Quantstamp

Cove Finance

Executive Summary

This audit report was prepared by Quantstamp, the leader in blockchain security.

Type	DeFi
Timeline	2024-02-25 through 2024-02-29
Language	Solidity
Methods	Architecture Review, Unit Testing, Functional Testing, Computer-Aided Verification, Manual Review
Specification	Cove Boosties Docs YSD, Yearn and CoveYFI Boosties Roles Breakdown
Source Code	• Storm-Labs-Inc/cove-contracts-boosties #a7959ce
Auditors	• Julio Aguilar Auditing Engineer • Guillermo Escobero Auditing Engineer • Valerian Callens Senior Auditing Engineer

Documentation quality	High
Test quality	High
Total Findings	16 Fixed: 11 Acknowledged: 2 Mitigated: 3
High severity findings	0
Medium severity findings	1 Fixed: 1
Low severity findings	7 Fixed: 4 Acknowledged: 1 Mitigated: 2
Undetermined severity findings	0
Informational findings	8 Fixed: 6 Acknowledged: 1 Mitigated: 1

Summary of Findings

Cove is a staking platform optimizing Yearn V3 **dYFI** emissions for users by providing a protocol-owned **veYFI** boost. This eliminates the need for users to hold both **veYFI** and gauge tokens, addressing the issue of earning lower **dYFI** emissions due to imbalanced positions. Any token related to the yearn gauge, including the vault token or underlying asset, can be used to deposit into the different strategies. Users can also decide whether to auto-compound or manually redeem and swap the **dYFI** emissions.

We commend the Cove team for the high quality of the codebase, tests, and documentation. The audit team managed to find 16 issues including 1 Medium-severity issue. We recommend the Cove team address the findings as soon as possible and before any deployments.

Update: The Cove team addressed all issues by either fixing or acknowledging them. They also added more test cases that cover the fixed issues, improving the test suite even more.

ID	DESCRIPTION	SEVERITY	STATUS
COVE-1	**Unsafe Management of Variable Types Can Negatively Impact the Reward Distribution Mechanism of `MiniChefV3`**	Medium	Fixed
COVE-2	**Updating the Wrong `PoolInfo` Could Break the Contract's Accounting**	Low	Fixed
COVE-3	**Missing Input Validation**	Low	Fixed
COVE-4	**Ownership Can Be Renounced**	Low	Mitigated
COVE-5	**Mismatch Between the `NatSpec` and the Implementation of `BaseRewardsGauge.depositRewardToken()`'s Access Control.**	Low	Fixed

ID	DESCRIPTION	SEVERITY	STATUS
COVE-6	Risks of Supporting Non-Standard `ERC20` Tokens	Low	Acknowledged
COVE-7	Dust of Reward Tokens Can Remain Forever in the Contract `BaseRewardsGauge`	Low	Fixed
COVE-8	Race Condition Exists Between the Function `forwardRewardToken()` and Privileged Operations	Low	Mitigated
COVE-9	Issues Related to the Deployment Script	Informational	Mitigated
COVE-10	The Function `GetAllGaugeInfo()` May Result in an Out-of-Gas Error if Its Size Becomes Too Large	Informational	Fixed
COVE-11	Whitelisting Events for Transfer Limitations Can Be Ambiguous for External Observers	Informational	Fixed
COVE-12	The Whitelisting Mechanism for Cove Transfers when the Token Is Paused Is Unclear	Informational	Fixed
COVE-13	Application Monitoring Can Be Improved by Emitting More Events	Informational	Fixed
COVE-14	Upgradability	Informational	Acknowledged
COVE-15	Unlocked Pragma	Informational	Fixed
COVE-16	Best Practices	Informational	Fixed

Assessment Breakdown

Quantstamp's objective was to evaluate the files in scope for security-related issues, code quality, and adherence to specifications and best practices.

Disclaimer

Only features that are contained within the repositories at the commit hashes specified on the front page of the report are within the scope of the audit and fix review. All features added in future revisions of the code are excluded from consideration in this report.

Possible issues we looked for included (but are not limited to):

- Transaction-ordering dependence
- Timestamp dependence
- Mishandled exceptions and call stack limits
- Unsafe external calls
- Integer overflow / underflow
- Number rounding errors
- Reentrancy and cross-function vulnerabilities
- Denial of service / logical oversights
- Access control
- Centralization of power
- Business logic contradicting the specification
- Code clones, functionality duplication
- Gas usage
- Arbitrary token minting

Methodology

1. Code review that includes the following
    1. Review of the specifications, sources, and instructions provided to Quantstamp to make sure we understand the size, scope, and functionality of the smart contract.
    2. Manual review of code, which is the process of reading source code line-by-line in an attempt to identify potential vulnerabilities.

3. Comparison to specification, which is the process of checking whether the code does what the specifications, sources, and instructions provided to Quantstamp describe.
2. Testing and automated analysis that includes the following:
    1. Test coverage analysis, which is the process of determining whether the test cases are actually covering the code and how much code is exercised when we run those test cases.
    2. Symbolic execution, which is analyzing a program to determine what inputs cause each part of a program to execute.
3. Best practices review, which is a review of the smart contracts to improve efficiency, effectiveness, clarity, maintainability, security, and control based on the established industry and academic practices, recommendations, and research.
4. Specific, itemized, and actionable recommendations to help you take steps to secure your smart contracts.

Scope

Files Included

Repo: https://github.com/Storm-Labs-Inc/cove-contracts-boosties(a7959ce)

Files:

- src/Yearn4626RouterExt.sol
- src/governance/CoveToken.sol
- src/registries/CoveYearnGaugeFactory.sol
- src/rewards/BaseRewardsGauge.sol
- src/rewards/YSDRewardsGauge.sol
- src/rewards/RewardForwarder.sol
- src/rewards/MiniChefV3.sol
- script/deployment.s.sol

Files Excluded

Repo: https://github.com/Storm-Labs-Inc/cove-contracts-boosties(a7959ce)

Files: Everything else

Findings

COVE-1

Unsafe Management of Variable Types Can Negatively Impact the Reward Distribution Mechanism of `MiniChefV3`

Medium | Fixed

Update

The parameter's data type was changed to `uint64`, and the SafeCast library is now used for casting.

Update

Marked as "Fixed" by the client. Addressed in: `86d4300ef645cfb34e4d2062a366a709e775920e`.

File(s) affected: `MiniChefV3.sol`

Description: In `MiniChefV3`, addresses with `DEFAULT_ADMIN_ROLE` can add or update LP tokens, specifying their reward distribution proportion through `allocPoint`. Although `allocPoint` is cast to `uint64` in `PoolInfo`, its original `uint256` value directly increments `totalAllocPoint`. If `allocPoint` exceeds the `uint64` limits, it irreversibly inflates `totalAllocPoint`, potentially disrupting reward distribution by creating disproportionate allocations among LP tokens.

Additionally, a similar issue is found in the `_updatePool()` function which updates the corresponding pool's `accRewardPerShare` by downcasting from `uint256` to `uint128` which, in case of a truncation, could break the accounting of the contract.

Recommendation: Consider using the parameter `uint64 allocPoint` in the functions `add()` and `set()`. Additionally, consider using `SafeCast` from `OpenZeppelin`, which reverts for unsafe explicit castings.

COVE-2

Updating the Wrong `PoolInfo` Could Break the Contract's Accounting

Low | Fixed

Update

Added the validation as recommended.

Update

Marked as "Fixed" by the client. Addressed in: `6d8e565e531f08734909340504556f021e32e9d9` .

File(s) affected: `MiniChefV3.sol`

Description: The function `set()` updates the `allocPoint` of the pool corresponding to the provided `pid` . Since the value of `pid` is not validated, it would be possible to update the wrong pool info. In this case, it could result in incorrect allocation proportions.

Recommendation: Consider adding the `lpToken` as a new input to validate that the `pid` is correct.

COVE-3 Missing Input Validation

Low Fixed

Update

All issues were fixed or mitigated.

