestingSchedulesTotalAmount
248     vestingSchedules[vestingScheduleId].totalReleased +=
 ↳ amountToRelease;
249 
250     amountReleasedIdSeed[_tokenId] += amountToRelease;
251 
252     vestingSchedulesTotalAmount -= amountToRelease;
253 
254     //transfer Cvg amount to release
255     cvg.transfer(msg.sender, amountToRelease);
256 }
```

BVSS:

AO:A/AC:L/AX:L/C:N/I:N/A:N/D:N/Y:C/R:N/S:U (10)

Proof of Concept:

1. An existing vesting schedule is revoked by the administrator.
2. A user with that vesting schedule assigned releases his vested CVG.
3. Vesting schedule gets executed even if it is revoked.
4. When the total released amount approaches the limit, the release functions will underflow for other legitimate users.

```
Running 1 test for test/Presale/VestingCvg.t.sol:VestingCvgTest
[FAIL. Reason: Assertion failed.] testFail_HAL02_Vesting_SeedVesting_Revoke() (gas: 1216895)
Logs:
  Granting preseed of 10_000 * 1e18 to ACTOR 1
  Actor 1 invests 10_000 * 1e18.
  Total CVG 100_000 * 1e18.
  Sale state is now over.
  Creating a vesting schedule with a Total CVG of 200_000 * 1e18.
  Revoking vesting schedule.
  Waiting for cliff...
  Calling releaseSeed(1)
  Waiting for slice 1...
  Calling releaseSeed(1)
  6250000000000000000000
  Waiting for slice 2...
  Calling releaseSeed(1)
  7500000000000000000000
  Waiting for slice 3...
  Calling releaseSeed(1)
  8750000000000000000000
  Waiting for slice 4...
  Calling releaseSeed(1)
  Final CVG balance of Actor 1:  100000000000000000000000

Test result: FAILED. 0 passed; 1 failed; finished in 4.62ms
```

Recommendation:

It is recommended to implement a check to prevent users from releasing CVG if their assigned schedule has been revoked.

Remediation Plan:

SOLVED: The Convergence Finance team solved this issue by preventing the release functions from being called with a revoked schedule in commit 20414f9.

4.3 (HAL-03) ORACLE RESPONSE NOT CHECKED FOR STALE PRICES - MEDIUM (5.0)

Description:

In the CvgOracle contract, the getPriceAggregator function allows the oracle to retrieve the price from a Chainlink aggregator. However, this function is not retrieving the answeredInRound and timestamp parameters to check if the prices are stale. This could lead to the oracle using stale prices in the event that the Chainlink oracle is not being updated.

Code Location:

Listing 3: contracts/Oracles/CvgOracle.sol

```
163 function getPriceAggregator(AggregatorV3Interface aggregator)
 ↳ public view returns (uint256) {
164     (, int256 chainlinkPrice, , , ) = aggregator.latestRoundData()
 ↳ ;
165     return uint256(chainlinkPrice) * 10 ** (18 - aggregator.
 ↳ decimals());
166 }
```

BVSS:

AO:A/AC:L/AX:H/C:N/I:C/A:N/D:C/Y:C/R:N/S:U (5.0)

Recommendation:

Make sure the prices returned by the Chainlink aggregator are not stale.

Listing 4: contracts/Oracles/CvgOracle.sol

```
193 function getPriceAggregator(
194     AggregatorV3Interface aggregator
195 ) public view returns (uint256) {
196     (
197         uint80 chain,
198         int256 chainlinkPrice,
199         ,
200         uint256 updatedAt,
201         uint80 answeredInRound
202     ) = aggregator.latestRoundData();
203 
204     require(answeredInRound >= roundID, "Stale price");
205     require(chainlinkPrice > 0, " Error.NEGATIVE_PRICE");
206     require(
207         block.timestamp <= updatedAt + stalePriceDelay,
208         Error.STALE_PRICE
209     );
210 
211     return uint256(chainlinkPrice) * 10 ** (18 - aggregator.
 ↳ decimals());
212 }
```

Remediation Plan:

SOLVED: The Convergence Finance team fixed the issue by checking for stale prices in the `getAndVerifyPrice()` function in commit `20414f9`.

4.4 (HAL-04) PRICE FEED AGGREGATOR NOT RETURNING ADDITIONAL PARAMETERS - MEDIUM (5.0)

Description:

In the CvgV3Aggregator contract, the latestRoundData function returns 0 for the roundId, startedAt, updatedAt, answeredInRound parameters.

This does not only disallow to check for stale prices, but it also prevents the aggregator from working with standard contracts that check for stale prices.

Code Location:

Listing 5: contracts/Oracles/CvgV3Aggregator.sol

```
52 function latestRoundData()
53     external
54     view
55     returns (uint80 roundId, int256 answer, uint256 startedAt,
↳ uint256 updatedAt, uint80 answeredInRound)
56 {
57     return (0, latestPrice, 0, 0, 0);
58 }
```

BVSS:

AO:A/AC:L/AX:H/C:N/I:C/A:N/D:C/Y:C/R:N/S:U (5.0)

Recommendation:

Return the proper values for each of the described parameters.

Remediation Plan:

RISK ACCEPTED: The Convergence Finance team accepted the risk of this issue.

4.5 (HAL-05) CENTRALIZATION RISK - MEDIUM (5.0)

Description:

In the current protocol model, the interactions with the aggregated protocol are performed manually through a multi-signature wallet. This means that all the funds received from users using the protocol are transferred to this wallet. This poses a concerning centralization risk as the user's funds are sent to a private wallet.

Moreover, if the private keys for the multi-signature wallets are ever compromised, all protocol funds could be stolen by a malicious actor.

BVSS:

AO:S/AC:L/AX:L/C:C/I:C/A:C/D:C/Y:C/R:N/S:C (5.0)

Recommendation:

Include as many interactions as possible into the smart contract's logic in order to decentralize the protocol.

Remediation Plan:

RISK ACCEPTED: The Convergence Finance team accepted the risk of this issue.

4.6 (HAL-06) NFT TIME LOCKING MECHANISM CAN BE BYPASSED - LOW (3.3)

Description:

The CvgERC721TimeLocking contract used to mint position NFTS implements a time-locking mechanism which allows users to lock the NFT, so no rewards can be claimed in order to place it for sale. This prevents a malicious user from front-running a sale transaction and claiming all rewards just before selling it.

However, the user can change the lock time frame at any time as long as the new timestamp is greater than the block timestamp. This prevents users from front-running with a new timestamp equal to the block timestamp, which would allow claiming rewards before selling.

However, it is possible for a malicious user to determine whether the sell transaction is going to execute in a different block and perform a double front-run, unlocking the NFT in the first block and immediately claiming the rewards in the next one.

Code Location:

Listing 6: contracts/Token/CvgERC721TimeLocking.sol (Line 63)

```
62 function setLock(uint256 tokenId, uint256 timestamp) external
↳ onlyNftOwner(tokenId) {
63     require(timestamp > block.timestamp && timestamp - block.
↳ timestamp < maxLockingTime, "WRONG_TIME_LOCK");
64     unlockingTimestampPerToken[tokenId] = timestamp;
65 }
```

BVSS:

AO:A/AC:L/AX:H/C:N/I:N/A:N/D:N/Y:C/R:N/S:U (3.3)

Recommendation:

Implement a buffer where the new timestamp cannot be earlier than a few minutes from the block timestamp.

Remediation Plan:

SOLVED: The Convergence Finance team fixed this issue by adding a buffer when changing the lock timestamp in commit 20414f9.

4.7 (HAL-07) ROUNDING ERROR WHEN COMPUTING RELEASABLE AMOUNT - LOW (2.5)

Description:

The calculateRelease function in the VestingCVG contract is in charge of computing the releasable amount on each slice of the vesting schedule. However, there is a small rounding error in the calculation of the release amount for each slice.

This results in slices releasing fewer tokens than what they should.

Code Location:

Listing 7: contracts/PresaleVesting/VestingCvg.sol (Lines 392-405)

```
378 function calculateRelease(
379     uint256 vestingSchedulesId,
380     uint256 totalAmount,
381     uint256 totalAmountReleased
382 ) private view returns (uint256 amountToRelease) {
383     uint256 amountReleasable;
384     uint256 slices = vestingSchedules[vestingSchedulesId].slices;
385     uint256 slicePeriod = vestingSchedules[vestingSchedulesId].
 ↳ slicePeriods;
386     uint256 releaseTimestamp = vestingSchedules[vestingSchedulesId
 ↳ ].cliff;
387     uint256 dropCliff = vestingSchedules[vestingSchedulesId].
 ↳ dropCliff;
388
389     uint256 amountDropCliff = (totalAmount * dropCliff) / 1000;
390     uint256 endRelease = releaseTimestamp + slices * slicePeriod;
391
392     if (block.timestamp >= releaseTimestamp) {
393         uint256 actualSlice = (block.timestamp - releaseTimestamp)
 ↳  / (slicePeriod);
394
395         if (slices <= actualSlice) {
```

```
396             amountReleasable = totalAmount;
397         } else {
398             uint256 a = (totalAmount - amountDropCliff) / (
 ↳ endRelease - releaseTimestamp);
399             int256 b = int256(amountDropCliff) - int256(a *
 ↳ releaseTimestamp);
400             uint256 x = releaseTimestamp + actualSlice *
 ↳ slicePeriod;
401 
402             amountReleasable = uint256(int256(a) * int256(x) + b);
403         }
404         amountToRelease = amountReleasable - totalAmountReleased;
405     }
406 }
```

BVSS:

AO:A/AC:L/AX:L/C:N/I:N/A:N/D:N/Y:L/R:N/S:U (2.5)

Recommendation:

Please see below for an optimized formula that simplifies the calculations, achieving the same result without rounding errors:

Listing 8: contracts/PresaleVesting/VestingCvg.sol (Lines 392-402)

```
378 function calculateRelease(
379     uint256 vestingSchedulesId,
380     uint256 totalAmount,
381     uint256 totalAmountReleased
382 ) private view returns (uint256 amountToRelease) {
383     uint256 amountReleasable;
384     uint256 slices = vestingSchedules[vestingSchedulesId].slices;
385     uint256 slicePeriod = vestingSchedules[vestingSchedulesId].
 ↳ slicePeriods;
386     uint256 releaseTimestamp = vestingSchedules[vestingSchedulesId
 ↳ ].cliff;
387     uint256 dropCliff = vestingSchedules[vestingSchedulesId].
 ↳ dropCliff;
388 
389     uint256 amountDropCliff = (totalAmount * dropCliff) / 1000;
```

```
390     uint256 endRelease = releaseTimestamp + slices * slicePeriod;
391
392     if (block.timestamp >= releaseTimestamp) {
393         uint256 actualSlice = (block.timestamp - releaseTimestamp)
 ↳ /
394             (slicePeriod);
395
396         if (slices <= actualSlice) {
397             amountReleasable = totalAmount;
398         } else {
399             amountReleasable = amountDropCliff + ((totalAmount -
 ↳ amountDropCliff) * actualSlice) / slices;
400         }
401         amountToRelease = amountReleasable - totalAmountReleased;
402     }
403 }
```

Remediation Plan:

SOLVED: The Convergence Finance team fixed the issue by switching to a different non-slice based model in commit 46799c7.

4.8 (HAL-08) VESTING SCHEDULES WITH AN AMOUNT LOWER THAN MAXIMUM SUPPLY REVERT - LOW (2.0)

Description:

When releasing CVG for the team or DAO schedule, the CVG is computed based on the max supply instead of the total amount specified in the schedule. Therefore, specifying any amount below the max supply when creating the schedule results in users unable to release due to underflow. The vesting schedules are set by the contract's owner.

Code Location:

Listing 9: contracts/PresaleVesting/VestingCvg.sol (Lines 325,329)

```
311 function _computeReleaseAmountTeamDao(
312     bool _isTeam
313 )
314     internal
315     view
316     returns (uint256 amountToRelease, uint256 _vestingScheduleId)
317 {
318     uint256 vestingType;
319     uint256 totalAmount;
320     uint256 totalAmountReleased;
321
322     if (_isTeam) {
323         totalAmountReleased = amountReleasedTeam;
324         vestingType = TYPE_TEAM;
325         totalAmount = MAX_SUPPLY_TEAM;
326     } else {
327         totalAmountReleased = amountReleasedDao;
328         vestingType = TYPE_DAO;
329         totalAmount = MAX_SUPPLY_DAO;
330     }
331
332     _vestingScheduleId = vestingIdForType[vestingType];
333     amountToRelease = calculateRelease(
```

```
334             _vestingScheduleId,
335             totalAmount,
336             totalAmountReleased
337         );
338     }
```

BVSS:

AO:S/AC:L/AX:L/C:N/I:N/A:C/D:N/Y:N/R:N/S:U (2.0)

Recommendation:

Use the total amount specified when creating the vesting schedule, or implement a check to prevent a vesting schedule for the team or the DAO with a total amount below max supply from being created.

Remediation Plan:

SOLVED: The Convergence Finance team fixed the issue by allowing to create vesting schedules for the team or DAO with the max amount only in commit 20414f9.

RETESTING

The issue described in this section was brought to Halborn's attention by the Convergence Finance team during the engagement.

5.1 CONVERGENCE01 - USER CAN SEND ALLOWANCE EXCESS TO THE CVGUTILITIES CONTRACT

Description:

The SwapperFactory contract is a utility contract that is meant for the CvgUtilities contract to swap tokens to TOKE through the 1inch protocol with the executeSwapForCvgToke() and executeSwapForTAsset() function. However, this function doesn't check that the msg.sender is the CvgUtilities contract address.

This allows a malicious user to call executeSwapForCvgToke() with the address of a user that has non-utilized allowance for the SwapperFactory and perform a griefing attack by sending the funds from the user to the SwapperFactory.

Code Location:

Listing 10: contracts/utils/SwapperFactory.sol (Line 106)

```
79 /**
80  * @notice Swap source tokens to TOKE through 1inch protocol
81  * @param _user address of the staking user
82  * @param _swapTransaction aggregation data used for the swap to
↳ occur through 1inch protocol
83  */
84 function executeSwapForCvgToke(
85     address _user,
86     IAggregationRouterV5.SwapTransaction calldata _swapTransaction
87 ) external returns (uint256 totalTokeAmount) {
88     ICvgControlTower _cvgControlTower = cvgControlTower;
89
90     require(_swapTransaction.description.amount > 0, "
```

```
↳ INVALID_AMOUNT");
91      require(
92          srcTokenAllowed[_swapTransaction.description.srcToken],
93          "SRC_TOKEN_NOT_ALLOWED"
94      );
95      require(
96          _swapTransaction.description.dstToken == _cvgControlTower.
↳ toke(),
97          "NOT_SWAPPING_TO_TOKE"
98      );
99      require(
100         _swapTransaction.description.dstReceiver ==
101             _cvgControlTower.cvgUtilities(),
102         "INVALID_RECEIVER"
103     );
104
105     /// @dev transfer user's tokens to this contract before
↳ swapping
106     _swapTransaction.description.srcToken.transferFrom(
107         _user,
108         address(this),
109         _swapTransaction.description.amount
110     );
111
112     /// @dev set allowance value to swapped amount
113     IAggregationRouterV5 _aggregationRouter = aggregationRouter;
114     _swapTransaction.description.srcToken.approve(
115         address(_aggregationRouter),
116         _swapTransaction.description.amount
117     );
118
119     /// @dev swap source token to TOKE
120     (totalTokeAmount, ) = _aggregationRouter.swap(
121         _swapTransaction.executor,
122         _swapTransaction.description,
123         _swapTransaction.permit,
124         _swapTransaction.data
125     );
126 }
```

BVSS::

AO:A/AC:L/AX:M/C:N/I:N/A:N/D:C/Y:N/R:N/S:U - (6.7 - Medium)

Recommendation:

Restrict the msg.sender so the executeSwapForCvgToke() and executeSwapForTAsset() functions can only be called by the CvgUtilities contract.

Remediation Plan:

SOLVED: The Convergence Finance team identified this issue and solved it by preventing calling the executeSwapForCvgToke() and executeSwapForTAsset() functions from any other address that is not the CvgUtilities contract. 225258abc90206315302ade5c2d8701cb792cbf3.

RETESTING

AUTOMATED TESTING

6.1 STATIC ANALYSIS REPORT

Description:

Halborn used automated testing techniques to enhance the coverage of certain areas of the smart contracts in scope. Among the tools used was Slither, a Solidity static analysis framework. After Halborn verified the smart contracts in the repository and was able to compile them correctly into their ABIs and binary format, Slither was run against the 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.

Results:

BondCalculator.sol

```
INFO:Detectors:
BondCalculator.computeRoi(uint256,uint256,uint256,uint256,uint256,uint256,uint256,uint256) (contracts/Bond/BondCalculator.sol#162-185) performs a multiplication on the result of a division:
        - intRange = computeNtrDivWtc(timestamp,bondRoundStartingTime,totalPeriodsNumber,composedFunction,maxCvgToMint,cvgMintedOnActualRound) / 250_000 (contracts/Bond/BondCalculator.sol#172-179)
        - 5000 * intRange >= maxRoi (contracts/Bond/BondCalculator.sol#180)
BondCalculator.computeRoi(uint256,uint256,uint256,uint256,uint256,uint256,uint256,uint256) (contracts/Bond/BondCalculator.sol#162-185) performs a multiplication on the result of a division:
        - intRange = computeNtrDivWtc(timestamp,bondRoundStartingTime,totalPeriodsNumber,composedFunction,maxCvgToMint,cvgMintedOnActualRound) / 250_000 (contracts/Bond/BondCalculator.sol#172-179)
        - bondRoi = maxRoi - 5000 * intRange (contracts/Bond/BondCalculator.sol#183)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#divide-before-multiply
INFO:Detectors:
Pragma version^0.8.0 (contracts/Bond/BondCalculator.sol#12) allows old versions
solc-0.8.17 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Slither:./contracts/Bond/BondCalculator.sol analyzed (2 contracts with 85 detectors), 4 result(s) found
```

BondDepository.sol

```
INFO:Detectors:
BondCalculator.computeRoi(uint256,uint256,uint256,uint256,uint256,uint256,uint256,uint256) (contracts/Bond/BondCalculator.sol#162-185) performs a multiplication on the result of a division:
        - intRange = computeNtrDivWtc(timestamp,bondRoundStartingTime,totalPeriodsNumber,composedFunction,maxCvgToMint,cvgMintedOnActualRound) / 250_000 (contracts/Bond/BondCalculator.sol#172-179)
        - 5000 * intRange >= maxRoi (contracts/Bond/BondCalculator.sol#180)
BondCalculator.computeRoi(uint256,uint256,uint256,uint256,uint256,uint256,uint256,uint256) (contracts/Bond/BondCalculator.sol#162-185) performs a multiplication on the result of a division:
        - intRange = computeNtrDivWtc(timestamp,bondRoundStartingTime,totalPeriodsNumber,composedFunction,maxCvgToMint,cvgMintedOnActualRound) / 250_000 (contracts/Bond/BondCalculator.sol#172-179)
        - bondRoi = maxRoi - 5000 * intRange (contracts/Bond/BondCalculator.sol#183)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#divide-before-multiply
INFO:Detectors:
Pragma version^0.8.0 (contracts/Bond/BondCalculator.sol#12) allows old versions
solc-0.8.17 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Slither:./contracts/Bond/BondCalculator.sol analyzed (2 contracts with 85 detectors), 4 result(s) found
```

BondLogo.sol

```
INFO:Detectors:
BondLogo._toString(uint256) (contracts/Bond/BondLogo.sol#28-45) uses a weak PRNG: "buffer[digits] = bytes1(uint8(48 + uint256(value % 10))) (contracts/Bond/BondLogo.sol#41)"
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#weak-PRNG
INFO:Detectors:
BondLogo._toString(uint256) (contracts/Bond/BondLogo.sol#28-45) uses a dangerous strict equality:
        - value == 0 (contracts/Bond/BondLogo.sol#29)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dangerous-strict-equalities
INFO:Detectors:
BondLogo._tokenURI(BondLogo.LogoInfos).hoursLock (contracts/Bond/BondLogo.sol#60) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables
INFO:Detectors:
BondLogo._tokenURI(BondLogo.LogoInfos).output (contracts/Bond/BondLogo.sol#55) is written in both
        output = string(abi.encodePacked(output,<text x="19.9%" y="87%" text-anchor="middle" font-size="25" font-family="andale mono, monospace">,_toString(hoursLock), h</text></svg>)) (contracts/Bond/BondLogo.sol#133-140)
        output = string(abi.encodePacked(data:application/json;base64,,json)) (contracts/Bond/BondLogo.sol#156)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#write-after-write
INFO:Detectors:
BondLogo._toString(uint256) (contracts/Bond/BondLogo.sol#28-45) uses timestamp for comparisons
        Dangerous comparisons:
        - value == 0 (contracts/Bond/BondLogo.sol#29)
        - temp != 0 (contracts/Bond/BondLogo.sol#34)
        - value != 0 (contracts/Bond/BondLogo.sol#39)
BondLogo._tokenURI(BondLogo.LogoInfos) (contracts/Bond/BondLogo.sol#55-157) uses timestamp for comparisons
        Dangerous comparisons:
        - logoInfos.unlockingTimestamp > block.timestamp (contracts/Bond/BondLogo.sol#61)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp
INFO:Detectors:
Pragma version^0.8.0 (contracts/Bond/BondLogo.sol#12) allows old versions
solc-0.8.17 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Function BondLogo._tokenURI(BondLogo.LogoInfos) (contracts/Bond/BondLogo.sol#55-157) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
BondLogo.cvgControlTower (contracts/Bond/BondLogo.sol#20) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
INFO:Slither:./contracts/Bond/BondLogo.sol analyzed (39 contracts with 85 detectors), 10 result(s) found
```

AUTOMATED TESTING

BondPositionManager.sol

```
INFO:Detectors:
CvgERC721TimeLocking.getTokenIdsForWallet(address).i (contracts/Token/CvgERC721TimeLocking.sol#39) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables
INFO:Detectors:
CvgERC721TimeLocking._checkTimeLock(uint256) (contracts/Token/CvgERC721TimeLocking.sol#49-51) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(unlockingTimestampPerToken[tokenId] < block.timestamp,TOKEN_TIMELOCKED) (contracts/Token/CvgERC721TimeLocking.sol#50)
CvgERC721TimeLocking.setLock(uint256,uint256) (contracts/Token/CvgERC721TimeLocking.sol#62-65) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(timestamp > block.timestamp && timestamp - block.timestamp < maxLockingTime,WRONG_TIME_LOCK) (contracts/Token/CvgERC721TimeLocking.sol#63)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp
INFO:Detectors:
CvgERC721TimeLocking._checkTimeLock(uint256) (contracts/Token/CvgERC721TimeLocking.sol#49-51) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
Pragma version^0.8.0 (contracts/Bond/BondPositionManager.sol#12) allows old versions
Pragma version^0.8.0 (contracts/Token/CvgERC721TimeLocking.sol#12) allows old versions
solc-0.8.17 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Parameter BondPositionManager.setLogo(IBondLogo)._logo (contracts/Bond/BondPositionManager.sol#81) is not in mixedCase
Parameter CvgERC721TimeLocking.getTokenIdsForWallet(address)._wallet (contracts/Token/CvgERC721TimeLocking.sol#36) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
BondPositionManager.cvgControlTower (contracts/Bond/BondPositionManager.sol#20) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
INFO:Slither:./contracts/Bond/BondPositionManager.sol analyzed (48 contracts with 85 detectors), 10 result(s) found
```

LockingPositionDelegate.sol