Update

Marked as "Fixed" by the client. Addressed in: `31d135e755a373a33b3ca6075478d73c7b2b8305` . The client provided the following explanation:

> https://github.com/Storm-Labs-Inc/cove-contracts-boosties/issues/222#issuecomment-1979707152

File(s) affected: `CoveToken.sol, CoveYearnGaugeFactory.sol, MiniChefV3.sol, RewardsForwarder.sol`

Related Issue(s): SWC-123

Description: It is important to validate inputs, even if they only come from trusted addresses, to avoid human error. The following is a non-exhaustive list of potential missing validation checks:

- In `CoveToken` :
  1. In the `constructor()` , there is no minimum or maximum value check for the value of `mintingAllowedAfter_` . It can be set to directly after or after a very long time from deployment.
- In `CoveYearnGaugeFactory` :
  1. In the three functions used to update the current implementation contracts stored by the contract, there is no check to make sure that the new address is a contract.
- In `MiniChefV3` :
  1. The constructor should validate the address of both inputs against `address(0)` . A mistake in this case could lead to an adminless contract or the `REWARD_TOKEN` being invalid resulting in a re-deployment.
  2. The function `add()` should validate the first two inputs:
     - Evaluate if `allocPoint` can be zero or `type(uint256).max` , and what the impact could be. Otherwise, consider adding and checking lower and upper limits.
     - The `lpToken_` should be validated against `address(0)` as well.
  3. Evaluate if the `rewardPerSecond` can be zero or `type(uint256).max` , and consider adding lower and upper bounds to the function `setRewardPerSecond()` when updating the variable.
  4. In the functions `deposit()` and `withdraw()` , it is possible to use an amount equal to zero.
- In `RewardsForwarder` :
  1. The `_setTreasuryBps()` function correctly validates that the new treasury bps is lower than the possible max. However, consider if this value should be validated against an upper bound a lot lower than the max. At the moment, the value can be the max allowing the treasury to take all the rewards leaving none for the corresponding gauge.

Recommendation: We recommend adding the relevant checks.

COVE-4 Ownership Can Be Renounced

Low Mitigated

Update

Marked as "Mitigated" by the client. Addressed in: `77db27f92a074cf96ded179865bab9d2b572698d` . The client provided the following explanation:

> We want to stick to using the vanilla OZ libraries as much as possible. We have replaced AccessControl with AccessControlEnumerable so we can add validation for any transactions with role related changes.

Description: If the admin renounces their role, the contract will be left without an admin. Consequently, any function guarded by the `onlyRole` modifier will no longer be able to be executed.

Recommendation: Confirm that this is the intended behavior for each of the contracts. If not, override the `renounceRole()` function in the affected contracts to not allow the last role owner to renounce it.

COVE-5
Mismatch Between the `NatSpec` and the Implementation of `BaseRewardsGauge.depositRewardToken()`'s Access Control.

Low | Fixed

> **Update**
> The missing requirement was implemented.

> **Update**
> Marked as "Fixed" by the client. Addressed in: `d8d5d0e37769e77ae2a422e4a81e6d87d03d850a` .

File(s) affected: `BaseRewardsGauge.sol`

Description: The function `depositRewardToken()` verifies that `msg.sender` is registered as the distributor of the reward token in the `rewardData` mapping. However, the NatSpec comment on the function states: "Only the distributor or an address with the manager role can call this."

Recommendation: Review the logic and confirm the intended behavior.

COVE-6 Risks of Supporting Non-Standard `ERC20` Tokens

Low | Acknowledged

> **Update**
> Marked as "Acknowledged" by the client. The client provided the following explanation:
>
> > We don't intend to use rebasing or fee on transfer tokens.

Description: Supporting tokens with specific features such as fees, rebasing, pausable, upgradeable, blacklist-able, or hooks on transfers could negatively impact the main flows of the system if no specific mitigation measure is enforced to limit the consequences.

For instance, using tokens where the amount requested to be transferred is not the same as the one received (usually when it has fees on transfers) could negatively impact the functions using `transfer()` or `transferFrom()`, like `BaseRewardsGauge.depositRewardToken()`.

Recommendation: Consider analyzing thoroughly from a security perspective any token you would like to support. Specifically, additional checks should be added if rebasing and fees on transfer tokens are to be supported.

COVE-7
Dust of Reward Tokens Can Remain Forever in the Contract

`BaseRewardsGauge`

Low | Fixed

> **Update**
> The related data structure now stores the `leftOver` which is updated when depositing the reward token.

> **Update**
> Marked as "Fixed" by the client. Addressed in: `c9ff7b71d8b8dc0f49dc69bb8d750e452fa2e6c2` .

Description: Reward tokens can be sent to the `BaseRewardsGauge` contract via the function `depositRewardToken()`. The amount passed as a parameter is transferred to the contract. Then, a rate per second is computed by dividing the amount by the number of seconds in one week (604800). In the worst case where `amount % 604800 == 604800 - 1`, applying the consecutive rates will let `604800 - 1` tokens in the contract. For USDC which has 6 decimals, it means 0.6 USD, but this could accumulate if the same amount is deposited multiple times. Handling this dust could be done by adding it to the `leftOver` amount calculated in the else block of the function.

Recommendation: Consider if managing the dust of reward tokens could be worth it. If yes, consider updating the code accordingly.

COVE-8

Race Condition Exists Between the Function `forwardRewardToken()` and Privileged Operations

Low | Mitigated

Update

When the function `setTreasuryBPS()` is called, reward tokens are automatically forwarded before the operation. However, updating the treasury via `setTreasury()` is still exposed to race conditions.

Update

Marked as "Fixed" by the client. Addressed in: `7d8596b84138e104ea3e9d534f2c44a6d0f7882b` .

File(s) affected: `RewardForwarder.sol`

Description: In the contract `RewardForwarder` , addresses with the role `DEFAULT_ADMIN_ROLE` can update the treasury address that will receive a portion of the rewards via the function `setTreasury()` , as well as the proportion of these rewards via the function `setTreasuryBps()` . In parallel, anyone can trigger the function `forwardRewardToken()` , which will perform this transfer of a given proportion of the rewards to the treasury address. These three functions can have different outcomes depending on whether one is executed before or after the other.

Recommendation: When performing privileged actions related to the treasury and its proportion of rewards, make sure that the treasury receives the correct amount of rewards.

COVE-9 Issues Related to the Deployment Script

Informational | Mitigated

Update

The Cove team stated that some points will be addressed before deployment.

Update

Marked as "Fixed" by the client. Addressed in: `007b7d18ae33f3fb7d0c84cf425ecc6a9e3c26d4` .

File(s) affected: `Deployments.s.sol`

Description: Some unexpected statements were found in the deployment script which can be inconsistent with the specification and should be reviewed by the team:

1. The `deploy()` function should add a check to validate the admin value in some way when deploying to mainnet. Otherwise, if the `ADMIN_MULTISIG` is missing, a re-deployment of the full system could be required.
2. In the function `deployYearnStakingDelegateStack()` , there is an unlikely possibility that the value returned by the function `DEFAULT_ADMIN_ROLE()` differs for the contracts `YearnStakingDelegate` and `SwapAndLock` . As a result, the following lines:

```
SwapAndLock(swapAndLock).grantRole(ysd.DEFAULT_ADMIN_ROLE(), admin);
SwapAndLock(swapAndLock).renounceRole(ysd.DEFAULT_ADMIN_ROLE(), broadcaster);
```

could be replaced by:

```
SwapAndLock(swapAndLock).grantRole(swapAndLock.DEFAULT_ADMIN_ROLE(), admin);
SwapAndLock(swapAndLock).renounceRole(swapAndLock.DEFAULT_ADMIN_ROLE(), broadcaster);
```

3. The initial roles that should be assigned to the different deployed contracts were not completely finalized before the audit, so some role assignments were confirmed to be incorrect like for the contracts `CoveToken` and `MiniChefV3` .
4. The following line was confirmed to be incorrect: `ysd.setSnapshotDelegate("veyfi.eth", treasury);` .
5. The function `deployMiniChefV3()` does not transfer the ownership of the `MiniChefV3` contract to the admin.
6. Even though each call executed in the `multicall()` should revert in failure, it would be a best practice to check the results in `allowlistCoveTokenTransfers()` and `registerContractsInMasterRegistry()` in case of an undetected failure.
7. The function `deploySablierStreams()` relies on `vesting.json` which currently contains dummy values.

Recommendation: Review, confirm, and add checks that allow the intended behavior.

COVE-10

The Function `GetAllGaugeInfo()` May Result in an Out-of-Gas Error if Its Size Becomes Too Large

Informational | Fixed

Update

The team added a starting offset and an iteration limit to loop over the gauges.

Update

Marked as "Fixed" by the client. Addressed in: `a93b3cbab2b45ace7a9c2aa7f5072e604d3335ae` .

File(s) affected: `CoveYearnGaugeFactory.sol`

Description: The function `getAllGaugeInfo()` returns a list containing information about all the gauges supported by the contract. However, in the unlikely scenario of the list `supportedYearnGauges` reaching a significant size, the function `getAllGaugeInfo()` may result in an out-of-gas error. This is problematic because the size of this list cannot be reduced once increased. This could negatively impact integrated third parties.

Recommendation: Consider adding to this view function two parameters `idBegin` and `idEnd` to make it possible to return information about a subset of elements from the list `supportedYearnGauges` .

COVE-11

Whitelisting Events for Transfer Limitations Can Be Ambiguous for External Observers

Informational Fixed

Update

Fixed as recommended.

Update

Marked as "Fixed" by the client. Addressed in: `acb722359d000995a778165fab1c3cf0c676dea2` .

File(s) affected: `CoveToken.sol`

Description: Four events can be emitted by the contract CoveToken when addresses are whitelisted or not for the actions of sending and receiving the token:

```
/// @dev Emitted when a transferrer is allowed.
event TransferrerAllowed(address indexed target);
/// @dev Emitted when a transferrer is disallowed.
event TransferrerDisallowed(address indexed target);
/// @dev Emitted when a transferee is allowed.
event TransfereeAllowed(address indexed target);
/// @dev Emitted when a transferee is disallowed.
event TransfereeDisallowed(address indexed target);
```

In the unlikely scenario of performing multiple opposite operations (whitelist/blacklist) for the same address within the same block or transaction, external observers may not be able to know if the address is currently whitelisted, or not, because sorting these events by timestamp will result in an ambiguous ordering.

Recommendation: Consider adding a field `++eventId` to these events to make them orderable for external observers.

COVE-12

The Whitelisting Mechanism for Cove Transfers when the Token Is Paused Is Unclear

Informational Fixed

Update

Marked as "Fixed" by the client. Addressed in: `0127a7d0beac7ea9c2e853a64952ca20486d51f3` . The client provided the following explanation:

> We will also update the user facing documentation.

File(s) affected: `CoveToken.sol`

Description: In the `CoveToken`, there is a whitelist for senders and another whitelist for receivers that is only active when the contract is paused. One assumption regarding its functionality could be that while the contract is paused only whitelisted senders can transfer to whitelisted receivers. However, that is not the case according to the implementation in the transfer hook `_beforeTokenTransfer()` which allows a whitelisted sender to transfer tokens to a not-whitelisted receiver, and a not-whitelisted sender can transfer to a whitelisted receiver.

Recommendation: Confirm that this is the desired behavior and update the technical and user-facing documentation.

COVE-13
Application Monitoring Can Be Improved by Emitting More Events

Informational | Fixed

> **Update**
> More monitoring events were added to the system.

> **Update**
> Marked as "Fixed" by the client. Addressed in: `d2d6feaf8c1e4ea32383ae58d98329fdb449713b`.

File(s) affected: `BaseRewardsGauge.sol`

Description: To validate the proper deployment and initialization of the contracts, it is a good practice to emit events. Also, any important state transitions can be logged, which is beneficial for monitoring the contract, and also tracking eventual bugs or hacks. The `BaseRewardsGauge` contract could benefit from emitting an event when a new distributor is set for a given `rewardToken` in the function `setRewardDistributor()`.

Recommendation: Consider emitting the event.

COVE-14 Upgradability

Informational | Acknowledged

File(s) affected: `RewardForwarder.sol, YSDRewardsGauge.sol, BaseRewardsGauge.sol`

Description: While upgradability is not a vulnerability in itself, users should be aware that some contracts can be upgraded at any given time. This audit does not guarantee the behavior of future contracts that the token may be upgraded to. In this case, this includes the `RewardForwarder`, `YSDRewardsGauge`, and `BaseRewardsGauge`. Additionally, there are other upgradeable contracts in the codebase that out outside the scope of this audit.

Recommendation: The fact that the contract can be upgraded and reasons for future upgrades should be communicated to users beforehand.

COVE-15 Unlocked Pragma

Informational | Fixed

> **Update**
> Fixed as recommended.

> **Update**
> Marked as "Fixed" by the client. Addressed in: `24c8394e02b93cd9a527dfdd120775813e448bfd`.

File(s) affected: `CoveToken.sol, CoveYearnGaugeFactory.sol, BaseRewardsGauge.sol, YSDRewardsGauge.sol, RewardForwarder.sol, MiniChefV3.sol, deployment.s.sol`

Related Issue(s): SWC-103

Description: Every Solidity file specifies in the header a version number of the format `pragma solidity (^)0.8.*`. The caret (`^`) before the version number implies an unlocked pragma, meaning that the compiler will use the specified version *and above*, hence the term "unlocked".

Recommendation: For consistency and to prevent unexpected behavior in the future, we recommend removing the caret to lock the file onto a specific Solidity version.

COVE-16 Best Practices

Informational | Fixed

> **Update**

All issues were addressed.

Update

Marked as "Fixed" by the client. Addressed in: `e0d25151016f57da3df3374e39f1ce9eda56710a`.

Description: The following is a non-exhaustive list of minor best practices and gas optimizations:

- In `MiniChefV3.sol`:
  1. The NatSpecs of the structs `UserInfo` and `PoolInfo` are incomplete.
- In `RewardForwarder.sol`:
  1. In the function `forwardRewardToken()`, the line `uint256 treasuryAmount = balance * treasuryBps[rewardToken] / _MAX_BPS;` can be moved inside the if condition to save gas when `balance > 0`.
- In `BaseRewardsGauge.sol`:
  1. In the function `_updateReward()`, consider clarifying the long arithmetic expression by adding parenthesis.
  2. In the function `claimableReward()`, consider caching in a memory variable the object `rewardData[rewardToken]` to save gas.
  3. In the function `depositRewardToken()`, consider replacing the expression `if (newRate <= 0)` with if `(newRate == 0)` for more clarity, and to save gas.

Recommendation: Consider the recommendations mentioned above.

Definitions

- **High severity** – High-severity issues usually put a large number of users' sensitive information at risk, or are reasonably likely to lead to catastrophic impact for client's reputation or serious financial implications for client and users.
- **Medium severity** – Medium-severity issues tend to put a subset of users' sensitive information at risk, would be detrimental for the client's reputation if exploited, or are reasonably likely to lead to moderate financial impact.
- **Low severity** – The risk is relatively small and could not be exploited on a recurring basis, or is a risk that the client has indicated is low impact in view of the client's business circumstances.
- **Informational** – The issue does not post an immediate risk, but is relevant to security best practices or Defence in Depth.
- **Undetermined** – The impact of the issue is uncertain.
- **Fixed** – Adjusted program implementation, requirements or constraints to eliminate the risk.
- **Mitigated** – Implemented actions to minimize the impact or likelihood of the risk.
- **Acknowledged** – The issue remains in the code but is a result of an intentional business or design decision. As such, it is supposed to be addressed outside the programmatic means, such as: 1) comments, documentation, README, FAQ; 2) business processes; 3) analyses showing that the issue shall have no negative consequences in practice (e.g., gas analysis, deployment settings).

Appendix

File Signatures

The following are the SHA-256 hashes of the reviewed files. A file with a different SHA-256 hash has been modified, intentionally or otherwise, after the security review. You are cautioned that a different SHA-256 hash could be (but is not necessarily) an indication of a changed condition or potential vulnerability that was not within the scope of the review.