```
INFO:Detectors:
LockingLogo._toString(uint256) (contracts/Locking/LockingLogo.sol#62-79) uses a weak PRNG: "buffer[digits] = bytes1(uint8(48 + uint256(value % 10))) (contracts/Locking/LockingLogo.sol#75)"
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#weak-PRNG
INFO:Detectors:
LockingLogo._toString(uint256) (contracts/Locking/LockingLogo.sol#62-79) uses a dangerous strict equality:
        - value == 0 (contracts/Locking/LockingLogo.sol#63)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dangerous-strict-equalities
INFO:Detectors:
LockingLogo._tokenURI(lockingLogo.LogoInfos).hoursLock (contracts/Locking/LockingLogo.sol#211) is a local variable never initialized
LockingLogo._getClaimableAmountInUsd(uint256,uint256,uint256).claimableUsdAmount (contracts/Locking/LockingLogo.sol#146) is a local variable never initialized
LockingLogo._getClaimableAmountInUsd(uint256,uint256,uint256).claimableDataLength (contracts/Locking/LockingLogo.sol#155) is a local variable never initialized
LockingLogo._getClaimableAmountInUsd(uint256,uint256,uint256).i (contracts/Locking/LockingLogo.sol#169) is a local variable never initialized
LockingLogo.findIndex(address,LockingLogo.ClaimableData[],uint256).i (contracts/Locking/LockingLogo.sol#92) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables
INFO:Detectors:
LockingLogo._tokenURI(lockingLogo.LogoInfos).output (contracts/Locking/LockingLogo.sol#201) is written in both
        output = string(abi.encodePacked(output,<text x="19.96" y="915" text-anchor="middle" font-size="25" font-family="andale mono, monospace">,_toString(hoursLock), h</text></svg>)) (contracts/Locking/LockingLogo.sol#355-362)
        output = string(abi.encodePacked(data:application/json;base64,,json)) (contracts/Locking/LockingLogo.sol#377)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#write-after-write
INFO:Detectors:
LockingLogo._toString(uint256) (contracts/Locking/LockingLogo.sol#62-79) uses timestamp for comparisons
        Dangerous comparisons:
        - value == 0 (contracts/Locking/LockingLogo.sol#63)
        - temp != 0 (contracts/Locking/LockingLogo.sol#68)
        - value != 0 (contracts/Locking/LockingLogo.sol#73)
LockingLogo._tokenURI(lockingLogo.LogoInfos) (contracts/Locking/LockingLogo.sol#201-378) uses timestamp for comparisons
        Dangerous comparisons:
        - _logoInfos.unlockingTimestamp > block.timestamp (contracts/Locking/LockingLogo.sol#212)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp
INFO:Detectors:
Pragma version^0.8.0 (contracts/Locking/LockingLogo.sol#2) allows old versions
solc-0.8.17 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Parameter LockingLogo.findIndex(address,LockingLogo.ClaimableData[],uint256)._token (contracts/Locking/LockingLogo.sol#88) is not in mixedCase
Parameter LockingLogo.findIndex(address,LockingLogo.ClaimableData[],uint256)._claimableData (contracts/Locking/LockingLogo.sol#89) is not in mixedCase
Parameter LockingLogo.findIndex(address,LockingLogo.ClaimableData[],uint256)._arrayLength (contracts/Locking/LockingLogo.sol#90) is not in mixedCase
Parameter LockingLogo.getTokenShare(ICvgControlTower,uint256,uint256,uint256,uint256)._cvgControlTower (contracts/Locking/LockingLogo.sol#115) is not in mixedCase
Parameter LockingLogo.getTokenShare(ICvgControlTower,uint256,uint256,uint256,uint256)._tokenId (contracts/Locking/LockingLogo.sol#116) is not in mixedCase
Parameter LockingLogo.getTokenShare(ICvgControlTower,uint256,uint256,uint256,uint256)._lockEnd (contracts/Locking/LockingLogo.sol#117) is not in mixedCase
Parameter LockingLogo.getTokenShare(ICvgControlTower,uint256,uint256,uint256,uint256)._cvgCycle (contracts/Locking/LockingLogo.sol#118) is not in mixedCase
Parameter LockingLogo.getTokenShare(ICvgControlTower,uint256,uint256,uint256,uint256)._tGId (contracts/Locking/LockingLogo.sol#119) is not in mixedCase
Parameter LockingLogo.getTokenShare(ICvgControlTower,uint256,uint256,uint256,uint256)._tGeDuration (contracts/Locking/LockingLogo.sol#120) is not in mixedCase
Function LockingLogo._tokenURI(LockingLogo.LogoInfos) (contracts/Locking/LockingLogo.sol#201-378) is not in mixedCase
Parameter LockingLogo._tokenURI(LockingLogo.LogoInfos)._logoInfos (contracts/Locking/LockingLogo.sol#201) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
LockingLogo.cvgControlTower (contracts/Locking/LockingLogo.sol#13) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
INFO:Slither:./contracts/Locking/LockingLogo.sol analyzed (41 contracts with 85 detectors), 24 result(s) found
```

LockingPositionManager.sol

```
INFO:Detectors:
LockingPositionManager.mintPosition(uint96,uint256,uint64,bool) (contracts/Locking/LockingPositionManager.sol#280-290) ignores return value by _cvgControlTower.cvgToken().transferFrom(msg.sender,address(this),amount) (contracts/Locking/LockingPositionManager.sol#280-284)
LockingPositionManager.increaseLockAmount(uint256,uint256) (contracts/Locking/LockingPositionManager.sol#308-368) ignores return value by _cvgControlTower.cvgToken().transferFrom(msg.sender,address(this),amount) (contracts/Locking/LockingPositionManager.sol#353-357)
LockingPositionManager.increaseLockTimeAndAmount(uint256,uint256,uint256) (contracts/Locking/LockingPositionManager.sol#438-522) ignores return value by _cvgControlTower.cvgToken().transferFrom(msg.sender,address(this),amount) (contracts/Locking/LockingPositionManager.sol#510-514
)
LockingPositionManager.burnPosition(uint256) (contracts/Locking/LockingPositionManager.sol#532-548) ignores return value by _cvgControlTower.cvgToken().transfer(msg.sender,totalCvgLocked) (contracts/Locking/LockingPositionManager.sol#545)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unchecked-transfer
INFO:Detectors:
LockingPositionManager._ysCvgCheckpoint(uint256,uint256,uint256,uint256) (contracts/Locking/LockingPositionManager.sol#594-636) performs a multiplication on the result of a division:
	- ysTotalAmount = (lockDuration * cvgLockAmount) / MAX_LOCK (contracts/Locking/LockingPositionManager.sol#601)
	- ysNextTdeAmount = ((nextTdeCycle - realStartCycle) * ysTotalAmount) / TDE_DURATION (contracts/Locking/LockingPositionManager.sol#612-613)
LockingPositionManager.balanceOfYsCvgAt(uint256,uint256) (contracts/Locking/LockingPositionManager.sol#680-722) performs a multiplication on the result of a division:
	- _ysTotal = (((_extension.endCycle - _extension.cycleId) * _extension.cvgLocked + _lockingPosition.ysPercentage) / MAX_PERCENTAGE) / MAX_LOCK (contracts/Locking/LockingPositionManager.sol#704-707)
	- _ysPartial = ((_firstTdeCycle - _extension.cycleId) * _ysTotal) / TDE_DURATION (contracts/Locking/LockingPositionManager.sol#708-709)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#divide-before-multiply
INFO:Detectors:
Reentrancy in LockingPositionManager.increaseLockAmount(uint256,uint256) (contracts/Locking/LockingPositionManager.sol#308-368):
	External calls:
	- _cvgControlTower.votingPowerEscrow().increase_amount(tokenId,amountVote / MAX_PERCENTAGE) (contracts/Locking/LockingPositionManager.sol#329-332)
	State variables written after the call(s):
	- lockingPositions[tokenId].mgCvgAmount += _newVotingPower (contracts/Locking/LockingPositionManager.sol#336)
	LockingPositionManager.lockingPositions (contracts/Locking/LockingPositionManager.sol#82) can be used in cross function reentrancies:
	- LockingPositionManager.balanceOfMgCvgAt(uint256,uint256) (contracts/Locking/LockingPositionManager.sol#724-751)
	- LockingPositionManager.balanceOfYsCvgAt(uint256,uint256) (contracts/Locking/LockingPositionManager.sol#680-722)
	- LockingPositionManager.burnPosition(uint256) (contracts/Locking/LockingPositionManager.sol#532-548)
	- LockingPositionManager.increaseLockAmount(uint256,uint256) (contracts/Locking/LockingPositionManager.sol#308-368)
	- LockingPositionManager.increaseLockTime(uint256,uint256) (contracts/Locking/LockingPositionManager.sol#371-427)
	- LockingPositionManager.increaseLockTimeAndAmount(uint256,uint256,uint256) (contracts/Locking/LockingPositionManager.sol#438-522)
	- LockingPositionManager.lockingPositions (contracts/Locking/LockingPositionManager.sol#82)
	- LockingPositionManager.mintPosition(uint96,uint256,uint64,bool) (contracts/Locking/LockingPositionManager.sol#280-290)
	- LockingPositionManager.tokenInfos(uint256) (contracts/Locking/LockingPositionManager.sol#647-674)
	- LockingPositionManager.tokenURI(uint256) (contracts/Locking/LockingPositionManager.sol#147-180)
	- lockingPositions[tokenId].totalCvgLocked += amount (contracts/Locking/LockingPositionManager.sol#340)
	LockingPositionManager.lockingPositions (contracts/Locking/LockingPositionManager.sol#82) can be used in cross function reentrancies:
	- LockingPositionManager.balanceOfMgCvgAt(uint256,uint256) (contracts/Locking/LockingPositionManager.sol#724-751)
	- LockingPositionManager.balanceOfYsCvgAt(uint256,uint256) (contracts/Locking/LockingPositionManager.sol#680-722)
	- LockingPositionManager.burnPosition(uint256) (contracts/Locking/LockingPositionManager.sol#532-548)
	- LockingPositionManager.increaseLockAmount(uint256,uint256) (contracts/Locking/LockingPositionManager.sol#308-368)
	- LockingPositionManager.increaseLockTime(uint256,uint256) (contracts/Locking/LockingPositionManager.sol#371-427)
	- LockingPositionManager.increaseLockTimeAndAmount(uint256,uint256,uint256) (contracts/Locking/LockingPositionManager.sol#438-522)
	- LockingPositionManager.lockingPositions (contracts/Locking/LockingPositionManager.sol#82)
	- LockingPositionManager.mintPosition(uint96,uint256,uint64,bool) (contracts/Locking/LockingPositionManager.sol#280-290)
	- LockingPositionManager.tokenInfos(uint256) (contracts/Locking/LockingPositionManager.sol#647-674)
	- LockingPositionManager.tokenURI(uint256) (contracts/Locking/LockingPositionManager.sol#147-180)
Reentrancy in LockingPositionManager.increaseLockTime(uint256,uint256) (contracts/Locking/LockingPositionManager.sol#371-427):
	External calls:
	- _cvgControlTower.votingPowerEscrow().increase_unlock_time(tokenId,block.timestamp + ((newEndCycle - actualCycle) * 604800)) (contracts/Locking/LockingPositionManager.sol#417-420)
	State variables written after the call(s):
	- lockingPosition.lastEndCycle = uint96(newEndCycle - 1) (contracts/Locking/LockingPositionManager.sol#424)
	LockingPositionManager.lockingPositions (contracts/Locking/LockingPositionManager.sol#82) can be used in cross function reentrancies:
	- LockingPositionManager.balanceOfMgCvgAt(uint256,uint256) (contracts/Locking/LockingPositionManager.sol#724-751)
	- LockingPositionManager.balanceOfYsCvgAt(uint256,uint256) (contracts/Locking/LockingPositionManager.sol#680-722)
	- LockingPositionManager.burnPosition(uint256) (contracts/Locking/LockingPositionManager.sol#532-548)
	- LockingPositionManager.increaseLockAmount(uint256,uint256) (contracts/Locking/LockingPositionManager.sol#308-368)
	- LockingPositionManager.increaseLockTime(uint256,uint256) (contracts/Locking/LockingPositionManager.sol#371-427)
	- LockingPositionManager.increaseLockTimeAndAmount(uint256,uint256,uint256) (contracts/Locking/LockingPositionManager.sol#438-522)
	- LockingPositionManager.lockingPositions (contracts/Locking/LockingPositionManager.sol#82)
	- LockingPositionManager.mintPosition(uint96,uint256,uint64,bool) (contracts/Locking/LockingPositionManager.sol#280-290)
	- LockingPositionManager.tokenInfos(uint256) (contracts/Locking/LockingPositionManager.sol#647-674)
	- LockingPositionManager.tokenURI(uint256) (contracts/Locking/LockingPositionManager.sol#147-180)
Reentrancy in LockingPositionManager.increaseLockTimeAndAmount(uint256,uint256,uint256) (contracts/Locking/LockingPositionManager.sol#438-522):
	External calls:
	- _cvgControlTower.votingPowerEscrow().increase_unlock_time(tokenId,block.timestamp + ((newEndCycle - actualCycle) * 604800)) (contracts/Locking/LockingPositionManager.sol#487-490)
	- _cvgControlTower.votingPowerEscrow().increase_amount(tokenId,amountVote / MAX_PERCENTAGE) (contracts/Locking/LockingPositionManager.sol#491-494)
	State variables written after the call(s):
	- lockExtensions[tokenId].push(_lockingExtension) (contracts/Locking/LockingPositionManager.sol#508)
	LockingPositionManager.lockExtensions (contracts/Locking/LockingPositionManager.sol#94) can be used in cross function reentrancies:
	- LockingPositionManager.balanceOfMgCvgAt(uint256,uint256) (contracts/Locking/LockingPositionManager.sol#724-751)
	- LockingPositionManager.balanceOfYsCvgAt(uint256,uint256) (contracts/Locking/LockingPositionManager.sol#680-722)
	- LockingPositionManager.increaseLockAmount(uint256,uint256) (contracts/Locking/LockingPositionManager.sol#308-368)
	- LockingPositionManager.increaseLockTimeAndAmount(uint256,uint256,uint256) (contracts/Locking/LockingPositionManager.sol#438-522)
	- LockingPositionManager.lockExtensions (contracts/Locking/LockingPositionManager.sol#94)
	- LockingPositionManager.mintPosition(uint96,uint256,uint64,bool) (contracts/Locking/LockingPositionManager.sol#280-290)
	- lockingPosition.lastEndCycle = uint96(newEndCycle - 1) (contracts/Locking/LockingPositionManager.sol#498)
	LockingPositionManager.lockingPositions (contracts/Locking/LockingPositionManager.sol#82) can be used in cross function reentrancies:
	- LockingPositionManager.balanceOfMgCvgAt(uint256,uint256) (contracts/Locking/LockingPositionManager.sol#724-751)
	- LockingPositionManager.balanceOfYsCvgAt(uint256,uint256) (contracts/Locking/LockingPositionManager.sol#680-722)
	- LockingPositionManager.burnPosition(uint256) (contracts/Locking/LockingPositionManager.sol#532-548)
	- LockingPositionManager.increaseLockAmount(uint256,uint256) (contracts/Locking/LockingPositionManager.sol#308-368)
	- LockingPositionManager.increaseLockTime(uint256,uint256) (contracts/Locking/LockingPositionManager.sol#371-427)
	- LockingPositionManager.increaseLockTimeAndAmount(uint256,uint256,uint256) (contracts/Locking/LockingPositionManager.sol#438-522)
	- LockingPositionManager.lockingPositions (contracts/Locking/LockingPositionManager.sol#82)
	- LockingPositionManager.mintPosition(uint96,uint256,uint64,bool) (contracts/Locking/LockingPositionManager.sol#280-290)
	- LockingPositionManager.tokenInfos(uint256) (contracts/Locking/LockingPositionManager.sol#647-674)
	- LockingPositionManager.tokenURI(uint256) (contracts/Locking/LockingPositionManager.sol#147-180)
	- lockingPosition.totalCvgLocked += amount (contracts/Locking/LockingPositionManager.sol#499)
	LockingPositionManager.lockingPositions (contracts/Locking/LockingPositionManager.sol#82) can be used in cross function reentrancies:
	- LockingPositionManager.balanceOfMgCvgAt(uint256,uint256) (contracts/Locking/LockingPositionManager.sol#724-751)
	- LockingPositionManager.balanceOfYsCvgAt(uint256,uint256) (contracts/Locking/LockingPositionManager.sol#680-722)
	- LockingPositionManager.burnPosition(uint256) (contracts/Locking/LockingPositionManager.sol#532-548)
	- LockingPositionManager.increaseLockAmount(uint256,uint256) (contracts/Locking/LockingPositionManager.sol#308-368)
	- LockingPositionManager.increaseLockTime(uint256,uint256) (contracts/Locking/LockingPositionManager.sol#371-427)
	- LockingPositionManager.increaseLockTimeAndAmount(uint256,uint256,uint256) (contracts/Locking/LockingPositionManager.sol#438-522)
	- LockingPositionManager.lockingPositions (contracts/Locking/LockingPositionManager.sol#82)
	- LockingPositionManager.mintPosition(uint96,uint256,uint64,bool) (contracts/Locking/LockingPositionManager.sol#280-290)
	- LockingPositionManager.tokenInfos(uint256) (contracts/Locking/LockingPositionManager.sol#647-674)
```