Contracts

- `c8a...c03 ./Deployments.s.sol`
- `b05...f3e ./IYearn4626RouterExt.sol`
- `6f9...e1a ./src/Yearn4626RouterExt.sol`
- `664...339 ./src/interfaces/rewards/IBaseRewardsGauge.sol`
- `666...2c3 ./src/interfaces/rewards/IMiniChefV3Rewarder.sol`
- `6d3...9d8 ./src/governance/CoveToken.sol`
- `3ee...edc ./src/libraries/Errors.sol`
- `4db...61c ./src/rewards/MiniChefV3.sol`
- `c48...169 ./src/rewards/BaseRewardsGauge.sol`
- `5e9...313 ./src/rewards/RewardForwarder.sol`
- `c43...0b2 ./src/rewards/YSDRewardsGauge.sol`
- `f1a...bf4 ./src/registries/CoveYearnGaugeFactory.sol`

Tests

- d55...3f0 ./test/forked/YearnGaugeStrategy.t.sol
- 1ab...b2e ./test/forked/CoveYFI.t.sol
- d73...cd7 ./test/forked/DYFIRedeemer.t.sol
- fdd...505 ./test/forked/YearnStakingDelegate.t.sol
- 6ad...3ad ./test/forked/SwapAndLock.t.sol
- 9cf...2d2 ./test/forked/Router.t.sol
- b3c...258 ./test/forked/swappers/CurveRouterSwapper.t.sol
- 761...415 ./test/utils/CurveSwapParamsConstants.sol
- 107...f10 ./test/utils/BaseTest.t.sol
- f52...1b4 ./test/utils/YearnV3BaseTest.t.sol
- 011...4c7 ./test/utils/Constants.sol
- a55...558 ./test/utils/VyperDeployer.sol
- cce...f28 ./test/integration/Integration.t.sol
- 427...81e ./test/invariant/YSDRewardsGauge.t.sol
- ce7...8c7 ./test/invariant/ERC20RewardsGauge.t.sol
- cc9...559 ./test/unit/GaugeRewardReceiver.t.sol
- 95a...2d1 ./test/unit/RewardForwarder.t.sol
- 399...538 ./test/unit/Rescuable.t.sol
- 7cb...d43 ./test/unit/StakingDelegateRewards.t.sol
- 71f...e53 ./test/unit/YearnStakingDelegate.t.sol
- e3c...cba ./test/unit/SwapAndLock.t.sol
- 887...19d ./test/unit/MasterRegistry.t.sol
- 603...4c0 ./test/unit/governance/CoveToken.t.sol
- abe...f9f ./test/unit/rewards/MiniChefV3.t.sol
- ffe...285 ./test/unit/rewards/YSDRewardsGauge.t.sol
- a97...b73 ./test/unit/rewards/ERC20RewardsGauge.t.sol
- 853...802 ./test/unit/strategies/YearnGaugeStrategy.t.sol
- 8d0...a17 ./test/unit/registries/CoveYearnGaugeFactory.t.sol
- d81...16d ./test/mocks/MockRescuable.sol
- 340...3a6 ./test/mocks/MockFlashLoanProvider.sol
- 1c2...235 ./test/mocks/MockNonPayable.sol
- dd8...a34 ./test/mocks/MockDYFIRedeemer.sol
- c6a...d82 ./test/mocks/MockCurveRouterSwapper.sol
- 807...24c ./test/mocks/MockCurveTwoAssetPool.sol
- 42f...631 ./test/mocks/MockYearnGaugeStrategy.sol
- 9a1...3f6 ./test/mocks/MockCurveRouter.sol
- 7ea...e20 ./test/mocks/MockChainLinkOracle.sol
- a8c...389 ./test/mocks/MockERC20RewardsGauge.sol
- ed9...275 ./test/mocks/MockStakingDelegateRewards.sol
- b64...a34 ./test/mocks/MockStrategy.sol
- bfe...ac2 ./test/mocks/MockRewardPool.sol
- c84...47d ./test/mocks/MockYearnStakingDelegate.sol
- 1db...622 ./test/mocks/MockRedemption.sol
- 551...189 ./test/mocks/MockGaugeRewardReceiver.sol
- a5f...7d3 ./test/mocks/MockGauge.sol
- 54e...0f9 ./test/mocks/MockVotingYFI.sol
- f1b...aa5 ./test/mocks/MockTarget.sol

Toolset

The notes below outline the setup and steps performed in the process of this audit.

Setup

Tool Setup:

- Slither v0.10.0

Steps taken to run the tools:

1. Install the Slither tool: `pip3 install slither-analyzer`
2. Run Slither from the project directory: `slither .`

Automated Analysis

Slither

All the Slither results were either identified as false positives or included in the findings of this report.

Test Suite Results

`pnpm test`

```
Ran 27 tests for test/unit/governance/CoveToken.t.sol:CoveToken_Test
 [32m[PASS] [0m testFuzz_addAllowedReceiver(uint256) (runs: 256, μ: 190986, ~: 192798)
 [32m[PASS] [0m testFuzz_addAllowedReceiver_revertWhen_CallerIsNotTimelock(address) (runs: 256, μ: 94606,
~: 94681)
 [32m[PASS] [0m testFuzz_addAllowedReceiver_revertWhen_CannotBeBothSenderAndReceiver(address) (runs: 256,
μ: 93586, ~: 93722)
 [32m[PASS] [0m testFuzz_addAllowedSender(address,uint256) (runs: 256, μ: 189765, ~: 191253)
 [32m[PASS] [0m testFuzz_addAllowedSender_revertWhen_CallerIsNotTimelock(address) (runs: 256, μ: 94573, ~:
94647)
 [32m[PASS] [0m testFuzz_addAllowedSender_revertWhen_CannotBeBothSenderAndReceiver(address) (runs: 256, μ:
93557, ~: 93713)
 [32m[PASS] [0m testFuzz_mint(uint256) (runs: 256, μ: 90045, ~: 90539)
 [32m[PASS] [0m testFuzz_mint_revertWhen_inflationTooEarly(uint256) (runs: 256, μ: 43375, ~: 43272)
 [32m[PASS] [0m testFuzz_mint_revertWhen_inflationTooLarge(uint256) (runs: 256, μ: 52430, ~: 52150)
 [32m[PASS] [0m testFuzz_mint_revertWhen_notMinter(uint256) (runs: 256, μ: 104659, ~: 104380)
 [32m[PASS] [0m testFuzz_removeAllowedReceiver(address,uint256) (runs: 256, μ: 194655, ~: 194832)
 [32m[PASS] [0m testFuzz_removeAllowedSender(address,uint256) (runs: 256, μ: 194374, ~: 194791)
 [32m[PASS] [0m testFuzz_removeFromAllowedReceiver_revertWhen_CallerIsNotTimelock(address) (runs: 256, μ:
150789, ~: 150942)
 [32m[PASS] [0m testFuzz_removeFromAllowedSender_revertWhen_CallerIsNotTimelock(address) (runs: 256, μ:
150783, ~: 150943)
 [32m[PASS] [0m testFuzz_transfer_revertWhen_notAllowedReceiver(uint256) (runs: 256, μ: 110120, ~: 110730)
 [32m[PASS] [0m testFuzz_transfer_revertWhen_notAllowedSender(address,uint256) (runs: 256, μ: 170287, ~:
170886)
 [32m[PASS] [0m test_anyoneCanUnpauseAfter() (gas: 7817)
 [32m[PASS] [0m test_availableSupplyToMint() (gas: 104404)
 [32m[PASS] [0m test_events_eventIdIncrements() (gas: 185466)
 [32m[PASS] [0m test_grantRole_TimelockRole_revertWhen_CallerIsNotTimelock() (gas: 200976)
 [32m[PASS] [0m test_initialize() (gas: 27429)
 [32m[PASS] [0m test_ownerCanUnpauseAfter() (gas: 7836)
 [32m[PASS] [0m test_unpause_anyone() (gas: 32934)
 [32m[PASS] [0m test_unpause_anyoneCanTransfer(address,address,uint256) (runs: 256, μ: 137051, ~: 138706)
 [32m[PASS] [0m test_unpause_owner() (gas: 33013)
 [32m[PASS] [0m test_unpause_revertWhen_notAdmin() (gas: 37777)
 [32m[PASS] [0m test_unpause_revertWhen_tooEarly() (gas: 67117)
Suite result:  [32mok [0m.  [32m27 [0m passed;  [31m0 [0m failed;  [33m0 [0m skipped; finished in 1.43s
(1.41s CPU time)

Ran 7 tests for test/forked/CoveYFI.t.sol:CoveYFI_ForkedTest
 [32m[PASS] [0m testFuzz_constructor(address) (runs: 256, μ: 36637, ~: 36637)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS] [0m test_deposit() (gas: 662753)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS] [0m test_deposit_passWhen_ReceiverIsGiven() (gas: 663190)
Logs:
```

```
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS] [0m test_deposit_passWhen_ReceiverIsZero() (gas: 663010)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS] [0m test_deposit_revertsOnZero() (gas: 32980)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS] [0m test_rescue() (gas: 45133)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS] [0m test_rescue_revertsOnNonOwner() (gas: 96623)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

Suite result:  [32mok [0m.  [32m7 [0m passed;  [31m0 [0m failed;  [33m0 [0m skipped; finished in 743.48ms
(52.67ms CPU time)

Ran 29 tests for test/unit/registries/CoveYearnGaugeFactory.t.sol:CoveYearnGaugeFactory_Test
 [32m[PASS] [0m testFuzz_getAllGaugeInfo(uint256,uint256) (runs: 256, μ: 1698125, ~: 1697780)
 [32m[PASS] [0m test_constructor() (gas: 54450)
 [32m[PASS] [0m test_deployCoveGauges() (gas: 3515559)
 [32m[PASS] [0m test_deployCoveGauges_revertWhen_GaugeAlreadyDeployed() (gas: 1728674)
 [32m[PASS] [0m test_deployCoveGauges_revertWhen_ZeroAddress() (gas: 32233)
 [32m[PASS] [0m test_deployCoveGauges_revertWhen_notManager() (gas: 124144)
 [32m[PASS] [0m test_getAllGaugeInfo() (gas: 3380619)
 [32m[PASS] [0m test_getGaugeInfo_revertWhen_GaugeNotDeployed(address) (runs: 256, μ: 1710392, ~: 1710392)
 [32m[PASS] [0m test_setBaseRewardsGaugeImplementation() (gas: 3644155)
 [32m[PASS] [0m test_setBaseRewardsGaugeImplementation_revertWhen_ZeroAddress() (gas: 32274)
 [32m[PASS] [0m test_setBaseRewardsGaugeImplementation_revertWhen_notAdmin() (gas: 3727780)
 [32m[PASS] [0m test_setBaseRewardsGaugeImplementation_revertWhen_notContract() (gas: 32385)
 [32m[PASS] [0m test_setGaugeAdmin() (gas: 63318)
 [32m[PASS] [0m test_setGaugeAdmin_revertWhen_ZeroAddress() (gas: 32325)
 [32m[PASS] [0m test_setGaugeManager() (gas: 63273)
 [32m[PASS] [0m test_setGaugeManager_revertWhen_ZeroAddress() (gas: 32239)
 [32m[PASS] [0m test_setGaugePauser() (gas: 63153)
 [32m[PASS] [0m test_setGaugePauser_revertWhen_ZeroAddress() (gas: 32282)
 [32m[PASS] [0m test_setRewardForwarderImplementation() (gas: 1290469)
 [32m[PASS] [0m test_setRewardForwarderImplementation_revertWhen_ZeroAddress() (gas: 32285)
 [32m[PASS] [0m test_setRewardForwarderImplementation_revertWhen_notAdmin() (gas: 1374107)
 [32m[PASS] [0m test_setRewardForwarderImplementation_revertWhen_notContract() (gas: 32430)
 [32m[PASS] [0m test_setTreasuryMultisig() (gas: 63255)
 [32m[PASS] [0m test_setTreasuryMultisig_revertWhen_ZeroAddress() (gas: 32283)
 [32m[PASS] [0m test_setTreasuryMultisig_revertWhen_notAdmin() (gas: 147135)
 [32m[PASS] [0m test_setYsdRewardsGaugeImplementation() (gas: 3970734)
 [32m[PASS] [0m test_setYsdRewardsGaugeImplementation_revertWhen_ZeroAddress() (gas: 32216)
 [32m[PASS] [0m test_setYsdRewardsGaugeImplementation_revertWhen_notAdmin() (gas: 4054360)
 [32m[PASS] [0m test_setYsdRewardsGaugeImplementation_revertWhen_notContract() (gas: 32405)
Suite result:  [32mok [0m.  [32m29 [0m passed;  [31m0 [0m failed;  [33m0 [0m skipped; finished in 475.07ms
(464.56ms CPU time)

Ran 20 tests for test/forked/swappers/CurveRouterSwapper.t.sol:CurveRouterSwapperTest
 [32m[PASS] [0m testFuzz_constructor_revertWhen_ZeroAddress() (gas: 92493)
Logs:
  Started fork  mainnet  at block  19309223
  with id 0

 [32m[PASS] [0m test_approveTokenForSwap() (gas: 111373)
Logs:
  Started fork  mainnet  at block  19309223
  with id 0

 [32m[PASS] [0m test_swap() (gas: 242482)
Logs:
```

```
  Started fork  mainnet  at block  19309223
  with id 0

 [32m[PASS][0m test_swap_MainnetWethYethGaugeCurveSwapParams() (gas: 717566)
Logs:
  Started fork  mainnet  at block  19309223
  with id 0

 [32m[PASS][0m test_swap_mainnetCrvYcrvPoolGaugeCurveSwapParams() (gas: 852880)
Logs:
  Started fork  mainnet  at block  19309223
  with id 0

 [32m[PASS][0m test_swap_mainnetDyfiEthGaugeCurveSwapParams() (gas: 874854)
Logs:
  Started fork  mainnet  at block  19309223
  with id 0

 [32m[PASS][0m test_swap_mainnetEthYfiGaugeCurveSwapParams() (gas: 601628)
Logs:
  Started fork  mainnet  at block  19309223
  with id 0

 [32m[PASS][0m test_swap_mainnetPrismaYprismaPoolGaugeCurveSwapParams() (gas: 916001)
Logs:
  Started fork  mainnet  at block  19309223
  with id 0

 [32m[PASS][0m test_swap_passWhenSwappingThroughStableAndCrypto() (gas: 417841)
Logs:
  Started fork  mainnet  at block  19309223
  with id 0