```
        - LockingPositionManager.tokenURI(uint256) (contracts/Locking/LockingPositionManager.sol#147-180)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-1
INFO:Detectors:
LockingPositionManager.votingPowerPerAddress(address)._i_scope_0 (contracts/Locking/LockingPositionManager.sol#788) is a local variable never initialized
LockingPositionManager.balanceOfVeCvgAt(uint256,uint256).i (contracts/Locking/LockingPositionManager.sol#698) is a local variable never initialized
LockingPositionManager.increaseLockTimeAndAmount(uint256,uint256,uint256) (contracts/Locking/LockingPositionManager.sol#477) is a local variable never initialized
LockingPositionManager.balanceOfMgCvgAt(uint256,uint256).i (contracts/Locking/LockingPositionManager.sol#742) is a local variable never initialized
LockingPositionManager.mgCvgVotingPowerPerAddress(address)._i (contracts/Locking/LockingPositionManager.sol#825) is a local variable never initialized
LockingPositionManager.votingPowerPerAddress(address)._totalVotingPower (contracts/Locking/LockingPositionManager.sol#744) is a local variable never initialized
LockingPositionManager.increaseLockAmount(uint256,uint256)._newVotingPower (contracts/Locking/LockingPositionManager.sol#324) is a local variable never initialized
LockingPositionManager.mgCvgVotingPowerPerAddress(address)._totalVeGovernance (contracts/Locking/LockingPositionManager.sol#814) is a local variable never initialized
LockingPositionManager.mgCvgVotingPowerPerAddress(address)._i_scope_0 (contracts/Locking/LockingPositionManager.sol#849) is a local variable never initialized
CvgERC721TimeLocking.getTokenIdsForWallet(address)._i (contracts/Token/CvgERC721TimeLocking.sol#39) is a local variable never initialized
LockingPositionManager.balanceOfMgCvgAt(uint256)._mgCvgBalance (contracts/Locking/LockingPositionManager.sol#732) is a local variable never initialized
LockingPositionManager.balanceOfVsCvgAt(uint256,uint256)._vsCvgBalance (contracts/Locking/LockingPositionManager.sol#688) is a local variable never initialized
LockingPositionManager.mintPosition(uint96,uint256,uint64,bool)._mgCvgCreated (contracts/Locking/LockingPositionManager.sol#224) is a local variable never initialized
LockingPositionManager.votingPowerPerAddress(address)._i (contracts/Locking/LockingPositionManager.sol#778) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables
INFO:Detectors:
LockingPositionManager.votingPowerPerAddress(address) (contracts/Locking/LockingPositionManager.sol#761-805) has external calls inside a loop: _user == _lockingPositionDelegate.delegatedVeCvg(_tokenId) (contracts/Locking/LockingPositionManager.sol#778)
LockingPositionManager.balanceOfVeCvg(uint256) (contracts/Locking/LockingPositionManager.sol#676-678) has external calls inside a loop: cvgControlTower.votingPowerEscrow().balanceOf(_tokenId) (contracts/Locking/LockingPositionManager.sol#677)
LockingPositionManager.votingPowerPerAddress(address) (contracts/Locking/LockingPositionManager.sol#761-805) has external calls inside a loop: _lockingPositionDelegate.delegatedVeCvg(_tokenId_scope_1) == address(0) && _user == ownerOf(_tokenId_scope_1) (contracts/Locking/LockingPositionManager.sol#792-794)
LockingPositionManager.mgCvgVotingPowerPerAddress(address) (contracts/Locking/LockingPositionManager.sol#811-873) has external calls inside a loop: (_toPercentage,_toIndex) = _lockingPositionDelegate.getMgDelegateeInfoPerTokenAndAddress(_tokenId,_user) (contracts/Locking/LockingPositionManager.sol#827-834)
LockingPositionManager.balanceOfMgCvgAt(uint256) (contracts/Locking/LockingPositionManager.sol#753-753) has external calls inside a loop: balanceOfMgCvgAt(_tokenId,cvgControlTower.cvgCycle()) (contracts/Locking/LockingPositionManager.sol#754)
LockingPositionManager.mgCvgVotingPowerPerAddress(address) (contracts/Locking/LockingPositionManager.sol#811-873) has external calls inside a loop: (_totalPercentageDelegated) = _lockingPositionDelegate.getMgDelegateeInfoPerTokenAndAddress(_tokenId_scope_1,_user) (contracts/Locking/LockingPositionManager.sol#853-858)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#calls-inside-a-loop
INFO:Detectors:
Reentrancy in LockingPositionManager.increaseLockAmount(uint256,uint256) (contracts/Locking/LockingPositionManager.sol#300-340):
        External calls:
        - _cvgControlTower.votingPowerEscrow().increase_amount(tokenId,amountVote / MAX_PERCENTAGE) (contracts/Locking/LockingPositionManager.sol#329-332)
        State variables written after the call(s):
        - lockExtensions[tokenId].push(lockingExtension) (contracts/Locking/LockingPositionManager.sol#338)
Reentrancy in LockingPositionManager.mintPosition(uint96,uint256,uint64,bool) (contracts/Locking/LockingPositionManager.sol#200-290):
        External calls:
        - _cvgControlTower.votingPowerEscrow().create_lock(tokenId,amountVote / MAX_PERCENTAGE,block.timestamp + (lockDuration + 1) * 604800) (contracts/Locking/LockingPositionManager.sol#248-244)
        - _cvgControlTower.lockingPositionDelegate().addTokenAtMint(tokenId,msg.sender) (contracts/Locking/LockingPositionManager.sol#251-254)
        State variables written after the call(s):
        - lockExtensions[tokenId].push(lockingExtension) (contracts/Locking/LockingPositionManager.sol#277)
        - lockingPositions[tokenId] = lockingPosition (contracts/Locking/LockingPositionManager.sol#267)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2
INFO:Detectors:
Reentrancy in LockingPositionManager.burnPosition(uint256) (contracts/Locking/LockingPositionManager.sol#532-548):
        External calls:
        - _cvgControlTower.votingPowerEscrow().withdraw(tokenId) (contracts/Locking/LockingPositionManager.sol#541)
        - _cvgControlTower.cvgToken().transfer(msg.sender,totalLvgLocked) (contracts/Locking/LockingPositionManager.sol#545)
        Event emitted after the call(s):
        - LockingPositionBurn(tokenId) (contracts/Locking/LockingPositionManager.sol#547)
Reentrancy in LockingPositionManager.increaseLockAmount(uint256,uint256) (contracts/Locking/LockingPositionManager.sol#300-340):
        External calls:
        - _cvgControlTower.votingPowerEscrow().increase_amount(tokenId,amountVote / MAX_PERCENTAGE) (contracts/Locking/LockingPositionManager.sol#329-332)
        - _cvgControlTower.cvgToken().transferFrom(msg.sender,address(this),amount) (contracts/Locking/LockingPositionManager.sol#333-357)
        Event emitted after the call(s):
        - IncreaseLockAmount(tokenId,lockingPosition,lockingExtension) (contracts/Locking/LockingPositionManager.sol#359)
Reentrancy in LockingPositionManager.increaseLockTime(uint256,uint256) (contracts/Locking/LockingPositionManager.sol#371-427):
        External calls:
        - _cvgControlTower.votingPowerEscrow().increase_unlock_time(tokenId,block.timestamp + ((newEndCycle - actualCycle) * 604800)) (contracts/Locking/LockingPositionManager.sol#417-420)
        Event emitted after the call(s):
        - IncreaseLockTime(tokenId,lockingPosition,oldEndCycle - 1) (contracts/Locking/LockingPositionManager.sol#426)
Reentrancy in LockingPositionManager.increaseLockTimeAndAmount(uint256,uint256,uint256) (contracts/Locking/LockingPositionManager.sol#438-522):
        External calls:
        - _cvgControlTower.votingPowerEscrow().increase_unlock_time(tokenId,block.timestamp + ((newEndCycle - actualCycle) * 604800)) (contracts/Locking/LockingPositionManager.sol#487-490)
        - _cvgControlTower.votingPowerEscrow().increase_amount(tokenId,amountVote / MAX_PERCENTAGE) (contracts/Locking/LockingPositionManager.sol#491-494)
        - _cvgControlTower.cvgToken().transferFrom(msg.sender,address(this),amount) (contracts/Locking/LockingPositionManager.sol#510-514)
        Event emitted after the call(s):
        - IncreaseLockTimeAndAmount(tokenId,lockingPosition,_lockingExtension,oldEndCycle - 1) (contracts/Locking/LockingPositionManager.sol#516-521)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3
INFO:Detectors:
CvgERC721TimeLocking._checkTimeLock(uint256) (contracts/Token/CvgERC721TimeLocking.sol#49-51) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(unlockingTimestampPerToken[tokenId] < block.timestamp,TOKEN_TIMELOCKED) (contracts/Token/CvgERC721TimeLocking.sol#50)
CvgERC721TimeLocking.setLock(uint256,uint256) (contracts/Token/CvgERC721TimeLocking.sol#62-65) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(timestamp > block.timestamp && timestamp - block.timestamp < maxLockingTime,WRONG_TIME_LOCK) (contracts/Token/CvgERC721TimeLocking.sol#63)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp
INFO:Detectors:
Pragma version^0.8.0 (contracts/Locking/LockingPositionManager.sol#12) allows old versions
Pragma version^0.8.0 (contracts/Token/CvgERC721TimeLocking.sol#12) allows old versions
solc-0.8.17 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Parameter LockingPositionManager.setBaseURI(string)._newBaseURI (contracts/Locking/LockingPositionManager.sol#139) is not in mixedCase
Parameter LockingPositionManager.setLogo(ILockingLogo)._logo (contracts/Locking/LockingPositionManager.sol#143) is not in mixedCase
Parameter LockingPositionManager.tokenInfos(uint256)._tokenId (contracts/Locking/LockingPositionManager.sol#648) is not in mixedCase
Parameter LockingPositionManager.balanceOfVeCvg(uint256)._tokenId (contracts/Locking/LockingPositionManager.sol#676) is not in mixedCase
Parameter LockingPositionManager.balanceOfVeCvgAt(uint256,uint256)._tokenId (contracts/Locking/LockingPositionManager.sol#681) is not in mixedCase
Parameter LockingPositionManager.balanceOfVsCvgAt(uint256,uint256)._cycleId (contracts/Locking/LockingPositionManager.sol#682) is not in mixedCase
Parameter LockingPositionManager.balanceOfMgCvgAt(uint256,uint256)._tokenId (contracts/Locking/LockingPositionManager.sol#725) is not in mixedCase
Parameter LockingPositionManager.balanceOfMgCvgAt(uint256,uint256)._cycleId (contracts/Locking/LockingPositionManager.sol#726) is not in mixedCase
Parameter LockingPositionManager.balanceOfMgCvg(uint256)._tokenId (contracts/Locking/LockingPositionManager.sol#753) is not in mixedCase
Parameter LockingPositionManager.votingPowerPerAddress(address)._user (contracts/Locking/LockingPositionManager.sol#762) is not in mixedCase
Parameter LockingPositionManager.mgCvgVotingPowerPerAddress(address)._user (contracts/Locking/LockingPositionManager.sol#812) is not in mixedCase
Parameter LockingPositionManager.totalSupplyOfVsCvgAt(uint256)._at (contracts/Locking/LockingPositionManager.sol#875) is not in mixedCase
Parameter LockingPositionManager.getTokenRewardAmount(uint256,uint256,IERC20)._token (contracts/Locking/LockingPositionManager.sol#898) is not in mixedCase
Parameter CvgERC721TimeLocking.getTokenIdsForWallet(address)._wallet (contracts/Token/CvgERC721TimeLocking.sol#36) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
Variable LockingPositionManager.balanceOfMgCvgAt(uint256,uint256)._lockingPosition (contracts/Locking/LockingPositionManager.sol#730) is too similar to lockingPositionManager.lockingPositions (contracts/Locking/LockingPositionManager.sol#82)
Variable LockingPositionManager.balanceOfVsCvgAt(uint256,uint256)._lockingPosition (contracts/Locking/LockingPositionManager.sol#686) is too similar to lockingPositionManager.lockingPositions (contracts/Locking/LockingPositionManager.sol#82)
Variable LockingPositionManager.tokenInfos(uint256)._lockingPosition (contracts/Locking/LockingPositionManager.sol#650) is too similar to lockingPositionManager.lockingPositions (contracts/Locking/LockingPositionManager.sol#82)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-too-similar
INFO:Detectors:
LockingPositionManager.cvgControlTower (contracts/Locking/LockingPositionManager.sol#106) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
INFO:Slither:./contracts/Locking/LockingPositionManager.sol analyzed (48 contracts with 85 detectors), 58 result(s) found
```

VveCVGCalculator.sol

```
INFO:Detectors:
VveCvgCalculator.calculateVveCvg(address) (contracts/Locking/VveCVGCalculator.sol#37-91) performs a multiplication on the result of a division:
        - a = (int256(0) - _vveCvgUser) / (end - start) (contracts/Locking/VveCVGCalculator.sol#88)
        - b = _vveCvgUser - (a * start) (contracts/Locking/VveCVGCalculator.sol#89)
VveCvgCalculator.calculateVveCvg(address) (contracts/Locking/VveCVGCalculator.sol#37-91) performs a multiplication on the result of a division:
        - a = (int256(0) - _vveCvgUser) / (end - start) (contracts/Locking/VveCVGCalculator.sol#88)
        - uint256(a * x + b) (contracts/Locking/VveCVGCalculator.sol#90)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#divide-before-multiply
INFO:Detectors:
VveCvgCalculator.calculateVveCvg(address).i (contracts/Locking/VveCVGCalculator.sol#64) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables
INFO:Detectors:
VveCvgCalculator.calculateVveCvg(address) (contracts/Locking/VveCVGCalculator.sol#37-91) has external calls inside a loop: totalCvgUser += _presaleSeed.presaleInfoTokenId[tokenIds[i]].cvgAmount (contracts/Locking/VveCVGCalculator.sol#65)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#calls-inside-a-loop
INFO:Detectors:
VveCvgCalculator.calculateVveCvg(address) (contracts/Locking/VveCVGCalculator.sol#37-91) uses timestamp for comparisons
        Dangerous comparisons:
        - x < start || x >= end (contracts/Locking/VveCVGCalculator.sol#85)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp
INFO:Detectors:
Pragma version^0.8.0 (contracts/Locking/VveCVGCalculator.sol#12) allows old versions
solc-0.8.17 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Parameter VveCvgCalculator.vestingVotingPowerPerAddress(address)._account (contracts/Locking/VveCVGCalculator.sol#99) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
VveCvgCalculator.cvgControlTower (contracts/Locking/VveCVGCalculator.sol#19) should be immutable
VveCvgCalculator.vestingCvg (contracts/Locking/VveCVGCalculator.sol#20) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
INFO:Slither:./contracts/Locking/VveCVGCalculator.sol analyzed (36 contracts with 85 detectors), 10 result(s) found
```

CvgOracle.sol

```
Compilation warnings/errors on ./contracts/Oracles/CvgOracle.sol:
Warning: SPDX license identifier not provided in source file. Before publishing, consider adding a comment containing "SPDX-License-Identifier: <SPDX-License>" to each source file. Use "SPDX-License-Identifier: UNLICENSED" for non-open-source code. Please see https://spdx.org for more information.
--> ./node_modules/@uniswap/v2-core/contracts/interfaces/IUniswapV2Pair.sol


INFO:Detectors:
CvgOracle.getV3Price(address,uint32,bool,bool) (contracts/Oracles/CvgOracle.sol#92-123) performs a multiplication on the result of a division:
        - price = (price * price) / FixedPoint96.Q96 (contracts/Oracles/CvgOracle.sol#118)
        - price = (price * 10 ** getDecimalsDelta(uniswapV3Pool)) / FixedPoint96.Q96 (contracts/Oracles/CvgOracle.sol#119-121)
CvgOracle.postTreatmentAndVerifyEth(uint256,bool,bool) (contracts/Oracles/CvgOracle.sol#169-187) performs a multiplication on the result of a division:
        - price = 10 ** 36 / price (contracts/Oracles/CvgOracle.sol#176)
        - price = (price * ethPrice) / 10 ** 18 (contracts/Oracles/CvgOracle.sol#184)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#divide-before-multiply
INFO:Detectors:
Pragma version^0.8.0 (contracts/Oracles/CvgOracle.sol#12) allows old versions
solc-0.8.17 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Variable CvgOracle.getDecimalsDelta(address).token0Decimals (contracts/Oracles/CvgOracle.sol#207-209) is too similar to CvgOracle.getDecimalsDelta(address).token1Decimals (contracts/Oracles/CvgOracle.sol#210-212)
Variable CvgOracle.getV2Price(address,bool,bool).uniswapV2Pool (contracts/Oracles/CvgOracle.sol#132) is too similar to CvgOracle.getV3Price(address,uint32,bool,bool).uniswapV3Pool (contracts/Oracles/CvgOracle.sol#93)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-too-similar
INFO:Slither:./contracts/Oracles/CvgOracle.sol analyzed (50 contracts with 85 detectors), 6 result(s) found
```

CvgV3Aggregator.sol

```
INFO:Detectors:
Pragma version^0.8.0 (contracts/Oracles/CvgV3Aggregator.sol#12) allows old versions
solc-0.8.17 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Parameter CvgV3Aggregator.initialize(ICvgControlTower,uint8,string)._cvgControlTower (contracts/Oracles/CvgV3Aggregator.sol#36) is not in mixedCase
Parameter CvgV3Aggregator.setLatestPrice(int256)._newPrice (contracts/Oracles/CvgV3Aggregator.sol#47) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Slither:./contracts/Oracles/CvgV3Aggregator.sol analyzed (40 contracts with 85 detectors), 4 result(s) found
```

SeedPresaleCvg.sol

```
INFO:Detectors:
SeedPresaleCvg.investMint(bool) (contracts/PresaleVesting/SeedPresaleCvg.sol#174-205) ignores return value by token.transferFrom(msg.sender,address(this),tokenAmount) (contracts/PresaleVesting/SeedPresaleCvg.sol#204)
SeedPresaleCvg.withdrawFunds() (contracts/PresaleVesting/SeedPresaleCvg.sol#279-291) ignores return value by Dai.transfer(msg.sender,balanceDai) (contracts/PresaleVesting/SeedPresaleCvg.sol#284)
SeedPresaleCvg.withdrawFunds() (contracts/PresaleVesting/SeedPresaleCvg.sol#279-291) ignores return value by Frax.transfer(msg.sender,balanceFrax) (contracts/PresaleVesting/SeedPresaleCvg.sol#285)
SeedPresaleCvg.withdrawFunds() (contracts/PresaleVesting/SeedPresaleCvg.sol#279-291) ignores return value by Dai.transfer(msg.sender,balanceDai) (contracts/PresaleVesting/SeedPresaleCvg.sol#287)
SeedPresaleCvg.withdrawFunds() (contracts/PresaleVesting/SeedPresaleCvg.sol#279-291) ignores return value by Frax.transfer(msg.sender,balanceFrax) (contracts/PresaleVesting/SeedPresaleCvg.sol#289)
SeedPresaleCvg.withdrawToken(IERC20) (contracts/PresaleVesting/SeedPresaleCvg.sol#293-297) ignores return value by _token.transfer(msg.sender,balance) (contracts/PresaleVesting/SeedPresaleCvg.sol#296)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unchecked-transfer
INFO:Detectors:
Pragma version^0.8.17 (contracts/PresaleVesting/SeedPresaleCvg.sol#12) allows old versions
solc-0.8.17 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Parameter SeedPresaleCvg.getTokenIdAndType(address,uint256)._wallet (contracts/PresaleVesting/SeedPresaleCvg.sol#102) is not in mixedCase
Parameter SeedPresaleCvg.getTokenIdAndType(address,uint256)._index (contracts/PresaleVesting/SeedPresaleCvg.sol#103) is not in mixedCase
Parameter SeedPresaleCvg.getTokenIdsForWallet(address)._wallet (contracts/PresaleVesting/SeedPresaleCvg.sol#109) is not in mixedCase
Parameter SeedPresaleCvg.getAllocation(address)._wallet (contracts/PresaleVesting/SeedPresaleCvg.sol#130) is not in mixedCase
Parameter SeedPresaleCvg.setSaleState(SeedPresaleCvg.SaleState)._saleState (contracts/PresaleVesting/SeedPresaleCvg.sol#144) is not in mixedCase
Parameter SeedPresaleCvg.grantPreseed(address,uint256)._wallet (contracts/PresaleVesting/SeedPresaleCvg.sol#149) is not in mixedCase
Parameter SeedPresaleCvg.grantPreseed(address,uint256)._amount (contracts/PresaleVesting/SeedPresaleCvg.sol#149) is not in mixedCase
Parameter SeedPresaleCvg.grantSeed(address,uint256)._wallet (contracts/PresaleVesting/SeedPresaleCvg.sol#155) is not in mixedCase
Parameter SeedPresaleCvg.grantSeed(address,uint256)._amount (contracts/PresaleVesting/SeedPresaleCvg.sol#155) is not in mixedCase
Parameter SeedPresaleCvg.investMint(bool)._isDai (contracts/PresaleVesting/SeedPresaleCvg.sol#174) is not in mixedCase
Parameter SeedPresaleCvg.withdrawToken(IERC20)._token (contracts/PresaleVesting/SeedPresaleCvg.sol#293) is not in mixedCase
Variable SeedPresaleCvg.Dai (contracts/PresaleVesting/SeedPresaleCvg.sol#76) is not in mixedCase
Variable SeedPresaleCvg.Frax (contracts/PresaleVesting/SeedPresaleCvg.sol#77) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Slither:./contracts/PresaleVesting/SeedPresaleCvg.sol analyzed (17 contracts with 85 detectors), 21 result(s) found
```

WlPresaleCvg.sol

```
INFO:Detectors:
VestingCvg.releaseSeed(uint256) (contracts/PresaleVesting/VestingCvg.sol#234-254) ignores return value by cvg.transfer(msg.sender,amountToRelease) (contracts/PresaleVesting/VestingCvg.sol#253)
VestingCvg.releaseWl(uint256) (contracts/PresaleVesting/VestingCvg.sol#260-279) ignores return value by cvg.transfer(msg.sender,amountToRelease) (contracts/PresaleVesting/VestingCvg.sol#278)
VestingCvg.releaseTeamOrDao(bool) (contracts/PresaleVesting/VestingCvg.sol#281-304) ignores return value by cvg.transfer(msg.sender,amountToRelease) (contracts/PresaleVesting/VestingCvg.sol#303)
VestingCvg.withdrawOnlyExcess(uint256) (contracts/PresaleVesting/VestingCvg.sol#409-416) ignores return value by cvg.transfer(msg.sender,_amount) (contracts/PresaleVesting/VestingCvg.sol#415)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unchecked-transfer
INFO:Detectors:
VestingCvg.calculateRrelease(uint256,uint256,uint256) (contracts/PresaleVesting/VestingCvg.sol#376-404) performs a multiplication on the result of a division:
        - actualSlice = (block.timestamp - releaseTimestamp) / (slicePeriod) (contracts/PresaleVesting/VestingCvg.sol#391-392)
        - amountReleasable = amountDropCliff + ((totalAmount - amountDropCliff) * actualSlice) / slices (contracts/PresaleVesting/VestingCvg.sol#397-400)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#divide-before-multiply
INFO:Detectors:
VestingCvg.createVestingSchedule(uint256,uint256,uint256,uint256,uint256,uint256,uint256) (contracts/PresaleVesting/VestingCvg.sol#171-212) should emit an event for:
        - vestingSchedulesTotalAmount += _totalAmount (contracts/PresaleVesting/VestingCvg.sol#195)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-events-arithmetic
INFO:Detectors:
VestingCvg.setWhitelistTeam(address).newWhitelistedTeam (contracts/PresaleVesting/VestingCvg.sol#114) lacks a zero-check on :
        - whitelistedTeam = newWhitelistedTeam (contracts/PresaleVesting/VestingCvg.sol#115)
VestingCvg.setWhitelistDao(address).newWhitelistedDao (contracts/PresaleVesting/VestingCvg.sol#118) lacks a zero-check on :
        - whitelistedDao = newWhitelistedDao (contracts/PresaleVesting/VestingCvg.sol#119)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation
INFO:Detectors:
VestingCvg.calculateRrelease(uint256,uint256,uint256) (contracts/PresaleVesting/VestingCvg.sol#376-404) uses timestamp for comparisons
        Dangerous comparisons:
        - block.timestamp >= releaseTimestamp (contracts/PresaleVesting/VestingCvg.sol#390)
        - slices <= actualSlice (contracts/PresaleVesting/VestingCvg.sol#394)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp
INFO:Detectors:
Pragma version^0.8.0 (contracts/PresaleVesting/VestingCvg.sol#12) allows old versions
solc-0.8.17 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Parameter VestingCvg.setCvg(IERC20)._cvg (contracts/PresaleVesting/VestingCvg.sol#108) is not in mixedCase
Parameter VestingCvg.setPresale(IPresaleCvgWl)._newPresaleWl (contracts/PresaleVesting/VestingCvg.sol#104) is not in mixedCase
Parameter VestingCvg.setPresaleSeed(IPresaleCvgSeed)._newPresaleSeed (contracts/PresaleVesting/VestingCvg.sol#109) is not in mixedCase
Parameter VestingCvg.getTotalReleasedScheduleId(uint256)._vestingScheduleId (contracts/PresaleVesting/VestingCvg.sol#130) is not in mixedCase
Parameter VestingCvg.getInfoVestingTokenId(uint256,bool)._tokenId (contracts/PresaleVesting/VestingCvg.sol#136) is not in mixedCase
Parameter VestingCvg.getInfoVestingTokenId(uint256,bool)._isSeed (contracts/PresaleVesting/VestingCvg.sol#137) is not in mixedCase
Parameter VestingCvg.createVestingSchedule(uint256,uint256,uint256,uint256,uint256,uint256,uint256)._totalAmount (contracts/PresaleVesting/VestingCvg.sol#172) is not in mixedCase
Parameter VestingCvg.createVestingSchedule(uint256,uint256,uint256,uint256,uint256,uint256,uint256)._start (contracts/PresaleVesting/VestingCvg.sol#173) is not in mixedCase
Parameter VestingCvg.createVestingSchedule(uint256,uint256,uint256,uint256,uint256,uint256,uint256)._cliff (contracts/PresaleVesting/VestingCvg.sol#174) is not in mixedCase
Parameter VestingCvg.createVestingSchedule(uint256,uint256,uint256,uint256,uint256,uint256,uint256)._slices (contracts/PresaleVesting/VestingCvg.sol#175) is not in mixedCase
Parameter VestingCvg.createVestingSchedule(uint256,uint256,uint256,uint256,uint256,uint256,uint256)._slicePeriods (contracts/PresaleVesting/VestingCvg.sol#176) is not in mixedCase
Parameter VestingCvg.createVestingSchedule(uint256,uint256,uint256,uint256,uint256,uint256,uint256)._vestingType (contracts/PresaleVesting/VestingCvg.sol#177) is not in mixedCase
Parameter VestingCvg.createVestingSchedule(uint256,uint256,uint256,uint256,uint256,uint256,uint256)._dropCliff (contracts/PresaleVesting/VestingCvg.sol#178) is not in mixedCase
Parameter VestingCvg.revokeVestingSchedule(uint256)._vestingScheduleId (contracts/PresaleVesting/VestingCvg.sol#219) is not in mixedCase
Parameter VestingCvg.releaseSeed(uint256)._tokenId (contracts/PresaleVesting/VestingCvg.sol#236) is not in mixedCase
Parameter VestingCvg.releaseWl(uint256)._tokenId (contracts/PresaleVesting/VestingCvg.sol#262) is not in mixedCase
Parameter VestingCvg.releaseTeamOrDao(bool)._isTeam (contracts/PresaleVesting/VestingCvg.sol#281) is not in mixedCase
Parameter VestingCvg.withdrawOnlyExcess(uint256)._amount (contracts/PresaleVesting/VestingCvg.sol#409) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
Variable VestingCvg._computeReleaseAmount(uint256,bool)._vestingScheduleId (contracts/PresaleVesting/VestingCvg.sol#348) is too similar to VestingCvg.calculateRelease(uint256,uint256,uint256).vestingSchedulesId (contracts/PresaleVesting/VestingCvg.sol#377)
Variable VestingCvg.getTotalReleasedScheduleId(uint256)._vestingScheduleId (contracts/PresaleVesting/VestingCvg.sol#130) is too similar to VestingCvg.calculateRelease(uint256,uint256,uint256).vestingSchedulesId (contracts/PresaleVesting/VestingCvg.sol#377)
Variable VestingCvg._computeReleaseAmountTeamDao(bool)._vestingScheduleId (contracts/PresaleVesting/VestingCvg.sol#314) is too similar to VestingCvg.calculateRelease(uint256,uint256,uint256).vestingSchedulesId (contracts/PresaleVesting/VestingCvg.sol#377)
Variable VestingCvg.revokeVestingSchedule(uint256)._vestingScheduleId (contracts/PresaleVesting/VestingCvg.sol#219) is too similar to VestingCvg.calculateRelease(uint256,uint256,uint256).vestingSchedulesId (contracts/PresaleVesting/VestingCvg.sol#377)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-too-similar
INFO:Slither:./contracts/PresaleVesting/VestingCvg.sol analyzed (42 contracts with 85 detectors), 33 result(s) found
```

CvgRewards.sol