 [32m[PASS][0m test_swap_passWhenUsingEthSwapThreeTimes() (gas: 626786)
Logs:
  Started fork  mainnet  at block  19309223
  with id 0

 [32m[PASS][0m test_swap_passWhenUsingUSDT() (gas: 271131)
Logs:
  Started fork  mainnet  at block  19309223
  with id 0

 [32m[PASS][0m test_swap_revertWhen_ExpectedNotReached() (gas: 295820)
Logs:
  Started fork  mainnet  at block  19309223
  with id 0

 [32m[PASS][0m test_swap_revertWhen_IndexOutOfRange() (gas: 212146)
Logs:
  Started fork  mainnet  at block  19309223
  with id 0

 [32m[PASS][0m test_swap_revertWhen_UsingInvalidSwapParams() (gas: 617836)
Logs:
  Started fork  mainnet  at block  19309223
  with id 0

 [32m[PASS][0m test_validateSwapParams() (gas: 191486)
Logs:
  Started fork  mainnet  at block  19309223
  with id 0

 [32m[PASS][0m test_validateSwapParams_revertWhen_EmptyRoute() (gas: 72649)
Logs:
  Started fork  mainnet  at block  19309223
  with id 0

 [32m[PASS][0m test_validateSwapParams_revertWhen_FromTokenInvalidTokenIndex() (gas: 154341)
Logs:
  Started fork  mainnet  at block  19309223
  with id 0
```

```
 [32m[PASS] [0m test_validateSwapParams_revertWhen_InvalidRouteFromTokenMismatch() (gas: 72915)
Logs:
  Started fork  mainnet  at block  19309223
  with id 0

 [32m[PASS] [0m test_validateSwapParams_revertWhen_InvalidRouteToTokenMismatch() (gas: 73943)
Logs:
  Started fork  mainnet  at block  19309223
  with id 0

 [32m[PASS] [0m test_validateSwapParams_revertWhen_NextInvalidTokenIndex() (gas: 124521)
Logs:
  Started fork  mainnet  at block  19309223
  with id 0

Suite result:  [32mok [0m.  [32m20 [0m passed;  [31m0 [0m failed;  [33m0 [0m skipped; finished in 646.29ms
(34.11ms CPU time)

Ran 22 tests for test/forked/DYFIRedeemer.t.sol:DYFIRedeemer_ForkedTest
 [32m[PASS] [0m testFuzz_kill_revertWhen_CallerIsNotAdmin(address) (runs: 256, μ: 94801, ~: 94801)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS] [0m testFuzz_minYfiRedeem(uint256) (runs: 256, μ: 115497, ~: 115497)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS] [0m testFuzz_receiveFlashLoan_revertWhen_NotAuthorized(address) (runs: 256, μ: 33470, ~:
33470)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS] [0m testFuzz_receiverFlashLoan_revertWhen_InvalidTokensReceived(address) (runs: 256, μ: 35126,
~: 35208)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS] [0m
testFuzz_receiverFlashLoan_revertWhen_InvalidTokensReceived_InvalidAmountsArrayLength(uint256) (runs:
256, μ: 261045, ~: 262132)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS] [0m
testFuzz_receiverFlashLoan_revertWhen_InvalidTokensReceived_InvalidFeesArrayLength(uint256) (runs: 256,
μ: 270968, ~: 279013)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS] [0m
testFuzz_receiverFlashLoan_revertWhen_InvalidTokensReceived_InvalidTokensArrayLength(uint256) (runs: 256,
μ: 279445, ~: 272520)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS] [0m testFuzz_setSlippage_revertWhen_CallerIsNotAdmin(address) (runs: 256, μ: 95094, ~: 95094)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS] [0m testFuzz_setSlippage_revertWhen_SlippageTooHigh(uint256) (runs: 256, μ: 35792, ~: 35832)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0
```

```
 [32m[PASS] [0m test_expectedMassRedeemReward_ReturnsZeroOnZeroTotalDYfi() (gas: 5645)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS] [0m test_getLatestPrice_revertWhen_PriceFeedIncorrectRound() (gas: 355323)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS] [0m test_getLatestPrice_revertWhen_PriceFeedOutdated() (gas: 50068)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS] [0m test_getLatestPrice_revertWhen_PriceFeedReturnedZeroPrice() (gas: 244345)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS] [0m test_kill_revertWhen_AlreadyPaused() (gas: 84454)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS] [0m test_massRedeem() (gas: 1322143)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS] [0m test_massRedeem_LargeAmounts() (gas: 1336233)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS] [0m test_massRedeem_passWhen_CallerIsGnosisSafe() (gas: 1327650)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS] [0m test_massRedeem_revertWhen_CallerRewardEthTransferFailed() (gas: 1234949)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS] [0m test_massRedeem_revertWhen_InvalidArrayLength() (gas: 47566)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS] [0m test_massRedeem_revertWhen_NoDYfiToRedeem() (gas: 90196)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS] [0m test_massRedeem_revertWhen_Paused() (gas: 92203)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS] [0m test_setSlippage() (gas: 41257)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

Suite result:  [32mok [0m.  [32m22 [0m passed;  [31m0 [0m failed;  [33m0 [0m skipped; finished in 287.35s
(286.70s CPU time)

Ran 31 tests for test/unit/rewards/ERC20RewardsGauge.t.sol:ERC20RewardsGauge_Test
 [32m[PASS] [0m testFuzz_addReward(address,address) (runs: 256, μ: 126286, ~: 126397)
 [32m[PASS] [0m testFuzz_claimRewards(uint256,uint256) (runs: 256, μ: 1035486, ~: 1035381)
```

```
 [32m[PASS] [0m testFuzz_claimRewards_multipleRewards(uint256,uint256[8]) (runs: 256, μ: 9946022, ~:
9947331)
 [32m[PASS] [0m testFuzz_claimRewards_multipleUsers(uint256[10],uint256) (runs: 256, μ: 5001630, ~:
5001662)
 [32m[PASS] [0m testFuzz_claimRewards_noReceiverProvided(uint256,uint256) (runs: 256, μ: 1031963, ~:
1032780)
 [32m[PASS] [0m testFuzz_claimRewards_passWhen_IntegralIsZero(uint256,uint256) (runs: 256, μ: 952477, ~:
952955)
 [32m[PASS] [0m testFuzz_deposit(uint256) (runs: 256, μ: 483260, ~: 483058)
 [32m[PASS] [0m testFuzz_depositRewardToken(uint256) (runs: 256, μ: 511872, ~: 511521)
 [32m[PASS] [0m testFuzz_depositRewardToken_withPartialRewardRemaining(uint256,uint256) (runs: 256, μ:
600465, ~: 600512)
 [32m[PASS] [0m testFuzz_deposit_revertsWhen_depositsPaused(uint256) (runs: 256, μ: 362689, ~: 362554)
 [32m[PASS] [0m testFuzz_initialize_revetWhen_AlreadyInitialized(address) (runs: 256, μ: 35697, ~: 35784)
 [32m[PASS] [0m testFuzz_setRewardDistributor(address,address,address) (runs: 256, μ: 161979, ~: 162242)
 [32m[PASS] [0m testFuzz_setRewardDistributor_revertWhen_unauthorized(address) (runs: 256, μ: 164723, ~:
164723)
 [32m[PASS] [0m testFuzz_setRewardsReceiver(address) (runs: 256, μ: 56510, ~: 56588)
 [32m[PASS] [0m testFuzz_unpause(uint256) (runs: 256, μ: 456220, ~: 456110)
 [32m[PASS] [0m test_addReward_multipleRewards() (gas: 783779)
 [32m[PASS] [0m test_addReward_revertWhen_notManager() (gas: 94380)
 [32m[PASS] [0m test_addReward_revertsWhen_maxRewardsReached() (gas: 712967)
 [32m[PASS] [0m test_addReward_revertsWhen_rewardTokenAlreadyAdded() (gas: 142727)
 [32m[PASS] [0m test_addReward_revertsWhen_rewardTokenIsAsset() (gas: 42663)
 [32m[PASS] [0m test_addReward_revertsWhen_rewardTokenZeroAddress() (gas: 66521)
 [32m[PASS] [0m test_claimRewards_revertWhen_claimForAnotherUser() (gas: 937875)
 [32m[PASS] [0m test_claimRewards_revertWhen_rewardAmountTooLow() (gas: 834683)
 [32m[PASS] [0m test_decimals() (gas: 15989)
 [32m[PASS] [0m test_depositRewardToken_revertWhen_Unauthorized(address,address) (runs: 256, μ: 153969, ~:
154052)
 [32m[PASS] [0m test_initialize() (gas: 30763)
 [32m[PASS] [0m test_initialize_revertWhen_zeroAddress() (gas: 148294)