```
INFO:Detectors:
Reentrancy in CvgRewards._checkpoints() (contracts/Rewards/CvgRewards.sol#144-199):
        External calls:
        - _gaugeController.set_lock(true) (contracts/Rewards/CvgRewards.sol#163)
        State variables written after the call(s):
        - cursor = 0 (contracts/Rewards/CvgRewards.sol#176)
        CvgRewards.cursor (contracts/Rewards/CvgRewards.sol#58) can be used in cross function reentrancies:
        - CvgRewards._checkpoints() (contracts/Rewards/CvgRewards.sol#144-199)
        - CvgRewards._distributeCvgRewards() (contracts/Rewards/CvgRewards.sol#250-328)
        - CvgRewards._setTotalWeight() (contracts/Rewards/CvgRewards.sol#201-244)
        - CvgRewards.cursor (contracts/Rewards/CvgRewards.sol#58)
        - cursor = _endChunk (contracts/Rewards/CvgRewards.sol#181)
        CvgRewards.cursor (contracts/Rewards/CvgRewards.sol#58) can be used in cross function reentrancies:
        - CvgRewards._checkpoints() (contracts/Rewards/CvgRewards.sol#144-199)
        - CvgRewards._distributeCvgRewards() (contracts/Rewards/CvgRewards.sol#250-328)
        - CvgRewards._setTotalWeight() (contracts/Rewards/CvgRewards.sol#201-244)
        - CvgRewards.cursor (contracts/Rewards/CvgRewards.sol#58)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-1
INFO:Detectors:
CvgRewards.stakingInflationAtCycle(uint256).index (contracts/Rewards/CvgRewards.sol#113) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables
INFO:Detectors:
Reentrancy in CvgRewards._checkpoints() (contracts/Rewards/CvgRewards.sol#144-199):
        External calls:
        - _gaugeController.set_lock(true) (contracts/Rewards/CvgRewards.sol#163)
        State variables written after the call(s):
        - state = State.LOCK_TOTAL_WEIGHT (contracts/Rewards/CvgRewards.sol#178)
Reentrancy in CvgRewards._distributeCvgRewards() (contracts/Rewards/CvgRewards.sol#250-328):
        External calls:
        - gaugeController.set_lock(false) (contracts/Rewards/CvgRewards.sol#280)
        State variables written after the call(s):
        - state = State.CHECKPOINT (contracts/Rewards/CvgRewards.sol#282)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2
INFO:Detectors:
Reentrancy in CvgRewards._checkpoints() (contracts/Rewards/CvgRewards.sol#144-199):
        External calls:
        - _gaugeController.set_lock(true) (contracts/Rewards/CvgRewards.sol#163)
        - _gaugeController.gauge_relative_weight_write(_gaugeController.gauges(_cursor)) (contracts/Rewards/CvgRewards.sol#187-189)
        Event emitted after the call(s):
        - Checkpoints(_cvgRewardsCycle) (contracts/Rewards/CvgRewards.sol#197)
Reentrancy in CvgRewards._distributeCvgRewards() (contracts/Rewards/CvgRewards.sol#250-328):
        External calls:
        - gaugeController.set_lock(false) (contracts/Rewards/CvgRewards.sol#280)
        - ICvgAssetStaking(gaugeAddress).processStakersRewards(cvgDistributed) (contracts/Rewards/CvgRewards.sol#308-310)
        Event emitted after the call(s):
        - EventChunkWriteStakingRewards(_cvgRewardsCycle,_totalWeight,inflationInfos) (contracts/Rewards/CvgRewards.sol#323-327)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3
INFO:Detectors:
CvgRewards._checkpoints() (contracts/Rewards/CvgRewards.sol#144-199) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(lastUpdatedTimestamp + 604800 <= block.timestamp,NEED_WAIT_7_DAYS) (contracts/Rewards/CvgRewards.sol#145-148)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp
INFO:Detectors:
Pragma version^0.8.0 (contracts/Rewards/CvgRewards.sol#12) allows old versions
solc-0.8.17 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
CvgRewards.slitherConstructorConstantVariables() (contracts/Rewards/CvgRewards.sol#19-329) uses literals with too many digits:
        - INITIAL_CYCLE_INFLATION = 2083846150000000000000000 (contracts/Rewards/CvgRewards.sol#71)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits
INFO:Detectors:
CvgRewards.cvgControlTower (contracts/Rewards/CvgRewards.sol#52) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
INFO:Slither:./contracts/Rewards/CvgRewards.sol analyzed (39 contracts with 85 detectors), 11 result(s) found
```

TAssetBlackHole.sol

```
INFO:Detectors:
TAssetBlackHole._distributeRewards() (contracts/Rewards/TAssetBlackHole.sol#240-302) ignores return value by _toke.transfer(gaugeAddress,tokeDistributed) (contracts/Rewards/TAssetBlackHole.sol#294)
TAssetBlackHole.withdraw(IERC20,address,uint256) (contracts/Rewards/TAssetBlackHole.sol#312-315) ignores return value by tAsset.transfer(receiver,amount) (contracts/Rewards/TAssetBlackHole.sol#314)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unchecked-transfer
INFO:Detectors:
Reentrancy in TAssetBlackHole._checkpoints() (contracts/Rewards/TAssetBlackHole.sol#153-196):
        External calls:
        - _gaugeController.set_lock(true) (contracts/Rewards/TAssetBlackHole.sol#163)
        State variables written after the call(s):
        - cursor = 0 (contracts/Rewards/TAssetBlackHole.sol#175)
        TAssetBlackHole.cursor (contracts/Rewards/TAssetBlackHole.sol#53) can be used in cross function reentrancies:
        - TAssetBlackHole._checkpoints() (contracts/Rewards/TAssetBlackHole.sol#153-196)
        - TAssetBlackHole._distributeRewards() (contracts/Rewards/TAssetBlackHole.sol#240-302)
        - TAssetBlackHole._setTotalWeight() (contracts/Rewards/TAssetBlackHole.sol#198-238)
        - TAssetBlackHole.cursor (contracts/Rewards/TAssetBlackHole.sol#53)
        - cursor = _endChunk (contracts/Rewards/TAssetBlackHole.sol#180)
        TAssetBlackHole.cursor (contracts/Rewards/TAssetBlackHole.sol#53) can be used in cross function reentrancies:
        - TAssetBlackHole._checkpoints() (contracts/Rewards/TAssetBlackHole.sol#153-196)
        - TAssetBlackHole._distributeRewards() (contracts/Rewards/TAssetBlackHole.sol#240-302)
        - TAssetBlackHole._setTotalWeight() (contracts/Rewards/TAssetBlackHole.sol#198-238)
        - TAssetBlackHole.cursor (contracts/Rewards/TAssetBlackHole.sol#53)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-1
INFO:Detectors:
TAssetBlackHole.processTokeRewardsEmergency(address[],uint256,uint256).i (contracts/Rewards/TAssetBlackHole.sol#92) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables
INFO:Detectors:
TAssetBlackHole.processTokeRewardsEmergency(address[],uint256,uint256) (contracts/Rewards/TAssetBlackHole.sol#87-95) has external calls inside a loop: ITAssetStaking(gauges[i]).processTokeRewards(amount,_tokeCycle) (contracts/Rewards/TAssetBlackHole.sol#93)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#calls-inside-a-loop
INFO:Detectors:
Reentrancy in TAssetBlackHole._checkpoints() (contracts/Rewards/TAssetBlackHole.sol#153-196):
        External calls:
        - _gaugeController.set_lock(true) (contracts/Rewards/TAssetBlackHole.sol#163)
        State variables written after the call(s):
        - state = State.LOCK_TOTAL_WEIGHT (contracts/Rewards/TAssetBlackHole.sol#177)
        - tokeBalanceLocked = toke.balanceOf(address(this)) (contracts/Rewards/TAssetBlackHole.sol#164)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2
INFO:Detectors:
Reentrancy in TAssetBlackHole._checkpoints() (contracts/Rewards/TAssetBlackHole.sol#153-196):
        External calls:
        - _gaugeController.set_lock(true) (contracts/Rewards/TAssetBlackHole.sol#163)
        - _gaugeController.gauge_relative_weight_write(_cvgControlTower.getTAssetStakingContract(_cursor)) (contracts/Rewards/TAssetBlackHole.sol#186)
        Event emitted after the call(s):
        - Checkpoints(tokeCycle) (contracts/Rewards/TAssetBlackHole.sol#194)
Reentrancy in TAssetBlackHole._distributeRewards() (contracts/Rewards/TAssetBlackHole.sol#240-302):
        External calls:
        - _gaugeController.set_lock(false) (contracts/Rewards/TAssetBlackHole.sol#264)
        - _toke.transfer(gaugeAddress,tokeDistributed) (contracts/Rewards/TAssetBlackHole.sol#294)
        - ITAssetStaking(gaugeAddress).processTokeRewards(tokeDistributed,_tokeCycle) (contracts/Rewards/TAssetBlackHole.sol#296)
        Event emitted after the call(s):
        - EventChunkWriteTAssetStakingRewards(_tokeCycle,distributionInfos) (contracts/Rewards/TAssetBlackHole.sol#301)
Reentrancy in TAssetBlackHole.claimTokeRewards(ITokeRewards.Recipient,uint8,bytes32,bytes32) (contracts/Rewards/TAssetBlackHole.sol#112-129):
        External calls:
        - cvgControlTower.tokeRewards().claim(recipient,v,r,s) (contracts/Rewards/TAssetBlackHole.sol#126)
        Event emitted after the call(s):
        - BlackHoleClaimTokeRewards(recipient.cycle,recipient.amount) (contracts/Rewards/TAssetBlackHole.sol#128)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3
INFO:Detectors:
Pragma version^0.8.0 (contracts/Rewards/TAssetBlackHole.sol#12) allows old versions
solc-0.8.17 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Parameter TAssetBlackHole.processTokeRewardsEmergency(address[],uint256,uint256)._tokeCycle (contracts/Rewards/TAssetBlackHole.sol#90) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Slither:./contracts/Rewards/TAssetBlackHole.sol analyzed (38 contracts with 85 detectors), 12 result(s) found
```

YsDistributor.sol

```
INFO:Detectors:
YsDistributor.depositMultipleToken(YsDistributor.TokenAmount[]) (contracts/Rewards/YsDistributor.sol#94-136) ignores return value by _token.transferFrom(msg.sender,address(this),_amount) (contracts/Rewards/YsDistributor.sol#129)
YsDistributor._claimTokenRewards(uint256,uint256,uint256,address,IAggregationRouterV5.SwapTransaction[],IERC20) (contracts/Rewards/YsDistributor.sol#204-257) ignores return value by _token.transfer(address(_swapperFactory),_amountUser) (contracts/Rewards/YsDistributor.sol#242)
YsDistributor._claimTokenRewards(uint256,uint256,uint256,address,IAggregationRouterV5.SwapTransaction[],IERC20) (contracts/Rewards/YsDistributor.sol#204-257) ignores return value by _token.transfer(receiver,_amountUser) (contracts/Rewards/YsDistributor.sol#245)
YsDistributor._claimTokenRewards(uint256,uint256,uint256,address,IAggregationRouterV5.SwapTransaction[],IERC20) (contracts/Rewards/YsDistributor.sol#204-257) ignores return value by _token.transfer(receiver,_amountUser) (contracts/Rewards/YsDistributor.sol#248)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unchecked-transfer
INFO:Detectors:
Reentrancy in YsDistributor.claimRewards(uint256,uint256,address,address,IAggregationRouterV5.SwapTransaction[],IERC20) (contracts/Rewards/YsDistributor.sol#154-192):
        External calls:
        - _claimTokenRewards(tokenId,tdeId,share,receiver,_swapTransactions,destinationToken) (contracts/Rewards/YsDistributor.sol#188)
            - _token.transfer(address(_swapperFactory),_amountUser) (contracts/Rewards/YsDistributor.sol#242)
            - _swapperFactory.swap(_swapTransaction) (contracts/Rewards/YsDistributor.sol#243)
            - _token.transfer(receiver,_amountUser) (contracts/Rewards/YsDistributor.sol#245)
            - _token.transfer(receiver,_amountUser) (contracts/Rewards/YsDistributor.sol#248)
        State variables written after the call(s):
        - rewardsClaimedForToken[tokenId][tdeId] = true (contracts/Rewards/YsDistributor.sol#191)
        YsDistributor.rewardsClaimedForToken (contracts/Rewards/YsDistributor.sol#53) can be used in cross function reentrancies:
        - YsDistributor.claimRewards(uint256,uint256,address,address,IAggregationRouterV5.SwapTransaction[],IERC20) (contracts/Rewards/YsDistributor.sol#154-192)
        - YsDistributor.getAllTokenRewardsForTde(uint256[],uint256) (contracts/Rewards/YsDistributor.sol#303-346)
        - YsDistributor.rewardsClaimedForToken (contracts/Rewards/YsDistributor.sol#53)
Reentrancy in YsDistributor.depositMultipleToken(YsDistributor.TokenAmount[]) (contracts/Rewards/YsDistributor.sol#94-136):
        External calls:
        - _token.transferFrom(msg.sender,address(this),_amount) (contracts/Rewards/YsDistributor.sol#129)
        State variables written after the call(s):
        - depositedTokenAddressForTde[_actualTDE].push(address(_token)) (contracts/Rewards/YsDistributor.sol#125)
        YsDistributor.depositedTokenAddressForTde (contracts/Rewards/YsDistributor.sol#59) can be used in cross function reentrancies:
        - YsDistributor._claimTokenRewards(uint256,uint256,uint256,address,IAggregationRouterV5.SwapTransaction[],IERC20) (contracts/Rewards/YsDistributor.sol#204-257)
        - YsDistributor._tokenRewardsForOneTde(uint256,uint256,ILockingPositionManager) (contracts/Rewards/YsDistributor.sol#348-369)
        - YsDistributor.depositMultipleToken(YsDistributor.TokenAmount[]) (contracts/Rewards/YsDistributor.sol#94-136)
        - YsDistributor.getTokensDepositedAtTde(uint256) (contracts/Rewards/YsDistributor.sol#371-373)
        - depositedTokenAmountForTde[_actualTDE][_token] += _amount (contracts/Rewards/YsDistributor.sol#107)
        YsDistributor.depositedTokenAmountForTde (contracts/Rewards/YsDistributor.sol#56) can be used in cross function reentrancies:
        - YsDistributor._calculateUserRewardAmount(uint256,IERC20,uint256) (contracts/Rewards/YsDistributor.sol#265-268)
        - YsDistributor.depositMultipleToken(YsDistributor.TokenAmount[]) (contracts/Rewards/YsDistributor.sol#94-136)
        - YsDistributor.depositedTokenAmountForTde (contracts/Rewards/YsDistributor.sol#56)
        - YsDistributor.getTokenRewardAmountForTde(IERC20,uint256,uint256) (contracts/Rewards/YsDistributor.sol#381-384)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-1
INFO:Detectors:
YsDistributor._findSwapTransaction(IAggregationRouterV5.SwapTransaction[],IERC20)._swapTransaction (contracts/Rewards/YsDistributor.sol#279) is a local variable never initialized
YsDistributor._findSwapTransaction(IAggregationRouterV5.SwapTransaction[],IERC20).i (contracts/Rewards/YsDistributor.sol#280) is a local variable never initialized
YsDistributor._claimTokenRewards(uint256,uint256,uint256,address,IAggregationRouterV5.SwapTransaction[],IERC20).i (contracts/Rewards/YsDistributor.sol#220) is a local variable never initialized
YsDistributor.depositMultipleToken(YsDistributor.TokenAmount[]).i (contracts/Rewards/YsDistributor.sol#103) is a local variable never initialized
YsDistributor.depositMultipleToken(YsDistributor.TokenAmount[]).flag (contracts/Rewards/YsDistributor.sol#113) is a local variable never initialized
YsDistributor._tokenRewardsForOneTde(uint256,uint256,ILockingPositionManager).i (contracts/Rewards/YsDistributor.sol#358) is a local variable never initialized
YsDistributor.depositMultipleToken(YsDistributor.TokenAmount[]).j (contracts/Rewards/YsDistributor.sol#114) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables
INFO:Detectors:
YsDistributor.depositMultipleToken(YsDistributor.TokenAmount[]) (contracts/Rewards/YsDistributor.sol#94-136) has external calls inside a loop: _token.transferFrom(msg.sender,address(this),_amount) (contracts/Rewards/YsDistributor.sol#129)
YsDistributor.getAllTokenRewardsForTde(uint256[],uint256) (contracts/Rewards/YsDistributor.sol#303-346) has external calls inside a loop: actualCycle <= cycleTde || rewardsClaimedForToken[_tokenId][_tdeIds[i]] || _lockingPositionManager.balanceOfYsCvpAt(_tokenId,cycleTde) == 0 (contracts/Rewards/YsDistributor.sol#318-320)
YsDistributor._tokenRewardsForOneTde(uint256,uint256,ILockingPositionManager) (contracts/Rewards/YsDistributor.sol#348-369) has external calls inside a loop: share = (_lockingPositionManager.balanceOfYsCvpAt(_tokenId,cycleClaimed) * 10 ** 20) / _lockingPositionManager.totalSupply YsCvpHistories[cycleClaimed] (contracts/Rewards/YsDistributor.sol#356-357)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#calls-inside-a-loop
INFO:Detectors:
Reentrancy in YsDistributor._claimTokenRewards(uint256,uint256,uint256,address,IAggregationRouterV5.SwapTransaction[],IERC20) (contracts/Rewards/YsDistributor.sol#204-257):
        External calls:
        - _token.transfer(address(_swapperFactory),_amountUser) (contracts/Rewards/YsDistributor.sol#242)
        - _swapperFactory.swap(_swapTransaction) (contracts/Rewards/YsDistributor.sol#243)
        - _token.transfer(receiver,_amountUser) (contracts/Rewards/YsDistributor.sol#245)
        - _token.transfer(receiver,_amountUser) (contracts/Rewards/YsDistributor.sol#248)
        Event emitted after the call(s):
        - TokensClaim(tokenId,tdeId,share,tokensClaimable) (contracts/Rewards/YsDistributor.sol#256)
Reentrancy in YsDistributor.depositMultipleToken(YsDistributor.TokenAmount[]) (contracts/Rewards/YsDistributor.sol#94-136):
        External calls:
        - _token.transferFrom(msg.sender,address(this),_amount) (contracts/Rewards/YsDistributor.sol#129)
        Event emitted after the call(s):
        - DepositedTokens(_actualCycle,deposits) (contracts/Rewards/YsDistributor.sol#135)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3
INFO:Detectors:
YsDistributor.claimRewards(uint256,uint256,address,address,IAggregationRouterV5.SwapTransaction[],IERC20) (contracts/Rewards/YsDistributor.sol#154-192) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(lockingPositionManager.unlockingTimestampPerToken(tokenId) < block.timestamp,TOKEN_TIMELOCKED) (contracts/Rewards/YsDistributor.sol#164)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp
INFO:Detectors:
YsDistributor.getAllTokenRewardsForTde(uint256[],uint256) (contracts/Rewards/YsDistributor.sol#303-346) uses assembly
        - INLINE ASM (contracts/Rewards/YsDistributor.sol#323-326)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
INFO:Detectors:
Pragma version^0.8.0 (contracts/Rewards/YsDistributor.sol#2) allows old versions
solc-0.8.17 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Parameter YsDistributor.claimRewards(uint256,uint256,address,address,IAggregationRouterV5.SwapTransaction[],IERC20)._swapTransactions (contracts/Rewards/YsDistributor.sol#159) is not in mixedCase
Parameter YsDistributor.getAllTokenRewardsForTde(uint256[],uint256)._tdeIds (contracts/Rewards/YsDistributor.sol#304) is not in mixedCase
Parameter YsDistributor.getAllTokenRewardsForTde(uint256[],uint256)._tokenId (contracts/Rewards/YsDistributor.sol#305) is not in mixedCase
Parameter YsDistributor.getTokensDepositedAtTde(uint256)._tdeId (contracts/Rewards/YsDistributor.sol#371) is not in mixedCase
Parameter YsDistributor.getTokenRewardAmountForTde(IERC20,uint256,uint256)._token (contracts/Rewards/YsDistributor.sol#381) is not in mixedCase
Parameter YsDistributor.getTokenRewardAmountForTde(IERC20,uint256,uint256)._tdeId (contracts/Rewards/YsDistributor.sol#381) is not in mixedCase
Parameter YsDistributor.getTokenRewardAmountForTde(IERC20,uint256,uint256)._share (contracts/Rewards/YsDistributor.sol#381) is not in mixedCase
Parameter YsDistributor.toggleSwapDestinationToken(IERC20)._token (contracts/Rewards/YsDistributor.sol#390) is not in mixedCase
Parameter YsDistributor.setSwapperFactory(ISwapperFactory)._swapperFactory (contracts/Rewards/YsDistributor.sol#405) is not in mixedCase
Parameter YsDistributor.setTokenOracleParams(address,IOracleStruct.OracleParams)._token (contracts/Rewards/YsDistributor.sol#413) is not in mixedCase
Parameter YsDistributor.setTokenOracleParams(address,IOracleStruct.OracleParams)._oracleParams (contracts/Rewards/YsDistributor.sol#416) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
YsDistributor.cvgControlTower (contracts/Rewards/YsDistributor.sol#44) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
```

CvgTokeStaking.sol

```
INFO:Detectors:
CvgTokeStaking.deposit(uint256,uint256,address) (contracts/Staking/CvgTokeStaking.sol#151-194) ignores return value by stakingAsset.transferFrom(msg.sender,address(this),amount) (contracts/Staking/CvgTokeStaking.sol#180)
CvgTokeStaking.withdraw(uint256,uint256) (contracts/Staking/CvgTokeStaking.sol#203-229) ignores return value by stakingAsset.transfer(msg.sender,amount) (contracts/Staking/CvgTokeStaking.sol#228)
TokeStakingCommon._transferOrSwap(uint256,IERC20,address,bool,bool) (contracts/Staking/TokeStakingCommon.sol#607-621) ignores return value by asset.transfer(receiver,amount) (contracts/Staking/TokeStakingCommon.sol#619)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unchecked-transfer
INFO:Detectors:
CvgTokeStaking.addAsset(IERC20Metadata).i (contracts/Staking/CvgTokeStaking.sol#574) is a local variable never initialized
CvgERC721TimeLocking.getTokenIdsForWallet(address).i (contracts/Token/CvgERC721TimeLocking.sol#393) is a local variable never initialized
CvgTokeStaking.constructor(ICvgControlTower,IERC20Metadata[]).i (contracts/Staking/CvgTokeStaking.sol#62) is a local variable never initialized
TokeStakingCommon.getAllClaimableCvgAmount(uint256,uint256,uint256).amountClaimable (contracts/Staking/TokeStakingCommon.sol#711) is a local variable never initialized
CvgTokeStaking.claimMultipleTokeRewards(uint256[],address,bool,bool).i_scope_0 (contracts/Staking/CvgTokeStaking.sol#385) is a local variable never initialized
CvgTokeStaking.claimTokeRewards(uint256,uint256,bool,bool).i (contracts/Staking/CvgTokeStaking.sol#271) is a local variable never initialized
CvgTokeStaking.claimMultipleTokeRewards(uint256,uint256[],address,bool,bool).i (contracts/Staking/CvgTokeStaking.sol#331) is a local variable never initialized
CvgTokeStaking.processTokeRewards(uint256[],uint256).i (contracts/Staking/CvgTokeStaking.sol#551) is a local variable never initialized
TokeStakingCommon.claimMultipleCvgRewards(uint256,uint256[],address).i (contracts/Staking/TokeStakingCommon.sol#229) is a local variable never initialized
CvgTokeStaking.getAllClaimableTokeAmounts(uint256,uint256,uint256).x (contracts/Staking/CvgTokeStaking.sol#431) is a local variable never initialized
CvgTokeStaking.claimMultipleTokeRewards(uint256,uint256[],address,bool,bool).j (contracts/Staking/CvgTokeStaking.sol#360) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables
INFO:Detectors:
TokeStakingCommon._swapToke(uint256,address,bool) (contracts/Staking/TokeStakingCommon.sol#624-649) ignores return value by toke.approve(address(_cvgToke),_amount) (contracts/Staking/TokeStakingCommon.sol#632)
TokeStakingCommon._swapToke(uint256,address,bool) (contracts/Staking/TokeStakingCommon.sol#624-649) ignores return value by toke.approve(address(_poolCvgToke),_amount) (contracts/Staking/TokeStakingCommon.sol#638)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return
INFO:Detectors:
TokeStakingCommon.setMaxViewCycle(uint256) (contracts/Staking/TokeStakingCommon.sol#169-171) should emit an event for:
        - maxViewCycle = _maxViewCycle (contracts/Staking/TokeStakingCommon.sol#170)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-events-arithmetic
INFO:Detectors:
TokeStakingCommon._swapToke(uint256,address,bool) (contracts/Staking/TokeStakingCommon.sol#624-649) has external calls inside a loop: _cvgToke = cvgControlTower.cvgToke() (contracts/Staking/TokeStakingCommon.sol#630)
TokeStakingCommon._swapToke(uint256,address,bool) (contracts/Staking/TokeStakingCommon.sol#624-649) has external calls inside a loop: toke.approve(address(_cvgToke),_amount) (contracts/Staking/TokeStakingCommon.sol#632)
TokeStakingCommon._swapToke(uint256,address,bool) (contracts/Staking/TokeStakingCommon.sol#624-649) has external calls inside a loop: _cvgToke.mint(address(_receiver),_amount) (contracts/Staking/TokeStakingCommon.sol#634)
TokeStakingCommon._swapToke(uint256,address,bool) (contracts/Staking/TokeStakingCommon.sol#624-649) has external calls inside a loop: toke.approve(address(_poolCvgToke),_amount) (contracts/Staking/TokeStakingCommon.sol#638)
TokeStakingCommon._swapToke(uint256,address,bool) (contracts/Staking/TokeStakingCommon.sol#624-649) has external calls inside a loop: _poolCvgToke.exchange(0,1,_amount,_poolCvgToke.get_dy(0,1,_amount),_receiver) (contracts/Staking/TokeStakingCommon.sol#641-647)
TokeStakingCommon._transferOrSwap(uint256,IERC20,address,bool,bool) (contracts/Staking/TokeStakingCommon.sol#607-621) has external calls inside a loop: asset.transfer(receiver,amount) (contracts/Staking/TokeStakingCommon.sol#619)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#calls-inside-a-loop
INFO:Detectors:
Reentrancy in TokeStakingCommon.claimCvgRewards(uint256,uint256) (contracts/Staking/TokeStakingCommon.sol#181-207):
        External calls:
        - cvgControlTower.cvgToken().mintStaking(msg.sender,_amount) (contracts/Staking/TokeStakingCommon.sol#204)
        Event emitted after the call(s):
        - ClaimCvg(_tokenId,msg.sender,_cycleId,_amount) (contracts/Staking/TokeStakingCommon.sol#206)
Reentrancy in TokeStakingCommon.claimMultipleCvgRewards(uint256,uint256[],address) (contracts/Staking/TokeStakingCommon.sol#215-261):
        External calls:
        - cvgControlTower.cvgToken().mintStaking(receiver,_totalAmount) (contracts/Staking/TokeStakingCommon.sol#258)
        Event emitted after the call(s):
        - ClaimCvgMultiple(_tokenId,receiver,multipleClaims) (contracts/Staking/TokeStakingCommon.sol#260)
Reentrancy in CvgTokeStaking.claimMultipleTokeRewards(uint256,uint256[],address,bool,bool) (contracts/Staking/CvgTokeStaking.sol#311-407):
        External calls:
        - _transferOrSwap(totalAmounts[i_scope_0],_assets[i_scope_0],operator,_isConvert,_isMint) (contracts/Staking/CvgTokeStaking.sol#387-393)
                - asset.transfer(receiver,amount) (contracts/Staking/TokeStakingCommon.sol#619)
                - toke.approve(address(_cvgToke),_amount) (contracts/Staking/TokeStakingCommon.sol#632)
                - _cvgToke.mint(address(_receiver),_amount) (contracts/Staking/TokeStakingCommon.sol#634)
                - toke.approve(address(_poolCvgToke),_amount) (contracts/Staking/TokeStakingCommon.sol#638)
                - _poolCvgToke.exchange(0,1,_amount,_poolCvgToke.get_dy(0,1,_amount),_receiver) (contracts/Staking/TokeStakingCommon.sol#641-647)
        Event emitted after the call(s):
        - CvgTokeClaimTokeMultiple(_tokenId,operator,_isConvert,_isMint,multipleClaims) (contracts/Staking/CvgTokeStaking.sol#400-406)
Reentrancy in CvgTokeStaking.claimTokeRewards(uint256,uint256,bool,bool) (contracts/Staking/CvgTokeStaking.sol#240-308):
        External calls:
        - _transferOrSwap(_rewardAmount,_assets[i],msg.sender,_isConvert,_isMint) (contracts/Staking/CvgTokeStaking.sol#279-285)
                - asset.transfer(receiver,amount) (contracts/Staking/TokeStakingCommon.sol#619)
                - toke.approve(address(_cvgToke),_amount) (contracts/Staking/TokeStakingCommon.sol#632)
                - _cvgToke.mint(address(_receiver),_amount) (contracts/Staking/TokeStakingCommon.sol#634)
                - toke.approve(address(_poolCvgToke),_amount) (contracts/Staking/TokeStakingCommon.sol#638)
                - _poolCvgToke.exchange(0,1,_amount,_poolCvgToke.get_dy(0,1,_amount),_receiver) (contracts/Staking/TokeStakingCommon.sol#641-647)
        Event emitted after the call(s):
        - CvgTokeClaimToke(_tokenId,msg.sender,_isConvert,_isMint,TokeClaim(_cycleId,_tokeRewards)) (contracts/Staking/CvgTokeStaking.sol#293-299)
Reentrancy in CvgTokeStaking.withdraw(uint256,uint256) (contracts/Staking/CvgTokeStaking.sol#203-229):
        External calls:
        - stakingAsset.transfer(msg.sender,amount) (contracts/Staking/CvgTokeStaking.sol#228)
        Event emitted after the call(s):
        - Withdraw(tokenId,msg.sender,_cvgStakingCycle,_tokeStakingCycle,amount) (contracts/Staking/CvgTokeStaking.sol#222-228)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3
INFO:Detectors:
CvgERC721TimeLocking._checkTimeLock(uint256) (contracts/Token/CvgERC721TimeLocking.sol#49-51) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(unlockingTimestampPerToken[tokenId] < block.timestamp,TOKEN_TIMELOCKED) (contracts/Token/CvgERC721TimeLocking.sol#50)
CvgERC721TimeLocking.setLock(uint256,uint256) (contracts/Token/CvgERC721TimeLocking.sol#62-65) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(timestamp > block.timestamp && timestamp - block.timestamp < maxLockingTime,WRONG_TIME_LOCK) (contracts/Token/CvgERC721TimeLocking.sol#63)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp
INFO:Detectors:
CvgTokeStaking.getClaimableTokeCyclesAndAmounts(uint256,uint256,uint256) (contracts/Staking/CvgTokeStaking.sol#448-522) uses assembly
        - INLINE ASM (contracts/Staking/CvgTokeStaking.sol#497-503)
        - INLINE ASM (contracts/Staking/CvgTokeStaking.sol#509-515)
TokeStakingCommon.getClaimableCvgCyclesAndAmounts(uint256,uint256,uint256) (contracts/Staking/TokeStakingCommon.sol#732-780) uses assembly
        - INLINE ASM (contracts/Staking/TokeStakingCommon.sol#767-773)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
INFO:Detectors:
Pragma version^0.8.0 (contracts/Staking/CvgTokeStaking.sol#12) allows old versions
Pragma version^0.8.0 (contracts/Staking/TokeStakingCommon.sol#12) allows old versions
Pragma version^0.8.0 (contracts/Token/CvgERC721TimeLocking.sol#12) allows old versions
solc-0.8.17 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Parameter CvgTokeStaking.claimTokeRewards(uint256,uint256,bool,bool)._tokenId (contracts/Staking/CvgTokeStaking.sol#241) is not in mixedCase
Parameter CvgTokeStaking.claimTokeRewards(uint256,uint256,bool,bool)._cycleId (contracts/Staking/CvgTokeStaking.sol#242) is not in mixedCase
Parameter CvgTokeStaking.claimTokeRewards(uint256,uint256,bool,bool)._isConvert (contracts/Staking/CvgTokeStaking.sol#243) is not in mixedCase
Parameter CvgTokeStaking.claimTokeRewards(uint256,uint256,bool,bool)._isMint (contracts/Staking/CvgTokeStaking.sol#244) is not in mixedCase
Parameter CvgTokeStaking.claimMultipleTokeRewards(uint256,uint256[],address,bool,bool)._tokenId (contracts/Staking/CvgTokeStaking.sol#312) is not in mixedCase
Parameter CvgTokeStaking.claimMultipleTokeRewards(uint256,uint256[],address,bool,bool)._cycleIds (contracts/Staking/CvgTokeStaking.sol#313) is not in mixedCase
Parameter CvgTokeStaking.claimMultipleTokeRewards(uint256,uint256[],address,bool,bool)._isConvert (contracts/Staking/CvgTokeStaking.sol#315) is not in mixedCase
Parameter CvgTokeStaking.claimMultipleTokeRewards(uint256,uint256[],address,bool,bool)._isMint (contracts/Staking/CvgTokeStaking.sol#316) is not in mixedCase
Parameter CvgTokeStaking.getAllClaimableTokeAmounts(uint256,uint256,uint256)._tokenId (contracts/Staking/CvgTokeStaking.sol#410) is not in mixedCase
Parameter CvgTokeStaking.getClaimableTokeCyclesAndAmounts(uint256,uint256,uint256)._tokenId (contracts/Staking/CvgTokeStaking.sol#449) is not in mixedCase
Parameter TokeStakingCommon.setMaxViewCycle(uint256)._maxViewCycle (contracts/Staking/TokeStakingCommon.sol#169) is not in mixedCase
Parameter TokeStakingCommon.claimCvgRewards(uint256,uint256)._tokenId (contracts/Staking/TokeStakingCommon.sol#182) is not in mixedCase
Parameter TokeStakingCommon.claimCvgRewards(uint256,uint256)._cycleId (contracts/Staking/TokeStakingCommon.sol#183) is not in mixedCase
Parameter TokeStakingCommon.claimMultipleCvgRewards(uint256,uint256[],address)._tokenId (contracts/Staking/TokeStakingCommon.sol#216) is not in mixedCase
Parameter TokeStakingCommon.claimMultipleCvgRewards(uint256,uint256[],address)._cycleIds (contracts/Staking/TokeStakingCommon.sol#217) is not in mixedCase
Parameter TokeStakingCommon.getRewardsForTokeCycle(uint256)._cycleId (contracts/Staking/TokeStakingCommon.sol#293) is not in mixedCase
Parameter TokeStakingCommon.findTokenStakedAmountForCvgCycle(uint256,uint256)._tokenId (contracts/Staking/TokeStakingCommon.sol#304) is not in mixedCase
Parameter TokeStakingCommon.findTokenStakedAmountForCvgCycle(uint256,uint256)._cycleId (contracts/Staking/TokeStakingCommon.sol#305) is not in mixedCase
Parameter TokeStakingCommon.findTokenStakedAmountForTokeCycle(uint256,uint256)._tokenId (contracts/Staking/TokeStakingCommon.sol#330) is not in mixedCase
Parameter TokeStakingCommon.findTokenStakedAmountForTokeCycle(uint256,uint256)._tokeCycleId (contracts/Staking/TokeStakingCommon.sol#331) is not in mixedCase
Parameter TokeStakingCommon.tokenTotalStaked(uint256)._tokenId (contracts/Staking/TokeStakingCommon.sol#356) is not in mixedCase
Parameter TokeStakingCommon.burn(uint256)._tokenId (contracts/Staking/TokeStakingCommon.sol#654) is not in mixedCase
Parameter TokeStakingCommon.getTokenInfoForCvgCycle(uint256,uint256)._tokenId (contracts/Staking/TokeStakingCommon.sol#680) is not in mixedCase
Parameter TokeStakingCommon.getTokenInfoForCvgCycle(uint256,uint256)._cycleId (contracts/Staking/TokeStakingCommon.sol#681) is not in mixedCase
Parameter TokeStakingCommon.getTokenInfoForTokeCycle(uint256,uint256)._tokenId (contracts/Staking/TokeStakingCommon.sol#691) is not in mixedCase
Parameter TokeStakingCommon.getTokenInfoForTokeCycle(uint256,uint256)._cycleId (contracts/Staking/TokeStakingCommon.sol#692) is not in mixedCase
Parameter TokeStakingCommon.getAllClaimableCvgAmount(uint256,uint256,uint256)._tokenId (contracts/Staking/TokeStakingCommon.sol#701) is not in mixedCase
Parameter TokeStakingCommon.getClaimableCvgCyclesAndAmounts(uint256,uint256,uint256)._tokenId (contracts/Staking/TokeStakingCommon.sol#713) is not in mixedCase
Parameter TokeStakingCommon.setPoolCvgToke(ICrvPoolPlain)._poolCvgToke (contracts/Staking/TokeStakingCommon.sol#794) is not in mixedCase
Parameter CvgERC721TimeLocking.getTokenIdsForWallet(address)._wallet (contracts/Token/CvgERC721TimeLocking.sol#36) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
Variable TokeStakingCommon.claimMultipleCvgRewards(uint256,uint256[],address)._totalAmount (contracts/Staking/TokeStakingCommon.sol#227) is too similar to CvgTokeStaking.claimMultipleTokeRewards(uint256,uint256[],address,bool,bool).totalAmounts (contracts/Staking/CvgTokeStaking.s
ol#327)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-too-similar
INFO:Detectors:
TokeStakingCommon.cvgControlTower (contracts/Staking/TokeStakingCommon.sol#106) should be immutable
TokeStakingCommon.stakingAsset (contracts/Staking/TokeStakingCommon.sol#109) should be immutable
TokeStakingCommon.toke (contracts/Staking/TokeStakingCommon.sol#115) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
INFO:Slither:./contracts/Staking/CvgTokeStaking.sol analyzed (49 contracts with 85 detectors), 75 result(s) found
```

StakingLogo.sol

```
INFO:Detectors:
StakingLogo._toString(uint256) (contracts/Staking/StakingLogo.sol#33-58) uses a weak PRNG: "buffer[digits] = bytes1(uint8(48 + uint256(value % 10))) (contracts/Staking/StakingLogo.sol#46)"
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#weak-PRNG
INFO:Detectors:
StakingLogo._toString(uint256) (contracts/Staking/StakingLogo.sol#33-58) uses a dangerous strict equality:
        - value == 0 (contracts/Staking/StakingLogo.sol#34)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dangerous-strict-equalities
INFO:Detectors:
StakingLogo._tokenURI(StakingLogo.LogoInfos).hoursLock (contracts/Staking/StakingLogo.sol#88) is a local variable never initialized
StakingLogo.setLogosStaking(string[],string[]).i (contracts/Staking/StakingLogo.sol#53) is a local variable never initialized
StakingLogo._tokenURI(StakingLogo.LogoInfos).claimableUnUsd (contracts/Staking/StakingLogo.sol#88) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables
INFO:Detectors:
StakingLogo._tokenURI(StakingLogo.LogoInfos).output (contracts/Staking/StakingLogo.sol#77) is written in both
        output = string(abi.encodePacked(output,<text x="19.9%" y="87%" text-anchor="middle" font-size="25" font-family="andale mono, monospace">,_toString(hoursLock), h</text></svg>)) (contracts/Staking/StakingLogo.sol#149-156)
        output = string(abi.encodePacked(data:application/json;base64,,json)) (contracts/Staking/StakingLogo.sol#172)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#write-after-write
INFO:Detectors:
StakingLogo._toString(uint256) (contracts/Staking/StakingLogo.sol#33-58) uses timestamp for comparisons
        Dangerous comparisons:
        - value == 0 (contracts/Staking/StakingLogo.sol#34)
        - temp != 0 (contracts/Staking/StakingLogo.sol#39)
        - value != 0 (contracts/Staking/StakingLogo.sol#46)
StakingLogo._tokenURI(StakingLogo.LogoInfos) (contracts/Staking/StakingLogo.sol#77-173) uses timestamp for comparisons
        Dangerous comparisons:
        - logoInfos.unlockingTimestamp > block.timestamp (contracts/Staking/StakingLogo.sol#89)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp
INFO:Detectors:
Pragma version^0.8.0 (contracts/Staking/StakingLogo.sol#12) allows old versions
solc-0.8.17 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Parameter StakingLogo.setLogosStaking(string[],string[])._symbols (contracts/Staking/StakingLogo.sol#52) is not in mixedCase
Parameter StakingLogo.setLogosStaking(string[],string[])._logos (contracts/Staking/StakingLogo.sol#52) is not in mixedCase
Parameter StakingLogo.setTokeParams(IOracleStruct.OracleParams)._tokeParams (contracts/Staking/StakingLogo.sol#63) is not in mixedCase
Function StakingLogo._tokenURI(StakingLogo.LogoInfos) (contracts/Staking/StakingLogo.sol#77-173) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
StakingLogo.cvgControlTower (contracts/Staking/StakingLogo.sol#28) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
INFO:Slither:./contracts/Staking/StakingLogo.sol analyzed (41 contracts with 85 detectors), 15 result(s) found
```

StakingViewer.sol

```
INFO:Detectors:
StakingViewer.getAprDatasCvg(uint256,address[]) (contracts/Staking/StakingViewer.sol#73-85) has external calls inside a loop: amounts[i] = ITokeStakingCommon(tAssetContracts[i]).cvgCycleInfo(cvgCycle) (contracts/Staking/StakingViewer.sol#81)
StakingViewer.getAprDatasToke(uint256,address[]) (contracts/Staking/StakingViewer.sol#97-110) has external calls inside a loop: totalStaked = ITokeStakingCommon(tAssetContracts[i]).tokeCycleInfo(tokeCycle).totalStaked (contracts/Staking/StakingViewer.sol#103)
StakingViewer.getAprDatasToke(uint256,address[]) (contracts/Staking/StakingViewer.sol#97-110) has external calls inside a loop: rewards = ITokeStakingCommon(tAssetContracts[i]).getRewardsTokeCycle() (contracts/Staking/StakingViewer.sol#104)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#calls-inside-a-loop
INFO:Detectors:
Pragma version^0.8.0 (contracts/Staking/StakingViewer.sol#14) allows old versions
solc-0.8.17 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Parameter StakingViewer.getGlobalViewTokeStaking(ITokeStakingCommon)._stakingContract (contracts/Staking/StakingViewer.sol#18) is not in mixedCase
Parameter StakingViewer.getTokenViewTokeStaking(ITokeStakingCommon,uint256)._stakingContract (contracts/Staking/StakingViewer.sol#43) is not in mixedCase
Parameter StakingViewer.getTokenViewTokeStaking(ITokeStakingCommon,uint256)._tokenId (contracts/Staking/StakingViewer.sol#44) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Slither:./contracts/Staking/StakingViewer.sol analyzed (36 contracts with 85 detectors), 8 result(s) found
```

TAssetStaking.sol