 [32m[PASS] [0m test_pause_revertWhen_notPauser() (gas: 37186)
 [32m[PASS] [0m test_setRewardDistributor_revertWhen_distributorNotSet() (gas: 40269)
 [32m[PASS] [0m test_setRewardDistributor_revertWhen_invalidDistributorAddress() (gas: 138275)
 [32m[PASS] [0m test_unpause_revertWhen_notAdmin() (gas: 93692)
Suite result:  [32mok [0m.  [32m31 [0m passed;  [31m0 [0m failed;  [33m0 [0m skipped; finished in 7.39s
(7.39s CPU time)

Ran 12 tests for test/unit/GaugeRewardReceiver.t.sol:GaugeRewardReceiver_Test
 [32m[PASS] [0m testFuzz_clone(address,address,address,address) (runs: 256, μ: 148062, ~: 148361)
 [32m[PASS] [0m testFuzz_harvest(uint256,uint80,uint80) (runs: 256, μ: 507229, ~: 504982)
 [32m[PASS] [0m testFuzz_rescue(uint256) (runs: 256, μ: 1195484, ~: 1195301)
 [32m[PASS] [0m testFuzz_rescue_revertWhen_CallerIsNotTheAdmin(address) (runs: 256, μ: 500343, ~: 500343)
 [32m[PASS] [0m test_clone() (gas: 154677)
 [32m[PASS] [0m test_harvest() (gas: 523546)
 [32m[PASS] [0m test_harvest_passWhen_NoRewards() (gas: 404410)
 [32m[PASS] [0m test_harvest_revertWhen_InvalidRewardSplit() (gas: 376471)
 [32m[PASS] [0m test_harvest_revertWhen_NotAuthorized() (gas: 375432)
 [32m[PASS] [0m test_initialize() (gas: 342074)
 [32m[PASS] [0m test_initialize_revertWhen_IsImplementation() (gas: 32313)
 [32m[PASS] [0m test_rescue_revertWhen_CannotRescueRewardToken() (gas: 441669)
Suite result:  [32mok [0m.  [32m12 [0m passed;  [31m0 [0m failed;  [33m0 [0m skipped; finished in 412.52ms
(404.49ms CPU time)

Ran 14 tests for test/integration/Integration.t.sol:YearnGaugeStrategy_IntegrationTest
 [32m[PASS] [0m testFuzz_deposit(uint256) (runs: 256, μ: 663371, ~: 668768)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS] [0m testFuzz_deposit_duringShutdown(uint256) (runs: 256, μ: 557468, ~: 560891)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS] [0m testFuzz_harvest_passWhen_RewardRateZero(uint256) (runs: 256, μ: 792774, ~: 825974)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS] [0m testFuzz_harvest_revertWhen_RewardRateTooLow(uint256) (runs: 256, μ: 945455, ~: 945470)
```

```
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS] [0m testFuzz_report_staking_rewards_profit(uint256) (runs: 256, μ: 2780288, ~: 2780166)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS] [0m testFuzz_report_staking_rewards_profit_erc20RewardsGauge_reward(uint256,uint256) (runs:
256, μ: 3126346, ~: 3126672)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS] [0m testFuzz_report_staking_rewards_profit_mulitple_users(uint256,uint256) (runs: 256, μ:
3311241, ~: 3312737)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS] [0m testFuzz_report_staking_rewards_profit_ysdRewardsGauge_reward(uint256,uint256) (runs: 256,
μ: 2004284, ~: 2004481)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS] [0m testFuzz_withdraw(uint256) (runs: 256, μ: 882549, ~: 895200)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS] [0m testFuzz_withdraw_duringShutdown(uint256) (runs: 256, μ: 914046, ~: 926394)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS] [0m testFuzz_withdraw_duringShutdownReport(uint256) (runs: 256, μ: 2666086, ~: 2666002)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS] [0m testFuzz_withdraw_withYSDGauge(uint256) (runs: 256, μ: 871439, ~: 878510)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS] [0m test_report_boosted_profit() (gas: 2040840)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS] [0m test_report_staking_rewards_profit_reward_split() (gas: 2641912)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

Suite result:  [32mok [0m.  [32m14 [0m passed;  [31m0 [0m failed;  [33m0 [0m skipped; finished in 15.08s
(14.29s CPU time)

Ran 31 tests for test/unit/MasterRegistry.t.sol:MasterRegistry_Test
 [32m[PASS] [0m testFuzz_addRegistry(bytes32,address) (runs: 256, μ: 105834, ~: 106010)
 [32m[PASS] [0m testFuzz_addRegistry_revertWhen_CalledByNonManager(bytes32,address) (runs: 256, μ: 95842,
~: 96057)
 [32m[PASS] [0m testFuzz_multicall_addRegistry(bytes32,bytes32,address,address) (runs: 256, μ: 190640, ~:
190962)
 [32m[PASS] [0m testFuzz_multicall_updateRegistry(bytes32,bytes32,address,address,address,address) (runs:
256, μ: 394383, ~: 394910)
 [32m[PASS] [0m testFuzz_resolveAddressToRegistryData(bytes32,bytes32,address,address,address) (runs: 256,
μ: 308194, ~: 308580)
 [32m[PASS] [0m testFuzz_resolveAddressToRegistryData_revertWhen_RegistryAddressNotFound(address) (runs:
256, μ: 14243, ~: 14243)
```

```
 [32m[PASS] [0m testFuzz_resolveNameAndVersionToAddress(bytes32,address,address) (runs: 256, μ: 207048, ~:
207331)
 [32m[PASS] [0m testFuzz_resolveNameAndVersionToAddress_revertWhen_NameAndVersionNotFound(bytes32,address)
(runs: 256, μ: 111338, ~: 111531)
 [32m[PASS] [0m testFuzz_resolveNameToAllAddresses(bytes32,address,address) (runs: 256, μ: 206903, ~:
207224)
 [32m[PASS] [0m testFuzz_resolveNameToAllAddresses_revertWhen_NamNotFound(bytes32) (runs: 256, μ: 13627,
~: 13627)
 [32m[PASS] [0m testFuzz_resolveNameToLatestAddress_revertWhen_NameNotFound(bytes32) (runs: 256, μ: 11748,
~: 11748)
 [32m[PASS] [0m testFuzz_updateRegistry(bytes32,address,address) (runs: 256, μ: 205512, ~: 205860)
 [32m[PASS] [0m testFuzz_updateRegistry_revertWhen_CalledByNonManager(bytes32,address,address) (runs: 256,
μ: 192746, ~: 193091)
 [32m[PASS] [0m
testFuzz_updateRegistry_revertWhen_CalledWithDuplicateAddress(bytes32,bytes32,address,address) (runs:
256, μ: 232605, ~: 233013)
 [32m[PASS] [0m testFuzz_updateRegistry_revertWhen_CalledWithEmptyAddress(bytes32) (runs: 256, μ: 33442,
~: 33590)
 [32m[PASS] [0m testFuzz_updateRegistry_revertWhen_CalledWithEmptyString(address) (runs: 256, μ: 33440, ~:
33521)
 [32m[PASS] [0m test_addRegistry_revertWhen_CalledWithDuplicateAddress(bytes32,bytes32,address) (runs:
256, μ: 135570, ~: 135906)
 [32m[PASS] [0m test_addRegistry_revertWhen_CalledWithDuplicateName(bytes32,address,address) (runs: 256,
μ: 133386, ~: 133751)
 [32m[PASS] [0m test_addRegistry_revertWhen_CalledWithEmptyAddress(bytes32) (runs: 256, μ: 33496, ~:
33622)
 [32m[PASS] [0m test_addRegistry_revertWhen_CalledWithEmptyString(address) (runs: 256, μ: 33437, ~: 33508)
 [32m[PASS] [0m test_getRoleAdmin_managerRole() (gas: 7896)
 [32m[PASS] [0m test_grantRole_adminRole() (gas: 202776)
 [32m[PASS] [0m test_grantRole_managerRole() (gas: 118574)
 [32m[PASS] [0m test_grantRole_revertWhen_CalledByNonAdmin() (gas: 97321)
 [32m[PASS] [0m test_init() (gas: 16943)
 [32m[PASS] [0m test_renouceRole_adminRole() (gas: 42383)
 [32m[PASS] [0m test_renouceRole_managerRole() (gas: 49005)
 [32m[PASS] [0m test_revokeRoleF_adminRole() (gas: 149661)
 [32m[PASS] [0m test_revokeRole_adminRole() (gas: 150467)
 [32m[PASS] [0m test_revokeRole_managerRole_revertWhen_RevokeRoleWithoutAdmin() (gas: 99727)
 [32m[PASS] [0m test_updateRegistry_revertWhen_NameNotFound(bytes32,address) (runs: 256, μ: 36474, ~:
36711)
Suite result:  [32mok [0m.  [32m31 [0m passed;  [31m0 [0m failed;  [33m0 [0m skipped; finished in 951.92ms
(950.44ms CPU time)

Ran 44 tests for test/unit/rewards/MiniChefV3.t.sol:MiniChefV3_Test
 [32m[PASS] [0m testFuzz_commitReward(uint256) (runs: 256, μ: 189871, ~: 193913)
 [32m[PASS] [0m testFuzz_pidOfLPToken_revertWhen_InvalidLPToken(address) (runs: 256, μ: 223486, ~: 223486)
 [32m[PASS] [0m testFuzz_rescue_passWhen_RescueLPToken(uint256,uint256) (runs: 256, μ: 563427, ~: 563313)