```
INFO:Detectors:
TAssetStaking.deposit(uint256,uint256,address) (contracts/Staking/TAssetStaking.sol#127-151) ignores return value by stakingAsset.transferFrom(msg.sender,address(cvgControlTower.tAssetBlackHole()),amount) (contracts/Staking/TAssetStaking.sol#144)
TokeStakingCommon._transferOrSwap(uint256,IERC20,address,bool,bool) (contracts/Staking/TokeStakingCommon.sol#607-621) ignores return value by asset.transfer(receiver,amount) (contracts/Staking/TokeStakingCommon.sol#619)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unchecked-transfer
INFO:Detectors:
TokeStakingCommon.claimMultipleCvgRewards(uint256,uint256[],address).i (contracts/Staking/TokeStakingCommon.sol#229) is a local variable never initialized
TAssetStaking.getAllClaimableTokeAmount(uint256,uint256,uint256).amountClaimable (contracts/Staking/TAssetStaking.sol#279) is a local variable never initialized
TAssetStaking.claimMultipleTokeRewards(uint256,uint256[],address,bool,bool).i (contracts/Staking/TAssetStaking.sol#246) is a local variable never initialized
CvgERC721TimeLocking.getTokenIdsForWallet(address).i (contracts/Token/CvgERC721TimeLocking.sol#39) is a local variable never initialized
TokeStakingCommon.getAllClaimableCvgAmount(uint256,uint256,uint256).amountClaimable (contracts/Staking/TokeStakingCommon.sol#711) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables
INFO:Detectors:
TokeStakingCommon._swapToke(uint256,address,bool) (contracts/Staking/TokeStakingCommon.sol#624-649) ignores return value by toke.approve(address(_cvgToke),_amount) (contracts/Staking/TokeStakingCommon.sol#632)
TokeStakingCommon._swapToke(uint256,address,bool) (contracts/Staking/TokeStakingCommon.sol#624-649) ignores return value by toke.approve(address(_poolCvgToke),_amount) (contracts/Staking/TokeStakingCommon.sol#638)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return
INFO:Detectors:
TokeStakingCommon.setMaxViewCycle(uint256) (contracts/Staking/TokeStakingCommon.sol#169-171) should emit an event for:
        - maxViewCycle = _maxViewCycle (contracts/Staking/TokeStakingCommon.sol#170)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-events-arithmetic
INFO:Detectors:
Reentrancy in TokeStakingCommon.claimCvgRewards(uint256,uint256) (contracts/Staking/TokeStakingCommon.sol#181-207):
        External calls:
        - cvgControlTower.cvgToken().mintStaking(msg.sender,_amount) (contracts/Staking/TokeStakingCommon.sol#204)
        Event emitted after the call(s):
        - ClaimCvg(_tokenId,msg.sender,_cycleId,_amount) (contracts/Staking/TokeStakingCommon.sol#206)
Reentrancy in TokeStakingCommon.claimMultipleCvgRewards(uint256,uint256[],address) (contracts/Staking/TokeStakingCommon.sol#215-261):
        External calls:
        - cvgControlTower.cvgToken().mintStaking(receiver,_totalAmount) (contracts/Staking/TokeStakingCommon.sol#258)
        Event emitted after the call(s):
        - ClaimCvgMultiple(_tokenId,receiver,multipleClaims) (contracts/Staking/TokeStakingCommon.sol#260)
Reentrancy in TAssetStaking.claimMultipleTokeRewards(uint256,uint256[],address,bool,bool) (contracts/Staking/TAssetStaking.sol#227-266):
        External calls:
        - _transferOrSwap(_totalAmount,_toke,operator,_isConvert,_isMint) (contracts/Staking/TAssetStaking.sol#263)
                - toke.approve(address(_cvgToke),_amount) (contracts/Staking/TokeStakingCommon.sol#632)
                - asset.transfer(receiver,amount) (contracts/Staking/TokeStakingCommon.sol#619)
                - _cvgToke.mint(address(_receiver),_amount) (contracts/Staking/TokeStakingCommon.sol#634)
                - toke.approve(address(_poolCvgToke),_amount) (contracts/Staking/TokeStakingCommon.sol#638)
                - _poolCvgToke.exchange(0,1,_amount,_poolCvgToke.get_dy(0,1,_amount),_receiver) (contracts/Staking/TokeStakingCommon.sol#641-647)
        Event emitted after the call(s):
        - TAssetClaimTokeMultiple(_tokenId,operator,_isConvert,_isMint,_toke,multipleClaims) (contracts/Staking/TAssetStaking.sol#265)
Reentrancy in TAssetStaking.claimTokeRewards(uint256,uint256,bool,bool) (contracts/Staking/TAssetStaking.sol#184-216):
        External calls:
        - _transferOrSwap(_amount,_toke,msg.sender,_isConvert,_isMint) (contracts/Staking/TAssetStaking.sol#206)
                - toke.approve(address(_cvgToke),_amount) (contracts/Staking/TokeStakingCommon.sol#632)
                - asset.transfer(receiver,amount) (contracts/Staking/TokeStakingCommon.sol#619)
                - _cvgToke.mint(address(_receiver),_amount) (contracts/Staking/TokeStakingCommon.sol#634)
                - toke.approve(address(_poolCvgToke),_amount) (contracts/Staking/TokeStakingCommon.sol#638)
                - _poolCvgToke.exchange(0,1,_amount,_poolCvgToke.get_dy(0,1,_amount),_receiver) (contracts/Staking/TokeStakingCommon.sol#641-647)
        Event emitted after the call(s):
        - TAssetClaimToke(_tokenId,msg.sender,_isConvert,_isMint,_toke,TokeClaim(_cycleId,_amount)) (contracts/Staking/TAssetStaking.sol#208-215)
Reentrancy in TAssetStaking.withdraw(uint256,uint256) (contracts/Staking/TAssetStaking.sol#158-173):
        External calls:
        - cvgControlTower.tAssetBlackHole().withdraw(stakingAsset,msg.sender,amount) (contracts/Staking/TAssetStaking.sol#170)
        Event emitted after the call(s):
        - Withdraw(tokenId,msg.sender,_cvgStakingCycle,_tokeStakingCycle,amount) (contracts/Staking/TAssetStaking.sol#172)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3
INFO:Detectors:
CvgERC721TimeLocking._checkTimeLock(uint256) (contracts/Token/CvgERC721TimeLocking.sol#49-51) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(unlockingTimestampPerToken[tokenId] < block.timestamp,TOKEN_TIMELOCKED) (contracts/Token/CvgERC721TimeLocking.sol#50)
CvgERC721TimeLocking.setLock(uint256,uint256) (contracts/Token/CvgERC721TimeLocking.sol#62-65) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(timestamp > block.timestamp && timestamp - block.timestamp < maxLockingTime,WRONG_TIME_LOCK) (contracts/Token/CvgERC721TimeLocking.sol#63)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp
INFO:Detectors:
TAssetStaking.getClaimableTokeCyclesAndAmounts(uint256,uint256,uint256) (contracts/Staking/TAssetStaking.sol#292-332) uses assembly
        - INLINE ASM (contracts/Staking/TAssetStaking.sol#322-325)
TokeStakingCommon.getClaimableCvgCyclesAndAmounts(uint256,uint256,uint256) (contracts/Staking/TokeStakingCommon.sol#732-780) uses assembly
        - INLINE ASM (contracts/Staking/TokeStakingCommon.sol#767-773)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
INFO:Detectors:
Pragma version^0.8.0 (contracts/Staking/TAssetStaking.sol#12) allows old versions
Pragma version^0.8.0 (contracts/Staking/TokeStakingCommon.sol#12) allows old versions
Pragma version^0.8.0 (contracts/Token/CvgERC721TimeLocking.sol#12) allows old versions
solc-0.8.17 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Parameter TAssetStaking.initialize(ICvgControlTower,IERC20Metadata,string,string)._cvgControlTower (contracts/Staking/TAssetStaking.sol#57) is not in mixedCase
Parameter TAssetStaking.initialize(ICvgControlTower,IERC20Metadata,string,string)._tAsset (contracts/Staking/TAssetStaking.sol#58) is not in mixedCase
Parameter TAssetStaking.claimTokeRewards(uint256,uint256,bool,bool)._tokenId (contracts/Staking/TAssetStaking.sol#185) is not in mixedCase
Parameter TAssetStaking.claimTokeRewards(uint256,uint256,bool,bool)._cycleId (contracts/Staking/TAssetStaking.sol#186) is not in mixedCase
Parameter TAssetStaking.claimTokeRewards(uint256,uint256,bool,bool)._isConvert (contracts/Staking/TAssetStaking.sol#187) is not in mixedCase
Parameter TAssetStaking.claimTokeRewards(uint256,uint256,bool,bool)._isMint (contracts/Staking/TAssetStaking.sol#188) is not in mixedCase
Parameter TAssetStaking.claimMultipleTokeRewards(uint256,uint256[],address,bool,bool)._tokenId (contracts/Staking/TAssetStaking.sol#228) is not in mixedCase
Parameter TAssetStaking.claimMultipleTokeRewards(uint256,uint256[],address,bool,bool)._cycleIds (contracts/Staking/TAssetStaking.sol#229) is not in mixedCase
Parameter TAssetStaking.claimMultipleTokeRewards(uint256,uint256[],address,bool,bool)._isConvert (contracts/Staking/TAssetStaking.sol#231) is not in mixedCase
Parameter TAssetStaking.claimMultipleTokeRewards(uint256,uint256[],address,bool,bool)._isMint (contracts/Staking/TAssetStaking.sol#232) is not in mixedCase
Parameter TAssetStaking.getAllClaimableTokeAmount(uint256,uint256,uint256)._tokenId (contracts/Staking/TAssetStaking.sol#278) is not in mixedCase
Parameter TAssetStaking.getClaimableTokeCyclesAndAmounts(uint256,uint256,uint256)._tokenId (contracts/Staking/TAssetStaking.sol#293) is not in mixedCase
Parameter TokeStakingCommon.setMaxViewCycle(uint256)._maxViewCycle (contracts/Staking/TokeStakingCommon.sol#169) is not in mixedCase
Parameter TokeStakingCommon.claimCvgRewards(uint256,uint256)._tokenId (contracts/Staking/TokeStakingCommon.sol#182) is not in mixedCase
Parameter TokeStakingCommon.claimCvgRewards(uint256,uint256)._cycleId (contracts/Staking/TokeStakingCommon.sol#183) is not in mixedCase
Parameter TokeStakingCommon.claimMultipleCvgRewards(uint256,uint256[],address)._tokenId (contracts/Staking/TokeStakingCommon.sol#216) is not in mixedCase
Parameter TokeStakingCommon.claimMultipleCvgRewards(uint256,uint256[],address)._cycleIds (contracts/Staking/TokeStakingCommon.sol#217) is not in mixedCase
Parameter TokeStakingCommon.getRewardsForTokeCycle(uint256)._cycleId (contracts/Staking/TokeStakingCommon.sol#293) is not in mixedCase
Parameter TokeStakingCommon.findTokenStakedAmountForCvgCycle(uint256,uint256)._tokenId (contracts/Staking/TokeStakingCommon.sol#304) is not in mixedCase
Parameter TokeStakingCommon.findTokenStakedAmountForCvgCycle(uint256,uint256)._cycleId (contracts/Staking/TokeStakingCommon.sol#305) is not in mixedCase
```

AUTOMATED TESTING

```
Parameter TokeStakingCommon.findTokenStakedAmountForTokeCycle(uint256,uint256)._tokenId (contracts/Staking/TokeStakingCommon.sol#330) is not in mixedCase
Parameter TokeStakingCommon.findTokenStakedAmountForTokeCycle(uint256,uint256)._tokeCycleId (contracts/Staking/TokeStakingCommon.sol#331) is not in mixedCase
Parameter TokeStakingCommon.tokenTotalStaked(uint256)._tokenId (contracts/Staking/TokeStakingCommon.sol#356) is not in mixedCase
Parameter TokeStakingCommon.burn(uint256)._tokenId (contracts/Staking/TokeStakingCommon.sol#654) is not in mixedCase
Parameter TokeStakingCommon.getTokenInfoForCvgCycle(uint256,uint256)._tokenId (contracts/Staking/TokeStakingCommon.sol#688) is not in mixedCase
Parameter TokeStakingCommon.getTokenInfoForCvgCycle(uint256,uint256)._cycleId (contracts/Staking/TokeStakingCommon.sol#681) is not in mixedCase
Parameter TokeStakingCommon.getTokenInfoForTokeCycle(uint256,uint256)._tokenId (contracts/Staking/TokeStakingCommon.sol#691) is not in mixedCase
Parameter TokeStakingCommon.getTokenInfoForTokeCycle(uint256,uint256)._cycleId (contracts/Staking/TokeStakingCommon.sol#692) is not in mixedCase
Parameter TokeStakingCommon.getAllClaimableCvgAmount(uint256,uint256,uint256)._tokenId (contracts/Staking/TokeStakingCommon.sol#701) is not in mixedCase
Parameter TokeStakingCommon.getClaimableCvgCyclesAndAmounts(uint256,uint256,uint256)._tokenId (contracts/Staking/TokeStakingCommon.sol#733) is not in mixedCase
Parameter TokeStakingCommon.cvgPoolCvgToke(ICrvPoolPlain)._poolCvgToke (contracts/Staking/TokeStakingCommon.sol#794) is not in mixedCase
Parameter CvgERC721TimeLocking.getTokenIdsForWallet(address)._wallet (contracts/Token/CvgERC721TimeLocking.sol#36) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Slither:./contracts/Staking/TAssetStaking.sol analyzed (50 contracts with 85 detectors), 55 result(s) found
```

TokeStaker.sol

```
INFO:Detectors:
TokeStaker.depositAndProcessTokeRewards(uint256[]) (contracts/Staking/TokeStaker.sol#118-142) ignores return value by asset.transferFrom(msg.sender,_treasuryBonds,treasuryFees) (contracts/Staking/TokeStaker.sol#132)
TokeStaker.depositAndProcessTokeRewards(uint256[]) (contracts/Staking/TokeStaker.sol#118-142) ignores return value by asset.transferFrom(msg.sender,address(_cvgTokeStaking),rewardAmounts[1]) (contracts/Staking/TokeStaker.sol#133)
TokeStaker.withdrawToken(IERC20) (contracts/Staking/TokeStaker.sol#148-150) ignores return value by _token.transfer(msg.sender,_token.balanceOf(address(this))) (contracts/Staking/TokeStaker.sol#149)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unchecked-transfer
INFO:Detectors:
TokeStaker.depositAndProcessTokeRewards(uint256[]).i (contracts/Staking/TokeStaker.sol#126) is a local variable never initialized
TokeStaker.constructor(ICvgControlTower,TokeStaker.RewardsInfo[],ITokeStaking,IAccToke).i (contracts/Staking/TokeStaker.sol#63) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables
INFO:Detectors:
TokeStaker.approveToken(IERC20,address,uint256) (contracts/Staking/TokeStaker.sol#155-157) ignores return value by _token.approve(_spender,_amount) (contracts/Staking/TokeStaker.sol#156)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return
INFO:Detectors:
TokeStaker.depositAndProcessTokeRewards(uint256[]) (contracts/Staking/TokeStaker.sol#118-142) has external calls inside a loop: asset.transferFrom(msg.sender,_treasuryBonds,treasuryFees) (contracts/Staking/TokeStaker.sol#132)
TokeStaker.depositAndProcessTokeRewards(uint256[]) (contracts/Staking/TokeStaker.sol#118-142) has external calls inside a loop: asset.transferFrom(msg.sender,address(_cvgTokeStaking),rewardAmounts[1]) (contracts/Staking/TokeStaker.sol#133)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#calls-inside-a-loop
INFO:Detectors:
Pragma version^0.8.0 (contracts/Staking/TokeStaker.sol#12) allows old versions
solc-0.8.17 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Parameter TokeStaker.addRewardInfo(TokeStaker.RewardsInfo)._rewardInfo (contracts/Staking/TokeStaker.sol#76) is not in mixedCase
Parameter TokeStaker.setRewardInfo(uint256,TokeStaker.RewardsInfo)._rewardsInfo (contracts/Staking/TokeStaker.sol#87) is not in mixedCase
Parameter TokeStaker.stakeAndUpdateTokeCycle(TokeStaker.Staking)._staking (contracts/Staking/TokeStaker.sol#107) is not in mixedCase
Parameter TokeStaker.withdrawToken(IERC20)._token (contracts/Staking/TokeStaker.sol#148) is not in mixedCase
Parameter TokeStaker.approveToken(IERC20,address,uint256)._token (contracts/Staking/TokeStaker.sol#155) is not in mixedCase
Parameter TokeStaker.approveToken(IERC20,address,uint256)._spender (contracts/Staking/TokeStaker.sol#155) is not in mixedCase
Parameter TokeStaker.approveToken(IERC20,address,uint256)._amount (contracts/Staking/TokeStaker.sol#155) is not in mixedCase
Parameter TokeStaker.processTokeRewardsEmergency(uint256[],uint256)._rewardsAmount (contracts/Staking/TokeStaker.sol#164) is not in mixedCase
Parameter TokeStaker.processTokeRewardsEmergency(uint256[],uint256)._tokeStakingCycle (contracts/Staking/TokeStaker.sol#165) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
Variable TokeStaker.addRewardInfo(TokeStaker.RewardsInfo)._rewardInfo (contracts/Staking/TokeStaker.sol#76) is too similar to TokeStaker.rewardsInfo (contracts/Staking/TokeStaker.sol#44)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-too-similar
INFO:Detectors:
TokeStaker.accToke (contracts/Staking/TokeStaker.sol#38) should be immutable
TokeStaker.cvgControlTower (contracts/Staking/TokeStaker.sol#36) should be immutable
TokeStaker.tokeStaking (contracts/Staking/TokeStaker.sol#37) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
INFO:Slither:./contracts/Staking/TokeStaker.sol analyzed (40 contracts with 85 detectors), 23 result(s) found
```

Cvg.sol

```
INFO:Detectors:
Cvg.constructor(ICvgControlTower,address,address)._airdrop (contracts/Token/Cvg.sol#38) lacks a zero-check on :
        - airdrop = _airdrop (contracts/Token/Cvg.sol#33)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation
INFO:Detectors:
Pragma version^0.8.0 (contracts/Token/Cvg.sol#12) allows old versions
solc-0.8.17 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Slither:./contracts/Token/Cvg.sol analyzed (37 contracts with 85 detectors), 3 result(s) found
```

CvgToke.sol

```
INFO:Detectors:
CvgToke.mint(address,uint256) (contracts/Token/CvgToke.sol#24-27) ignores return value by cvgControlTower.toke().transferFrom(msg.sender,cvgControlTower.treasuryStaking(),amount) (contracts/Token/CvgToke.sol#25)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unchecked-transfer
INFO:Detectors:
Pragma version^0.8.0 (contracts/Token/CvgToke.sol#12) allows old versions
solc-0.8.17 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
CvgToke.cvgControlTower (contracts/Token/CvgToke.sol#18) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
INFO:Slither:./contracts/Token/CvgToke.sol analyzed (37 contracts with 85 detectors), 4 result(s) found
```

CvgUtilities.sol

```
INFO:Detectors:
CvgUtilities.swapTokenAndStakeCvgToke(uint256,uint256,uint256,bool,IAggregationRouterV5.SwapTransaction) (contracts/utils/CvgUtilities.sol#151-204) ignores return value by toke.transferFrom(msg.sender,address(this),_tokeAmount) (contracts/utils/CvgUtilities.sol#171)
CvgUtilities.swapTokenAndStakeCvgToke(uint256,uint256,uint256,bool,IAggregationRouterV5.SwapTransaction) (contracts/utils/CvgUtilities.sol#151-204) ignores return value by _cvgToke.transferFrom(msg.sender,address(this),_cvgTokeAmount) (contracts/utils/CvgUtilities.sol#195)
CvgUtilities.swapTokenAndStakeTasset(uint256,uint256,uint256,IERC20,IAggregationRouterV5.SwapTransaction) (contracts/utils/CvgUtilities.sol#214-255) ignores return value by _underlyingAsset.transferFrom(msg.sender,address(this),_underlyingAssetAmount) (contracts/utils/CvgUtilities.sol#234)
CvgUtilities.swapTokenAndStakeTasset(uint256,uint256,uint256,IERC20,IAggregationRouterV5.SwapTransaction) (contracts/utils/CvgUtilities.sol#214-255) ignores return value by _tAssetData.tAsset.transferFrom(msg.sender,address(this),_tAssetAmount) (contracts/utils/CvgUtilities.sol#246)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unchecked-transfer
INFO:Detectors:
CvgUtilities.claimMultipleLocking(CvgUtilities.ClaimTokenTde[]).j (contracts/utils/CvgUtilities.sol#107) is a local variable never initialized
CvgUtilities.claimMultipleBonds(CvgUtilities.ClaimBondContract[],address).i (contracts/utils/CvgUtilities.sol#132) is a local variable never initialized
CvgUtilities.swapTokenAndStakeCvgToke(uint256,uint256,uint256,bool,IAggregationRouterV5.SwapTransaction).totalTokeAmount (contracts/utils/CvgUtilities.sol#159) is a local variable never initialized
CvgUtilities.claimMultipleStakingToke(CvgUtilities.ClaimStakingContract[]).j (contracts/utils/CvgUtilities.sol#78) is a local variable never initialized
CvgUtilities.claimMultipleStakingToke(CvgUtilities.ClaimStakingContract[]).i (contracts/utils/CvgUtilities.sol#75) is a local variable never initialized
CvgUtilities.swapTokenAndStakeTasset(uint256,uint256,uint256,IERC20,IAggregationRouterV5.SwapTransaction).amount (contracts/utils/CvgUtilities.sol#224) is a local variable never initialized
CvgUtilities.claimMultipleLocking(CvgUtilities.ClaimTokenTde[]).i (contracts/utils/CvgUtilities.sol#105) is a local variable never initialized
CvgUtilities.claimMultipleBonds(CvgUtilities.ClaimBondContract[],address).i (contracts/utils/CvgUtilities.sol#129) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables
INFO:Detectors:
CvgUtilities.claimMultipleBonds(CvgUtilities.ClaimBondContract[],address) (contracts/utils/CvgUtilities.sol#128-142) ignores return value by bondDepository.redeem(claimContracts[i].tokenIds[j],recipient,msg.sender) (contracts/utils/CvgUtilities.sol#133)
CvgUtilities.swapTokenAndStakeCvgToke(uint256,uint256,uint256,bool,IAggregationRouterV5.SwapTransaction) (contracts/utils/CvgUtilities.sol#151-204) ignores return value by toke.approve(address(_cvgToke),totalTokeAmount) (contracts/utils/CvgUtilities.sol#181)
CvgUtilities.swapTokenAndStakeCvgToke(uint256,uint256,uint256,bool,IAggregationRouterV5.SwapTransaction) (contracts/utils/CvgUtilities.sol#151-204) ignores return value by toke.approve(address(_poolCvgToke),totalTokeAmount) (contracts/utils/CvgUtilities.sol#187)
CvgUtilities.swapTokenAndStakeCvgToke(uint256,uint256,uint256,bool,IAggregationRouterV5.SwapTransaction) (contracts/utils/CvgUtilities.sol#151-204) ignores return value by _cvgToke.approve(address(_cvgControlTower.cvgTokeStaking()),totalTokeAmount) (contracts/utils/CvgUtilities.sol#200)
CvgUtilities.swapTokenAndStakeTasset(uint256,uint256,uint256,IERC20,IAggregationRouterV5.SwapTransaction) (contracts/utils/CvgUtilities.sol#214-255) ignores return value by _underlyingAsset.approve(address(_tAssetData.tAsset),amount) (contracts/utils/CvgUtilities.sol#240)
CvgUtilities.swapTokenAndStakeTasset(uint256,uint256,uint256,IERC20,IAggregationRouterV5.SwapTransaction) (contracts/utils/CvgUtilities.sol#214-255) ignores return value by _tAssetData.tAsset.approve(address(_tAssetData.stakingContract),amount) (contracts/utils/CvgUtilities.sol#251)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return
INFO:Detectors:
CvgUtilities.claimMultipleStakingToke(CvgUtilities.ClaimStakingContract[]) (contracts/utils/CvgUtilities.sol#73-101) has external calls inside a loop: tokeStaking.claimMultipleCvgRewards(tokenId,claim.cvgCycles,msg.sender) (contracts/utils/CvgUtilities.sol#82)
CvgUtilities.claimMultipleStakingToke(CvgUtilities.ClaimStakingContract[]) (contracts/utils/CvgUtilities.sol#73-101) has external calls inside a loop: tokeStaking.claimMultipleTokeRewards(tokenId,claim.tokeCycles,msg.sender,claim.isConvert,claim.isMint) (contracts/utils/CvgUtilities.sol#85-91)
CvgUtilities.claimMultipleLocking(CvgUtilities.ClaimTokenTde[]) (contracts/utils/CvgUtilities.sol#103-126) has external calls inside a loop: ydistributor.claimRewards(tokenId,tdeId,msg.sender,msg.sender,new IAggregationRouterV5.SwapTransaction[](0),IERC20(address(0))) (contracts/utils/CvgUtilities.sol#110-117)
CvgUtilities.claimMultipleBonds(CvgUtilities.ClaimBondContract[],address) (contracts/utils/CvgUtilities.sol#128-142) has external calls inside a loop: bondDepository.redeem(claimContracts[i].tokenIds[j],recipient,msg.sender) (contracts/utils/CvgUtilities.sol#133)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#calls-inside-a-loop
INFO:Detectors:
Pragma version^0.8.0 (contracts/utils/CvgUtilities.sol#12) allows old versions
solc-0.8.17 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Parameter CvgUtilities.swapTokenAndStakeCvgToke(uint256,uint256,uint256,bool,IAggregationRouterV5.SwapTransaction)._tokenId (contracts/utils/CvgUtilities.sol#152) is not in mixedCase
Parameter CvgUtilities.swapTokenAndStakeCvgToke(uint256,uint256,uint256,bool,IAggregationRouterV5.SwapTransaction)._tokeAmount (contracts/utils/CvgUtilities.sol#153) is not in mixedCase
Parameter CvgUtilities.swapTokenAndStakeCvgToke(uint256,uint256,uint256,bool,IAggregationRouterV5.SwapTransaction)._cvgTokeAmount (contracts/utils/CvgUtilities.sol#154) is not in mixedCase
Parameter CvgUtilities.swapTokenAndStakeCvgToke(uint256,uint256,uint256,bool,IAggregationRouterV5.SwapTransaction)._swapTransaction (contracts/utils/CvgUtilities.sol#156) is not in mixedCase
Parameter CvgUtilities.swapTokenAndStakeTasset(uint256,uint256,uint256,IERC20,IAggregationRouterV5.SwapTransaction)._tokenId (contracts/utils/CvgUtilities.sol#215) is not in mixedCase
Parameter CvgUtilities.swapTokenAndStakeTasset(uint256,uint256,uint256,IERC20,IAggregationRouterV5.SwapTransaction)._underlyingAssetAmount (contracts/utils/CvgUtilities.sol#216) is not in mixedCase
Parameter CvgUtilities.swapTokenAndStakeTasset(uint256,uint256,uint256,IERC20,IAggregationRouterV5.SwapTransaction)._tAssetAmount (contracts/utils/CvgUtilities.sol#217) is not in mixedCase
Parameter CvgUtilities.swapTokenAndStakeTasset(uint256,uint256,uint256,IERC20,IAggregationRouterV5.SwapTransaction)._underlyingAsset (contracts/utils/CvgUtilities.sol#218) is not in mixedCase
Parameter CvgUtilities.swapTokenAndStakeTasset(uint256,uint256,uint256,IERC20,IAggregationRouterV5.SwapTransaction)._swapTransaction (contracts/utils/CvgUtilities.sol#219) is not in mixedCase
Parameter CvgUtilities.enableTAssetForSwap(address,address)._tAssetContract (contracts/utils/CvgUtilities.sol#262) is not in mixedCase
Parameter CvgUtilities.enableTAssetForSwap(address,address)._stakingContract (contracts/utils/CvgUtilities.sol#262) is not in mixedCase
Parameter CvgUtilities.disableTAssetForSwap(address)._tAssetContract (contracts/utils/CvgUtilities.sol#274) is not in mixedCase
Parameter CvgUtilities.setPoolCvgToke(ICrvPoolPlain)._poolCvgToke (contracts/utils/CvgUtilities.sol#280) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
Variable CvgUtilities.swapTokenAndStakeTasset(uint256,uint256,uint256,IERC20,IAggregationRouterV5.SwapTransaction)._underlyingAsset (contracts/utils/CvgUtilities.sol#218) is too similar to CvgUtilities.underlyingTAsset (contracts/utils/CvgUtilities.sol#58)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-too-similar
INFO:Detectors:
CvgUtilities.cvgControlTower (contracts/utils/CvgUtilities.sol#52) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
INFO:Slither:./contracts/utils/CvgUtilities.sol analyzed (39 contracts with 85 detectors), 39 result(s) found
```

SwapperFactory.sol

```
INFO:Detectors:
SwapperFactory.executeSwapForCvgToke(address,IAggregationRouterV5.SwapTransaction) (contracts/utils/SwapperFactory.sol#84-126) uses arbitrary from in transferFrom: _swapTransaction.description.srcToken.transferFrom(_user,address(this),_swapTransaction.description.amount) (contracts/utils/SwapperFactory.sol#106-110)
SwapperFactory.executeSwapForTAsset(address,IAggregationRouterV5.SwapTransaction) (contracts/utils/SwapperFactory.sol#133-169) uses arbitrary from in transferFrom: _swapTransaction.description.srcToken.transferFrom(_user,address(this),_swapTransaction.description.amount) (contracts/utils/SwapperFactory.sol#149-153)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#arbitrary-from-in-transferfrom
INFO:Detectors:
SwapperFactory.executeSwapForCvgToke(address,IAggregationRouterV5.SwapTransaction) (contracts/utils/SwapperFactory.sol#84-126) ignores return value by _swapTransaction.description.srcToken.transferFrom(_user,address(this),_swapTransaction.description.amount) (contracts/utils/SwapperFactory.sol#106-110)
SwapperFactory.executeSwapForTAsset(address,IAggregationRouterV5.SwapTransaction) (contracts/utils/SwapperFactory.sol#133-169) ignores return value by _swapTransaction.description.srcToken.transferFrom(_user,address(this),_swapTransaction.description.amount) (contracts/utils/SwapperFactory.sol#149-153)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unchecked-transfer
INFO:Detectors:
SwapperFactory.swap(IAggregationRouterV5.SwapTransaction) (contracts/utils/SwapperFactory.sol#52-77) ignores return value by _swapTransaction.description.srcToken.approve(address(_aggregationRouter),0) (contracts/utils/SwapperFactory.sol#62-65)
SwapperFactory.swap(IAggregationRouterV5.SwapTransaction) (contracts/utils/SwapperFactory.sol#52-77) ignores return value by _swapTransaction.description.srcToken.approve(address(_aggregationRouter),_swapTransaction.description.amount) (contracts/utils/SwapperFactory.sol#66-69)
SwapperFactory.swap(IAggregationRouterV5.SwapTransaction) (contracts/utils/SwapperFactory.sol#52-77) ignores return value by _aggregationRouter.swap(_swapTransaction.executor,_swapTransaction.description,_swapTransaction.permit,_swapTransaction.data) (contracts/utils/SwapperFactory.sol#71-76)
SwapperFactory.executeSwapForCvgToke(address,IAggregationRouterV5.SwapTransaction) (contracts/utils/SwapperFactory.sol#84-126) ignores return value by _swapTransaction.description.srcToken.approve(address(_aggregationRouter),_swapTransaction.description.amount) (contracts/utils/SwapperFactory.sol#114-117)
SwapperFactory.executeSwapForTAsset(address,IAggregationRouterV5.SwapTransaction) (contracts/utils/SwapperFactory.sol#133-169) ignores return value by _swapTransaction.description.srcToken.approve(address(_aggregationRouter),_swapTransaction.description.amount) (contracts/utils/SwapperFactory.sol#157-160)
SwapperFactory.approveRouterTokenSpending(IERC20,uint256) (contracts/utils/SwapperFactory.sol#176-181) ignores return value by _token.approve(address(aggregationRouter),_amount) (contracts/utils/SwapperFactory.sol#180)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return
INFO:Detectors:
Pragma version^0.8.0 (contracts/utils/SwapperFactory.sol#12) allows old versions
solc-0.8.17 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Parameter SwapperFactory.swap(IAggregationRouterV5.SwapTransaction)._swapTransaction (contracts/utils/SwapperFactory.sol#53) is not in mixedCase
Parameter SwapperFactory.executeSwapForCvgToke(address,IAggregationRouterV5.SwapTransaction)._user (contracts/utils/SwapperFactory.sol#85) is not in mixedCase
Parameter SwapperFactory.executeSwapForCvgToke(address,IAggregationRouterV5.SwapTransaction)._swapTransaction (contracts/utils/SwapperFactory.sol#86) is not in mixedCase
Parameter SwapperFactory.executeSwapForTAsset(address,IAggregationRouterV5.SwapTransaction)._user (contracts/utils/SwapperFactory.sol#134) is not in mixedCase
Parameter SwapperFactory.executeSwapForTAsset(address,IAggregationRouterV5.SwapTransaction)._swapTransaction (contracts/utils/SwapperFactory.sol#135) is not in mixedCase
Parameter SwapperFactory.approveRouterTokenSpending(IERC20,uint256)._token (contracts/utils/SwapperFactory.sol#177) is not in mixedCase
Parameter SwapperFactory.approveRouterTokenSpending(IERC20,uint256)._amount (contracts/utils/SwapperFactory.sol#178) is not in mixedCase
Parameter SwapperFactory.toggleSourceToken(IERC20)._token (contracts/utils/SwapperFactory.sol#187) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Slither:./contracts/utils/SwapperFactory.sol analyzed (39 contracts with 85 detectors), 20 result(s) found
```

CloneFactory.sol

```
INFO:Detectors:
Reentrancy in CloneFactory.createBond(IBondStruct.BondParams,IOracleStruct.OracleParams,uint256) (contracts/CloneFactory.sol#49-61):
        External calls:
        - IBondDepository(newClone).initialize(_cvgControlTower,_bondParams,_oracleParams) (contracts/CloneFactory.sol#58)
        - _cvgControlTower.insertNewBond(newClone,_version) (contracts/CloneFactory.sol#59)
        Event emitted after the call(s):
        - BondCreated(baseImplementation,newClone,_bondParams) (contracts/CloneFactory.sol#60)
Reentrancy in CloneFactory.createCvgAggregator(uint8,string,uint256) (contracts/CloneFactory.sol#91-99):
        External calls:
        - ICvgAggregatorV3(newClone).initialize(_cvgControlTower,_decimals,_description) (contracts/CloneFactory.sol#96)
        - _cvgControlTower.insertNewAggregator(newClone,_version) (contracts/CloneFactory.sol#97)
        Event emitted after the call(s):
        - AggregatorCreated(baseImplementation,newClone) (contracts/CloneFactory.sol#98)
Reentrancy in CloneFactory.createTAssetStaking(IERC20Metadata,uint256,string,string) (contracts/CloneFactory.sol#68-84):
        External calls:
        - ITAssetStaking(newClone).initialize(_cvgControlTower,_asset,_name,_symbol) (contracts/CloneFactory.sol#79)
        - _cvgControlTower.insertNewTAssetStaking(newClone,_version) (contracts/CloneFactory.sol#81)
        Event emitted after the call(s):
        - TAssetStakingCreated(baseImplementation,newClone,_asset) (contracts/CloneFactory.sol#83)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3
INFO:Detectors:
CloneFactory._clone(address) (contracts/CloneFactory.sol#110-128) uses assembly
        - INLINE ASM (contracts/CloneFactory.sol#112-118)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
INFO:Detectors:
Pragma version^0.8.0 (contracts/CloneFactory.sol#12) allows old versions
solc-0.8.17 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Parameter CloneFactory.initialize(ICvgControlTower)._cvgControlTower (contracts/CloneFactory.sol#34) is not in mixedCase
Parameter CloneFactory.createBond(IBondStruct.BondParams,IOracleStruct.OracleParams,uint256)._bondParams (contracts/CloneFactory.sol#50) is not in mixedCase
Parameter CloneFactory.createBond(IBondStruct.BondParams,IOracleStruct.OracleParams,uint256)._oracleParams (contracts/CloneFactory.sol#51) is not in mixedCase
Parameter CloneFactory.createBond(IBondStruct.BondParams,IOracleStruct.OracleParams,uint256)._version (contracts/CloneFactory.sol#52) is not in mixedCase
Parameter CloneFactory.createTAssetStaking(IERC20Metadata,uint256,string,string)._asset (contracts/CloneFactory.sol#69) is not in mixedCase
Parameter CloneFactory.createTAssetStaking(IERC20Metadata,uint256,string,string)._version (contracts/CloneFactory.sol#70) is not in mixedCase
Parameter CloneFactory.createTAssetStaking(IERC20Metadata,uint256,string,string)._name (contracts/CloneFactory.sol#71) is not in mixedCase
Parameter CloneFactory.createTAssetStaking(IERC20Metadata,uint256,string,string)._symbol (contracts/CloneFactory.sol#72) is not in mixedCase
Parameter CloneFactory.createCvgAggregator(uint8,string,uint256)._decimals (contracts/CloneFactory.sol#91) is not in mixedCase
Parameter CloneFactory.createCvgAggregator(uint8,string,uint256)._description (contracts/CloneFactory.sol#91) is not in mixedCase
Parameter CloneFactory.createCvgAggregator(uint8,string,uint256)._version (contracts/CloneFactory.sol#91) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
CloneFactory._clone(address) (contracts/CloneFactory.sol#110-128) uses literals with too many digits:
        - mstore(uint256,uint256)(ptr__clone_asm_0,0x3d602d80600a3d3981f3363d3d373d3d3d363d73bebebebebebebebebebebebebebebebe) (contracts/CloneFactory.sol#114)
CloneFactory._clone(address) (contracts/CloneFactory.sol#110-128) uses literals with too many digits:
        - mstore(uint256,uint256)(ptr__clone_asm_0 + 0x28,0x5af43d82803e903d91602b57fd5bf3bebebebebebebebebebebebebebebebe) (contracts/CloneFactory.sol#116)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits
INFO:Slither:./contracts/CloneFactory.sol analyzed (41 contracts with 85 detectors), 19 result(s) found
```

CvgControlTower.sol

```
INFO:Detectors:
CvgControlTower.getTAssetStakingContractsPerVersion(uint256,uint256,uint256).i (contracts/CvgControlTower.sol#231) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables
INFO:Detectors:
CvgControlTower.initialize(address,address,address)._treasuryBonds (contracts/CvgControlTower.sol#90) lacks a zero-check on :
                - treasuryBonds = _treasuryBonds (contracts/CvgControlTower.sol#91)
CvgControlTower.initialize(address,address,address)._treasuryStaking (contracts/CvgControlTower.sol#90) lacks a zero-check on :
                - treasuryStaking = _treasuryStaking (contracts/CvgControlTower.sol#92)
CvgControlTower.initialize(address,address,address)._treasuryDao (contracts/CvgControlTower.sol#90) lacks a zero-check on :
                - treasuryDao = _treasuryDao (contracts/CvgControlTower.sol#93)
CvgControlTower.setTreasuryStaking(address).newTreasuryStakingMultisig (contracts/CvgControlTower.sol#275) lacks a zero-check on :
                - treasuryStaking = newTreasuryStakingMultisig (contracts/CvgControlTower.sol#276)
CvgControlTower.setTreasuryBonds(address).newTreasuryBondsMultisig (contracts/CvgControlTower.sol#279) lacks a zero-check on :
                - treasuryBonds = newTreasuryBondsMultisig (contracts/CvgControlTower.sol#280)
CvgControlTower.setTreasuryDao(address).newTreasuryDao (contracts/CvgControlTower.sol#287) lacks a zero-check on :
                - treasuryDao = newTreasuryDao (contracts/CvgControlTower.sol#288)
CvgControlTower.setCloneFactory(address).newCloneFactory (contracts/CvgControlTower.sol#311) lacks a zero-check on :
                - cloneFactory = newCloneFactory (contracts/CvgControlTower.sol#312)
CvgControlTower.setCvgUtilities(address)._cvgUtilities (contracts/CvgControlTower.sol#379) lacks a zero-check on :
                - cvgUtilities = _cvgUtilities (contracts/CvgControlTower.sol#380)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation
INFO:Detectors:
CvgControlTower.getBondContractsPerVersion(uint256,uint256,uint256) (contracts/CvgControlTower.sol#174-193) has external calls inside a loop: bondStructs[i] = IBondDepository(bondAddresses[i]).getBondView() (contracts/CvgControlTower.sol#187)
CvgControlTower.getTAssetStakingContractsPerVersion(uint256,uint256,uint256) (contracts/CvgControlTower.sol#217-238) has external calls inside a loop: stakingStruct[i] = _stakingViewer.getGlobalViewTokeStaking(stakingAddresses[i]) (contracts/CvgControlTower.sol#232)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#calls-inside-a-loop
INFO:Detectors:
Reentrancy in CvgControlTower.updateCvgCycle() (contracts/CvgControlTower.sol#392-398):
        External calls:
        - lockingPositionManager.updateYsTotalSupply() (contracts/CvgControlTower.sol#396)
        Event emitted after the call(s):
        - NewCycle(_newCycle) (contracts/CvgControlTower.sol#397)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3
INFO:Detectors:
Pragma version^0.8.0 (contracts/CvgControlTower.sol#12) allows old versions
solc-0.8.17 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Parameter CvgControlTower.initialize(address,address,address)._treasuryBonds (contracts/CvgControlTower.sol#90) is not in mixedCase
Parameter CvgControlTower.initialize(address,address,address)._treasuryStaking (contracts/CvgControlTower.sol#90) is not in mixedCase
Parameter CvgControlTower.initialize(address,address,address)._treasuryDao (contracts/CvgControlTower.sol#90) is not in mixedCase
Parameter CvgControlTower.insertNewBond(address,uint256)._newClone (contracts/CvgControlTower.sol#102) is not in mixedCase
Parameter CvgControlTower.insertNewBond(address,uint256)._version (contracts/CvgControlTower.sol#102) is not in mixedCase
Parameter CvgControlTower.insertNewTAssetStaking(address,uint256)._newClone (contracts/CvgControlTower.sol#111) is not in mixedCase
Parameter CvgControlTower.insertNewTAssetStaking(address,uint256)._version (contracts/CvgControlTower.sol#111) is not in mixedCase
Parameter CvgControlTower.insertNewAggregator(address,uint256)._newClone (contracts/CvgControlTower.sol#124) is not in mixedCase
Parameter CvgControlTower.insertNewAggregator(address,uint256)._version (contracts/CvgControlTower.sol#124) is not in mixedCase
Parameter CvgControlTower.getBondLenghtPerVersion(uint256)._baseBondVersion (contracts/CvgControlTower.sol#170) is not in mixedCase
Parameter CvgControlTower.getBondContractsPerVersion(uint256,uint256,uint256)._baseBondVersion (contracts/CvgControlTower.sol#175) is not in mixedCase
Parameter CvgControlTower.getBondContractsPerVersion(uint256,uint256,uint256)._cursor (contracts/CvgControlTower.sol#176) is not in mixedCase
Parameter CvgControlTower.getBondContractsPerVersion(uint256,uint256,uint256)._lengthDesired (contracts/CvgControlTower.sol#177) is not in mixedCase
Parameter CvgControlTower.getTAssetStakingContractsPerVersion(uint256,uint256,uint256)._baseStakingVersion (contracts/CvgControlTower.sol#218) is not in mixedCase
Parameter CvgControlTower.getTAssetStakingContractsPerVersion(uint256,uint256,uint256)._cursor (contracts/CvgControlTower.sol#219) is not in mixedCase
Parameter CvgControlTower.getTAssetStakingContractsPerVersion(uint256,uint256,uint256)._lengthDesired (contracts/CvgControlTower.sol#220) is not in mixedCase
Parameter CvgControlTower.setNewVersionBaseBond(address)._newBaseBond (contracts/CvgControlTower.sol#315) is not in mixedCase
Parameter CvgControlTower.setNewVersionBaseTAssetStaking(address)._newBaseStaking (contracts/CvgControlTower.sol#319) is not in mixedCase
Parameter CvgControlTower.setNewVersionBaseAggregator(address)._newBaseAggregator (contracts/CvgControlTower.sol#323) is not in mixedCase
Parameter CvgControlTower.setTokeManager(ITokeManager)._tokeManager (contracts/CvgControlTower.sol#335) is not in mixedCase
Parameter CvgControlTower.setTokeRewards(ITokeRewards)._tokeRewards (contracts/CvgControlTower.sol#339) is not in mixedCase
Parameter CvgControlTower.setWethRewards(ITokeRewards)._wethRewards (contracts/CvgControlTower.sol#343) is not in mixedCase
Parameter CvgControlTower.setToke(IERC20Metadata)._toke (contracts/CvgControlTower.sol#347) is not in mixedCase
Parameter CvgControlTower.setYsDistributor(IYsDistributor)._ysDistributor (contracts/CvgControlTower.sol#351) is not in mixedCase
Parameter CvgControlTower.setCvgToke(IERC20Mintable)._cvgToke (contracts/CvgControlTower.sol#355) is not in mixedCase
Parameter CvgControlTower.setCvg(ICvg)._cvgToken (contracts/CvgControlTower.sol#359) is not in mixedCase
Parameter CvgControlTower.setTAssetBlackHole(ITAssetBlackHole)._tAssetBlackHole (contracts/CvgControlTower.sol#363) is not in mixedCase
Parameter CvgControlTower.setBondPositionManager(IBondPositionManager)._bondPositionManager (contracts/CvgControlTower.sol#367) is not in mixedCase
Parameter CvgControlTower.setStakingViewer(IStakingViewer)._stakingViewer (contracts/CvgControlTower.sol#371) is not in mixedCase
Parameter CvgControlTower.setStakingLogo(IStakingLogo)._stakingLogo (contracts/CvgControlTower.sol#375) is not in mixedCase
Parameter CvgControlTower.setCvgUtilities(address)._cvgUtilities (contracts/CvgControlTower.sol#379) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Slither:./contracts/CvgControlTower.sol analyzed (40 contracts with 85 detectors), 45 result(s) found
```

The above output was reviewed, and all vulnerabilities were determined to be false positives and were not included in the report.

6.2 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 the smart contracts and sent the compiled results to the analyzers in order to locate any vulnerabilities.

Please see the following pages for the results.

Report for contracts/Bond/BondPositionManager.sol
https://dashboard.mythx.io/#/console/analyses/f1581917-bb5f-4bf8-b872-f454c7e4a1f7

Line	SWC Title	Severity	Short Description
12	(SWC-103) Floating Pragma	Low	A floating pragma is set.
47	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
65	(SWC-110) Assert Violation	Unknown	Out of bounds array access
67	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered

Report for contracts/Token/CvgERC721TimeLocking.sol
https://dashboard.mythx.io/#/console/analyses/b36c4bca-ea81-4e2c-8188-c530df43df44
https://dashboard.mythx.io/#/console/analyses/d8cceb9a-2cd4-4cc2-a43d-613db3fc3bf4
https://dashboard.mythx.io/#/console/analyses/f1581917-bb5f-4bf8-b872-f454c7e4a1f7
https://dashboard.mythx.io/#/console/analyses/07994b21-6a39-4f87-b12a-1d0c27f7d19d

Line	SWC Title	Severity	Short Description
12	(SWC-103) Floating Pragma	Low	A floating pragma is set.
39	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
40	(SWC-110) Assert Violation	Unknown	Out of bounds array access
63	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered

Report for Bond/BondCalculator.sol
https://dashboard.mythx.io/#/console/analyses/9a591ac9-b9e8-4eef-a1cf-d6e7112939d6

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

Report for libs/ABDKMathQuad.sol
https://dashboard.mythx.io/#/console/analyses/9a591ac9-b9e8-4eef-a1cf-d6e7112939d6

Line	SWC Title	Severity	Short Description
545	(SWC-101) Integer Overflow and Underflow	High	The arithmetic operation can underflow.
589	(SWC-101) Integer Overflow and Underflow	High	The arithmetic operation can underflow.
979	(SWC-101) Integer Overflow and Underflow	High	The arithmetic operation can overflow.

Report for contracts/Bond/BondLogo.sol
https://dashboard.mythx.io/#/console/analyses/61343cd1-0601-4e98-bc67-a6e8b5b9d2d8

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

Report for contracts/libs/Base64.sol
https://dashboard.mythx.io/#/console/analyses/212e676d-ebc6-4fa2-aa93-6fe1792c1e58
https://dashboard.mythx.io/#/console/analyses/770ebed8-3f78-4918-aa1d-6707b04a413e

Line	SWC Title	Severity	Short Description
17	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
17	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
17	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
20	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered

Report for contracts/Bond/BondDepository.sol
https://dashboard.mythx.io/#/console/analyses/e992532f-e265-4f40-bbb6-415628b01634

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

Report for contracts/Locking/LockingLogo.sol
https://dashboard.mythx.io/#/console/analyses/212e676d-ebc6-4fa2-aa93-6fe1792c1e58

Line	SWC Title	Severity	Short Description
2	(SWC-103) Floating Pragma	Low	A floating pragma is set.
69	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
70	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/=" discovered
74	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-=" discovered
75	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered

75	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "%" discovered
75	(SWC-110) Assert Violation	Unknown	Out of bounds array access
76	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/=" discovered
93	(SWC-110) Assert Violation	Unknown	Out of bounds array access
98	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
122	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
132	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "**" discovered
132	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
132	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
152	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
153	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
157	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "--" discovered
171	(SWC-110) Assert Violation	Unknown	Out of bounds array access
175	(SWC-110) Assert Violation	Unknown	Out of bounds array access
180	(SWC-110) Assert Violation	Unknown	Out of bounds array access
182	(SWC-110) Assert Violation	Unknown	Out of bounds array access
184	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
184	(SWC-110) Assert Violation	Unknown	Out of bounds array access
184	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
186	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
189	(SWC-110) Assert Violation	Unknown	Out of bounds array access
189	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
189	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
193	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
203	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
213	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
213	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
271	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
278	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
278	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
279	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
279	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
291	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
309	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
309	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "**" discovered
311	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "**" discovered
311	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
327	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
327	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "**" discovered
335	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "**" discovered
335	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
343	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "**" discovered
343	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
351	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
351	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "**" discovered

Report for contracts/Locking/LockingPositionManager.sol
https://dashboard.mythx.io/#/console/analyses/d8cceb9a-2cd4-4cc2-a43d-613db3fc3bf4

Line	SWC Title	Severity	Short Description

12	(SWC-103) Floating Pragma	Low	A floating pragma is set.
94	(SWC-110) Assert Violation	Unknown	Public state variable with array type causing reacheable exception by default.
130	(SWC-115) Authorization through tx.origin	Low	Use of "tx.origin" as a part of authorization control.
210	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "%" discovered
218	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
220	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "%" discovered
222	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
229	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
229	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
237	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
237	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
242	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
243	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
243	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
247	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
247	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
248	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
317	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
318	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
318	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
327	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
328	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
331	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
334	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
334	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
334	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
335	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
336	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
340	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
385	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
386	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
393	(SWC-101) Integer Overflow and Underflow	Unknown	Compiler-rewritable "<uint> - 1" discovered
393	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
400	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "%" discovered
407	(SWC-101) Integer Overflow and Underflow	Unknown	Compiler-rewritable "<uint> - 1" discovered
407	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
409	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
411	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
419	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
419	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
419	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
424	(SWC-101) Integer Overflow and Underflow	Unknown	Compiler-rewritable "<uint> - 1" discovered
424	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
426	(SWC-101) Integer Overflow and Underflow	Unknown	Compiler-rewritable "<uint> - 1" discovered
426	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
454	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
455	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
459	(SWC-101) Integer Overflow and Underflow	Unknown	Compiler-rewritable "<uint> - 1" discovered

459	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
462	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "%" discovered
465	(SWC-101) Integer Overflow and Underflow	Unknown	Compiler-rewritable "<uint> - 1" discovered
465	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
466	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
467	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
470	(SWC-101) Integer Overflow and Underflow	Unknown	Compiler-rewritable "<uint> - 1" discovered
470	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
471	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
471	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
473	(SWC-101) Integer Overflow and Underflow	Unknown	Compiler-rewritable "<uint> - 1" discovered
473	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
480	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
481	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
483	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
483	(SWC-101) Integer Overflow and Underflow	Unknown	Compiler-rewritable "<uint> - 1" discovered
483	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
483	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
484	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
485	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
489	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
489	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
489	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
493	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
498	(SWC-101) Integer Overflow and Underflow	Unknown	Compiler-rewritable "<uint> - 1" discovered
498	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
499	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
503	(SWC-101) Integer Overflow and Underflow	Unknown	Compiler-rewritable "<uint> - 1" discovered
503	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
520	(SWC-101) Integer Overflow and Underflow	Unknown	Compiler-rewritable "<uint> - 1" discovered
520	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
569	(SWC-101) Integer Overflow and Underflow	Unknown	Compiler-rewritable "<uint> - 1" discovered
569	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
574	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
574	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
581	(SWC-101) Integer Overflow and Underflow	Unknown	Compiler-rewritable "<uint> - 1" discovered
581	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
601	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
601	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
602	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
603	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
606	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "%" discovered
608	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
608	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
608	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
612	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
612	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
612	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered

617	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
621	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
622	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
624	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
628	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
633	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
634	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
700	(SWC-110) Assert Violation	Unknown	Out of bounds array access
701	(SWC-110) Assert Violation	Unknown	Out of bounds array access
702	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
703	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
703	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
704	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
704	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
704	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
708	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
708	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
708	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
711	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
712	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
714	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
719	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
743	(SWC-110) Assert Violation	Unknown	Out of bounds array access
745	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
747	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
776	(SWC-110) Assert Violation	Unknown	Out of bounds array access
779	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
783	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
789	(SWC-110) Assert Violation	Unknown	Out of bounds array access
796	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
800	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
826	(SWC-110) Assert Violation	Unknown	Out of bounds array access
838	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
839	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
839	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
844	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
850	(SWC-110) Assert Violation	Unknown	Out of bounds array access
861	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
862	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
862	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
863	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
868	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
883	(SWC-101) Integer Overflow and Underflow	Unknown	Compiler-rewritable "<uint> - 1" discovered
883	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
885	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
886	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-=" discovered
887	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
900	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered

901	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
901	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "**" discovered
901	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered

Report for contracts/Locking/LockingPositionDelegate.sol
https://dashboard.mythx.io/#/console/analyses/7089e1a5-a045-4570-82f3-cc91d7d8b343

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

Report for contracts/Locking/VveCVGCalculator.sol
https://dashboard.mythx.io/#/console/analyses/43e9f307-c217-455b-95bf-49239becf026

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

Report for contracts/Oracles/CvgV3Aggregator.sol
https://dashboard.mythx.io/#/console/analyses/b4caa49f-9326-49b5-a435-932b997647e3

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

Report for contracts/Oracles/CvgOracle.sol
https://dashboard.mythx.io/#/console/analyses/67f3ddac-fd86-42b2-aa1d-26218998e1f6

Line	SWC Title	Severity	Short Description
12	(SWC-103) Floating Pragma	Low	A floating pragma is set.
104	(SWC-110) Assert Violation	Unknown	Out of bounds array access
105	(SWC-110) Assert Violation	Unknown	Out of bounds array access
113	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
113	(SWC-110) Assert Violation	Unknown	Out of bounds array access
113	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
118	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
118	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
120	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
120	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "**" discovered
120	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
139	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "**" discovered
139	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
139	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
140	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
140	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "**" discovered
140	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
176	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
176	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "**" discovered
184	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "**" discovered
184	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
184	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
197	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "**" discovered
197	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
197	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
214	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
215	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
249	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "**" discovered
254	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
254	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
258	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered

259	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered

Report for contracts/libs/TickMath.sol
https://dashboard.mythx.io/#/console/analyses/67f3ddac-fd86-42b2-aa1d-26218998e1f6

Line	SWC Title	Severity	Short Description
28	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
29	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
30	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
31	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
32	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
33	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
34	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
35	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
36	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
37	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
38	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
39	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
40	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
41	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
42	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
43	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
44	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
45	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
46	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
48	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
53	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
53	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "%" discovered
109	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
110	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
112	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
198	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
200	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
201	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered

Report for node_modules/@openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol
https://dashboard.mythx.io/#/console/analyses/eb512d63-cfd1-4e4f-bf95-432fc21abefe
https://dashboard.mythx.io/#/console/analyses/fc1f83ab-a126-453c-90af-fc92861a99f7
https://dashboard.mythx.io/#/console/analyses/69722728-e9fd-4893-b41c-0ade645699c6

Line	SWC Title	Severity	Short Description
66	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
78	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
96	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered

Report for contracts/PresaleVesting/SeedPresaleCvg.sol
https://dashboard.mythx.io/#/console/analyses/fc1f83ab-a126-453c-90af-fc92861a99f7

Line	SWC Title	Severity	Short Description
12	(SWC-103) Floating Pragma	Low	A floating pragma is set.
50	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "**" discovered
50	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
51	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "**" discovered
51	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
53	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "**" discovered

53	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
54	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "**" discovered
54	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
94	(SWC-115) Authorization through tx.origin	Low	Use of "tx.origin" as a part of authorization control.
112	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
113	(SWC-110) Assert Violation	Unknown	Out of bounds array access
119	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
123	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
127	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
133	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
133	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
136	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
136	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
201	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
217	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
217	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
219	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
221	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
233	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
233	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
235	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
237	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
249	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-=" discovered
249	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
251	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
251	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
264	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
264	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-=" discovered
266	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
266	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered

Report for contracts/PresaleVesting/VestingCvg.sol
https://dashboard.mythx.io/#/console/analyses/69722728-e9fd-4893-b41c-0ade645699c6

Line	SWC Title	Severity	Short Description
12	(SWC-103) Floating Pragma	Low	A floating pragma is set.
58	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
58	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "**" discovered
59	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "**" discovered
59	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
62	(SWC-110) Assert Violation	Unknown	Public state variable with array type causing reacheable exception by default.
126	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
190	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
195	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
211	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
226	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
228	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-=" discovered
246	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
248	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
250	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-=" discovered

271	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
273	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
275	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-=" discovered
291	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
294	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
298	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
300	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-=" discovered
387	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
387	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
388	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
388	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
391	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
391	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
398	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
399	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
399	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
399	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
402	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered

Report for contracts/PresaleVesting/WlPresaleCvg.sol
https://dashboard.mythx.io/#/console/analyses/b0eadc08-b1ac-4e24-9778-d0a84c266261

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

Report for contracts/Rewards/CvgRewards.sol
https://dashboard.mythx.io/#/console/analyses/a6dc379b-129b-44ac-adc0-e56d65b59809

Line	SWC Title	Severity	Short Description
12	(SWC-103) Floating Pragma	Low	A floating pragma is set.
111	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
111	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
111	(SWC-101) Integer Overflow and Underflow	Unknown	Compiler-rewritable "<uint> - 1" discovered
111	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
113	(SWC-101) Integer Overflow and Underflow	Unknown	Compiler-rewritable "<uint> - 1" discovered
113	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
114	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
114	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "**" discovered
114	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
116	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
146	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
167	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
191	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
209	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
229	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
233	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
238	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
263	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
274	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
295	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
305	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
305	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered

312	(SWC-110) Assert Violation	Unknown	Out of bounds array access
319	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
320	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered

Report for contracts/Rewards/YsDistributor.sol
https://dashboard.mythx.io/#/console/analyses/eb512d63-cfd1-4e4f-bf95-432fc21abefe

Line	SWC Title	Severity	Short Description
12	(SWC-103) Floating Pragma	Low	A floating pragma is set.
96	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "%" discovered
97	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
98	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
98	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
104	(SWC-110) Assert Violation	Unknown	Out of bounds array access
105	(SWC-110) Assert Violation	Unknown	Out of bounds array access
107	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
115	(SWC-110) Assert Violation	Unknown	Out of bounds array access
120	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
132	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
167	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
181	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
181	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
181	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "**" discovered
221	(SWC-110) Assert Violation	Unknown	Out of bounds array access
224	(SWC-110) Assert Violation	Unknown	Out of bounds array access
252	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
267	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
267	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "**" discovered
267	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
281	(SWC-110) Assert Violation	Unknown	Out of bounds array access
282	(SWC-110) Assert Violation	Unknown	Out of bounds array access
287	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
316	(SWC-110) Assert Violation	Unknown	Out of bounds array access
316	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
319	(SWC-110) Assert Violation	Unknown	Out of bounds array access
328	(SWC-110) Assert Violation	Unknown	Out of bounds array access
330	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
334	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
339	(SWC-110) Assert Violation	Unknown	Out of bounds array access
341	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
355	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
356	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "**" discovered
356	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
356	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
359	(SWC-110) Assert Violation	Unknown	Out of bounds array access
362	(SWC-110) Assert Violation	Unknown	Out of bounds array access
364	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
383	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
383	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
383	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "**" discovered

Report for contracts/Rewards/TAssetBlackHole.sol
https://dashboard.mythx.io/#/console/analyses/226c7ed4-b0b1-4204-945b-bb9ecf746f76

Line	SWC Title	Severity	Short Description
12	(SWC-103) Floating Pragma	Low	A floating pragma is set.
92	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
93	(SWC-110) Assert Violation	Unknown	Out of bounds array access
168	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
188	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
207	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
225	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
227	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
232	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
251	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
275	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
285	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
285	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
287	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
287	(SWC-110) Assert Violation	Unknown	Out of bounds array access
298	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered

Report for contracts/Staking/TAssetStaking.sol
https://dashboard.mythx.io/#/console/analyses/0569302b-e5dc-4b63-980b-a9b5b13e3e5a

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

Report for contracts/Staking/TokeStakingCommon.sol
https://dashboard.mythx.io/#/console/analyses/da5f54a4-086c-4143-8e77-63fbc0221999
https://dashboard.mythx.io/#/console/analyses/07994b21-6a39-4f87-b12a-1d0c27f7d19d

Line	SWC Title	Severity	Short Description
12	(SWC-103) Floating Pragma	Low	A floating pragma is set.
186	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
229	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
230	(SWC-110) Assert Violation	Unknown	Out of bounds array access
234	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
253	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
254	(SWC-110) Assert Violation	Unknown	Out of bounds array access
276	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
280	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
281	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
312	(SWC-110) Assert Violation	Unknown	Out of bounds array access
315	(SWC-101) Integer Overflow and Underflow	Unknown	Compiler-rewritable "<uint> - 1" discovered
315	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
316	(SWC-110) Assert Violation	Unknown	Out of bounds array access
316	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "--" discovered
319	(SWC-110) Assert Violation	Unknown	Out of bounds array access
339	(SWC-110) Assert Violation	Unknown	Out of bounds array access
343	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
343	(SWC-101) Integer Overflow and Underflow	Unknown	Compiler-rewritable "<uint> - 1" discovered
344	(SWC-110) Assert Violation	Unknown	Out of bounds array access
345	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "--" discovered
348	(SWC-110) Assert Violation	Unknown	Out of bounds array access

364	(SWC-110) Assert Violation	Unknown	Out of bounds array access
365	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
365	(SWC-101) Integer Overflow and Underflow	Unknown	Compiler-rewritable "<uint> - 1" discovered
381	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
394	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
397	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
399	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
400	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
413	(SWC-110) Assert Violation	Unknown	Out of bounds array access
414	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
414	(SWC-101) Integer Overflow and Underflow	Unknown	Compiler-rewritable "<uint> - 1" discovered
419	(SWC-101) Integer Overflow and Underflow	Unknown	Compiler-rewritable "<uint> - 1" discovered
419	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
439	(SWC-110) Assert Violation	Unknown	Out of bounds array access
440	(SWC-101) Integer Overflow and Underflow	Unknown	Compiler-rewritable "<uint> - 1" discovered
440	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
445	(SWC-101) Integer Overflow and Underflow	Unknown	Compiler-rewritable "<uint> - 1" discovered
445	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
471	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
472	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
481	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
482	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
485	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
488	(SWC-110) Assert Violation	Unknown	Out of bounds array access
490	(SWC-101) Integer Overflow and Underflow	Unknown	Compiler-rewritable "<uint> - 1" discovered
490	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
492	(SWC-110) Assert Violation	Unknown	Out of bounds array access
494	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
494	(SWC-101) Integer Overflow and Underflow	Unknown	Compiler-rewritable "<uint> - 1" discovered
535	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-=" discovered
536	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-=" discovered
545	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-=" discovered
554	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-=" discovered
555	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-=" discovered
558	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-=" discovered
566	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-=" discovered
568	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-=" discovered
579	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-=" discovered
587	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-=" discovered
588	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-=" discovered
591	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-=" discovered
599	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-=" discovered
601	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-=" discovered
721	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
722	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
722	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
726	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
745	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
757	(SWC-110) Assert Violation	Unknown	Out of bounds array access

364	(SWC-110) Assert Violation	Unknown	Out of bounds array access
365	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
365	(SWC-101) Integer Overflow and Underflow	Unknown	Compiler-rewritable "<uint> - 1" discovered
381	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
394	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
397	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
399	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
400	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
413	(SWC-110) Assert Violation	Unknown	Out of bounds array access
414	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
414	(SWC-101) Integer Overflow and Underflow	Unknown	Compiler-rewritable "<uint> - 1" discovered
419	(SWC-101) Integer Overflow and Underflow	Unknown	Compiler-rewritable "<uint> - 1" discovered
419	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
439	(SWC-110) Assert Violation	Unknown	Out of bounds array access
440	(SWC-101) Integer Overflow and Underflow	Unknown	Compiler-rewritable "<uint> - 1" discovered
440	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
445	(SWC-101) Integer Overflow and Underflow	Unknown	Compiler-rewritable "<uint> - 1" discovered
445	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
471	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
472	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
481	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
482	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
485	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
488	(SWC-110) Assert Violation	Unknown	Out of bounds array access
490	(SWC-101) Integer Overflow and Underflow	Unknown	Compiler-rewritable "<uint> - 1" discovered
490	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
492	(SWC-110) Assert Violation	Unknown	Out of bounds array access
494	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
494	(SWC-101) Integer Overflow and Underflow	Unknown	Compiler-rewritable "<uint> - 1" discovered
535	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-=" discovered
536	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-=" discovered
545	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-=" discovered
554	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-=" discovered
555	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-=" discovered
558	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-=" discovered
566	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-=" discovered
568	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-=" discovered
579	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-=" discovered
587	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-=" discovered
588	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-=" discovered
591	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-=" discovered
599	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-=" discovered
601	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-=" discovered
721	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
722	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
722	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
726	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
745	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
757	(SWC-110) Assert Violation	Unknown	Out of bounds array access

758	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
761	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
761	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
776	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered

Report for contracts/Staking/StakingLogo.sol
https://dashboard.mythx.io/#/console/analyses/770ebed8-3f78-4918-aa1d-6707b04a413e

Line	SWC Title	Severity	Short Description
12	(SWC-103) Floating Pragma	Low	A floating pragma is set.
40	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
41	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/=" discovered
45	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-=" discovered
46	(SWC-110) Assert Violation	Unknown	Out of bounds array access
46	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
46	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "%" discovered
47	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/=" discovered
56	(SWC-110) Assert Violation	Unknown	Out of bounds array access
58	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
83	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
83	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
84	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
84	(SWC-110) Assert Violation	Unknown	Out of bounds array access
84	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
86	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
86	(SWC-110) Assert Violation	Unknown	Out of bounds array access
86	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
90	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
90	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
123	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
123	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "**" discovered
133	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
133	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "**" discovered
143	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
143	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "**" discovered

Report for contracts/Staking/TokeStaker.sol
https://dashboard.mythx.io/#/console/analyses/c3aab878-f4de-43a8-8e3c-8059564695d7

Line	SWC Title	Severity	Short Description
12	(SWC-103) Floating Pragma	Low	A floating pragma is set.
44	(SWC-110) Assert Violation	Unknown	Public state variable with array type causing reacheable exception by default.
64	(SWC-110) Assert Violation	Unknown	Out of bounds array access
65	(SWC-110) Assert Violation	Unknown	Out of bounds array access
67	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
89	(SWC-110) Assert Violation	Unknown	Out of bounds array access
127	(SWC-110) Assert Violation	Unknown	Out of bounds array access
129	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
129	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
129	(SWC-110) Assert Violation	Unknown	Out of bounds array access
130	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-=" discovered
130	(SWC-110) Assert Violation	Unknown	Out of bounds array access

133	(SWC-110) Assert Violation	Unknown	Out of bounds array access
136	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered

Report for contracts/Staking/CvgTokeStaking.sol
https://dashboard.mythx.io/#/console/analyses/07994b21-6a39-4f87-b12a-1d0c27f7d19d

Line	SWC Title	Severity	Short Description
12	(SWC-103) Floating Pragma	Low	A floating pragma is set.
43	(SWC-110) Assert Violation	Unknown	Public state variable with array type causing reacheable exception by default.
63	(SWC-110) Assert Violation	Unknown	Out of bounds array access
65	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
114	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
118	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
126	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
134	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
169	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
176	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
177	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
247	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
272	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
272	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
273	(SWC-110) Assert Violation	Unknown	Out of bounds array access
274	(SWC-110) Assert Violation	Unknown	Out of bounds array access
275	(SWC-110) Assert Violation	Unknown	Out of bounds array access
281	(SWC-110) Assert Violation	Unknown	Out of bounds array access
289	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
332	(SWC-110) Assert Violation	Unknown	Out of bounds array access
336	(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
362	(SWC-110) Assert Violation	Unknown	Out of bounds array access
364	(SWC-110) Assert Violation	Unknown	Out of bounds array access
364	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
365	(SWC-110) Assert Violation	Unknown	Out of bounds array access
366	(SWC-110) Assert Violation	Unknown	Out of bounds array access
371	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
374	(SWC-110) Assert Violation	Unknown	Out of bounds array access
380	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
386	(SWC-110) Assert Violation	Unknown	Out of bounds array access
388	(SWC-110) Assert Violation	Unknown	Out of bounds array access
389	(SWC-110) Assert Violation	Unknown	Out of bounds array access
396	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
432	(SWC-110) Assert Violation	Unknown	Out of bounds array access
432	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
433	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
433	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
433	(SWC-110) Assert Violation	Unknown	Out of bounds array access
437	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
442	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
461	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered

476	(SWC-110) Assert Violation	Unknown	Out of bounds array access
478	(SWC-110) Assert Violation	Unknown	Out of bounds array access
482	(SWC-110) Assert Violation	Unknown	Out of bounds array access
486	(SWC-110) Assert Violation	Unknown	Out of bounds array access
487	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
487	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
491	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
505	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
518	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
529	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
531	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
548	(SWC-101) Integer Overflow and Underflow	Unknown	Compiler-rewritable "<uint> - 1" discovered
548	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
552	(SWC-110) Assert Violation	Unknown	Out of bounds array access
556	(SWC-110) Assert Violation	Unknown	Out of bounds array access
557	(SWC-110) Assert Violation	Unknown	Out of bounds array access
561	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
575	(SWC-110) Assert Violation	Unknown	Out of bounds array access
577	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered

Report for contracts/Staking/StakingViewer.sol
https://dashboard.mythx.io/#/console/analyses/d2aa2c14-915f-4c9d-829d-5dca86a297d6

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

Report for node_modules/@openzeppelin/contracts/token/ERC20/ERC20.sol
https://dashboard.mythx.io/#/console/analyses/cdc44996-c2fb-492d-858b-953c552af532

Line	SWC Title	Severity	Short Description
183	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
206	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
239	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
242	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
264	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
267	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
293	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
295	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-=" discovered
345	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered

Report for contracts/Token/Cvg.sol
https://dashboard.mythx.io/#/console/analyses/cdc44996-c2fb-492d-858b-953c552af532

Line	SWC Title	Severity	Short Description
12	(SWC-103) Floating Pragma	Low	A floating pragma is set.
18	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "**" discovered
18	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
19	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "**" discovered
19	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
20	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "**" discovered
20	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
21	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "**" discovered

21	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
38	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
40	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
46	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
48	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered
54	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
56	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+=" discovered

Report for contracts/Token/CvgToke.sol
https://dashboard.mythx.io/#/console/analyses/4ddc62b6-b4c9-4a8d-b7ec-2810bc2a6ade

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

Report for contracts/CvgControlTower.sol
https://dashboard.mythx.io/#/console/analyses/222e08a8-514d-4b76-8d2f-7c09022ff5a1

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

Report for contracts/CloneFactory.sol
https://dashboard.mythx.io/#/console/analyses/791d6a44-1800-4299-b682-6bc562bf0f10

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

The above output was reviewed, and all vulnerabilities were determined to be false positives and were not included in the report.

THANK YOU FOR CHOOSING

// HALBORN