 [32m[PASS] [0m testFuzz_rescue_revertWhen_CallerIsNotAdmin(address,address,uint256) (runs: 256, μ: 95907,
~: 95945)
 [32m[PASS] [0m testFuzz_rescue_revertWhen_InsufficientBalance_RewardToken(uint256,uint256) (runs: 256, μ:
530937, ~: 530812)
 [32m[PASS] [0m
testFuzz_rescue_revertWhen_InsufficientBalance_RewardTokenIsLPToken(uint256,uint256,uint256) (runs: 256,
μ: 765588, ~: 765601)
 [32m[PASS] [0m testFuzz_setRewardPerSecond(uint256) (runs: 256, μ: 57087, ~: 57837)
 [32m[PASS] [0m testFuzz_setRewardPerSecond_revertWhen_CallerIsNotTimelock(uint256) (runs: 256, μ: 98559,
~: 98271)
 [32m[PASS] [0m testFuzz_setRewardPerSecond_revertWhen_RewardRateTooHigh(uint256) (runs: 256, μ: 37391, ~:
37506)
 [32m[PASS] [0m test_add() (gas: 947311)
 [32m[PASS] [0m test_add_revertWhen_CallerIsNotTimelock() (gas: 96914)
 [32m[PASS] [0m test_add_revertWhen_LPTokenAlreadyAdded() (gas: 247429)
 [32m[PASS] [0m test_add_revertWhen_LPTokenIsZero() (gas: 32922)
 [32m[PASS] [0m test_constructor() (gas: 21078)
 [32m[PASS] [0m test_constructor_revertWhen_AdminIsZero() (gas: 310387)
 [32m[PASS] [0m test_constructor_revertWhen_RewardTokenIsZero() (gas: 308273)
 [32m[PASS] [0m test_deposit() (gas: 463609)
 [32m[PASS] [0m test_deposit_passWhen_RewarderIsNotZero() (gas: 478643)
 [32m[PASS] [0m test_deposit_revertWhen_Paused() (gas: 369255)
 [32m[PASS] [0m test_deposit_revertWhen_ZeroAmount() (gas: 247727)
 [32m[PASS] [0m test_emergencyWithdraw() (gas: 517683)
 [32m[PASS] [0m test_emergencyWithdraw_passWhen_RewarderIsFaulty() (gas: 540064)
 [32m[PASS] [0m test_emergencyWithdraw_passWhen_RewarderIsNotZero() (gas: 537098)
```

```
 [32m[PASS] [0m test_grantRole_TimelockRole_revertWhen_CallerIsNotTimelock() (gas: 198708)
 [32m[PASS] [0m test_harvest() (gas: 817985)
 [32m[PASS] [0m test_harvest_passWhen_RewarderIsNotZero() (gas: 832984)
 [32m[PASS] [0m test_harvest_passWhen_UnpaidRewards() (gas: 1047313)
 [32m[PASS] [0m test_isLPTokenAdded() (gas: 221676)
 [32m[PASS] [0m test_pause_revertWhen_notPauser() (gas: 36987)
 [32m[PASS] [0m test_pidOfLPToken() (gas: 219038)
 [32m[PASS] [0m test_poolLength() (gas: 12327158)
 [32m[PASS] [0m test_rescue() (gas: 586680)
 [32m[PASS] [0m test_rescue_passWhen_RandomToken() (gas: 856606)
 [32m[PASS] [0m test_rescue_revertWhen_InsufficientBalance() (gas: 525500)
 [32m[PASS] [0m test_set() (gas: 312122)
 [32m[PASS] [0m test_set_revertWhen_CallerIsNotTimelock() (gas: 308254)
 [32m[PASS] [0m test_set_revertWhen_LPTokenDoesNotMatchPoolId() (gas: 1823120)
 [32m[PASS] [0m test_set_revertWhen_LPTokenNotAdded() (gas: 38349)
 [32m[PASS] [0m test_unPause() (gas: 80286)
 [32m[PASS] [0m test_unpause_revertWhen_notAdmin() (gas: 94401)
 [32m[PASS] [0m test_updatePool() (gas: 246803)
 [32m[PASS] [0m test_withdraw() (gas: 518084)
 [32m[PASS] [0m test_withdraw_passWhen_RewarderIsNotZero() (gas: 540231)
 [32m[PASS] [0m test_withdraw_revertWhen_ZeroAmount() (gas: 476664)
Suite result: [32mok [0m. [32m44 [0m passed; [31m0 [0m failed; [33m0 [0m skipped; finished in 704.41ms
(695.71ms CPU time)

Ran 9 tests for test/unit/Rescuable.t.sol:Rescuable_Test
 [32m[PASS] [0m testFuzz_rescue_erc20(uint256) (runs: 256, μ: 300244, ~: 300192)
 [32m[PASS] [0m testFuzz_rescue_eth(uint256) (runs: 256, μ: 66408, ~: 66313)
 [32m[PASS] [0m test_rescue_erc20_balanceExceedsTotalBalance() (gas: 299878)
 [32m[PASS] [0m test_rescue_erc20_revertsOnZeroBalance() (gas: 40854)
 [32m[PASS] [0m test_rescue_erc20_zeroBalance() (gas: 299774)
 [32m[PASS] [0m test_rescue_eth_balanceExceedsTotalBalance() (gas: 40215)
 [32m[PASS] [0m test_rescue_eth_revertsOnFailedTransfer() (gas: 40691)
 [32m[PASS] [0m test_rescue_eth_revertsOnZeroBalance() (gas: 33117)
 [32m[PASS] [0m test_rescue_eth_zeroBalance() (gas: 40064)
Suite result: [32mok [0m. [32m9 [0m passed; [31m0 [0m failed; [33m0 [0m skipped; finished in 219.21ms
(214.72ms CPU time)

Ran 9 tests for test/unit/RewardForwarder.t.sol:RewardForwarder_Test
 [32m[PASS] [0m testFuzz_setTreasury(address) (runs: 256, μ: 47134, ~: 47204)
 [32m[PASS] [0m testFuzz_setTreasuryBps(uint256) (runs: 256, μ: 68236, ~: 71187)
 [32m[PASS] [0m testFuzz_setTreasuryBps_revertWhen_invalidTreasuryBps(uint256) (runs: 256, μ: 53160, ~:
53162)
 [32m[PASS] [0m test_approveRewardToken() (gas: 65948)
 [32m[PASS] [0m test_forwardRewardToken() (gas: 567447)
 [32m[PASS] [0m test_initialize() (gas: 24290)
 [32m[PASS] [0m test_initialize_revertWhen_zeroDestination() (gas: 153378)
 [32m[PASS] [0m test_setTreasuryBps_revertWhen_notAdmin() (gas: 99763)
 [32m[PASS] [0m test_setTreasury_revertWhen_notAdmin() (gas: 97248)
Suite result: [32mok [0m. [32m9 [0m passed; [31m0 [0m failed; [33m0 [0m skipped; finished in 101.44ms
(98.56ms CPU time)

Ran 4 tests for test/forked/Router.t.sol:Router_ForkedTest
 [32m[PASS] [0m test_curveLpTokenToYearnGauge() (gas: 539069)
Logs:
  Started fork  mainnet  at block  19072737
  with id 0

 [32m[PASS] [0m test_curveLpTokenToYearnGauge_revertWhen_insufficientShares() (gas: 476281)
Logs:
  Started fork  mainnet  at block  19072737
  with id 0

 [32m[PASS] [0m test_pullTokensWithPermit2() (gas: 374368)
Logs:
  Started fork  mainnet  at block  19072737
  with id 0

 [32m[PASS] [0m test_pullTokensWithPermit2_revertWhen_InvalidTo() (gas: 319452)
Logs:
  Started fork  mainnet  at block  19072737
  with id 0
```

```
Suite result: [32mok[0m. [32m4[0m passed; [31m0[0m failed; [33m0[0m skipped; finished in 647.78ms
(30.55ms CPU time)

Ran 33 tests for test/unit/StakingDelegateRewards.t.sol:StakingDelegateRewards_Test
[32m[PASS][0m testFuzz_constructor(address,address,address,address) (runs: 256, μ: 1945409, ~: 1945647)
[32m[PASS][0m testFuzz_notifyRewardAmount(uint256) (runs: 256, μ: 467452, ~: 467472)
[32m[PASS][0m testFuzz_notifyRewardAmount_revertWhen_RewardRateTooLow(uint256) (runs: 256, μ: 387636, ~:
389013)
[32m[PASS][0m testFuzz_setRewardReceiver(address) (runs: 256, μ: 57177, ~: 57251)
[32m[PASS][0m test_addStakingToken() (gas: 93783)
[32m[PASS][0m test_addStakingToken_revertWhen_CallerIsNotStakingDelegate() (gas: 35134)
[32m[PASS][0m test_addStakingToken_revertWhen_StakingTokenAlreadyAdded() (gas: 110863)
[32m[PASS][0m test_constructor() (gas: 20644)
[32m[PASS][0m test_constructor_revertWhen_ZeroAddress() (gas: 481694)
[32m[PASS][0m test_earned() (gas: 571507)
[32m[PASS][0m test_earned_passWhen_MultipleUsers() (gas: 674321)
[32m[PASS][0m test_getReward() (gas: 808716)
[32m[PASS][0m test_getRewardForDuration() (gas: 466668)
[32m[PASS][0m test_getReward_passWhen_CalledByUsers() (gas: 1027367)
[32m[PASS][0m test_getReward_passWhen_MultipleUsers() (gas: 1027969)
[32m[PASS][0m test_getReward_passWhen_RewardReceiverIsSet() (gas: 862920)
[32m[PASS][0m test_grantRole_TimelockRole_revertWhen_CallerIsNotTimelock() (gas: 200966)
[32m[PASS][0m test_lastTimeRewardApplicable() (gas: 469191)
[32m[PASS][0m test_notifyRewardAmount() (gas: 466574)
[32m[PASS][0m test_notifyRewardAmount_passWhen_RemainingRewardExists() (gas: 691222)
[32m[PASS][0m test_notifyRewardAmount_revertWhen_CallerIsNotRewardDistributor() (gas: 115155)
[32m[PASS][0m test_recoverERC20() (gas: 985249)
[32m[PASS][0m test_recoverERC20_revertWhen_CallerIsNotAdmin() (gas: 97708)
[32m[PASS][0m test_recoverERC20_revertWhen_RescueRewardToken_RescueNotAllowed() (gas: 38386)
[32m[PASS][0m test_recoverERC20_revertWhen_RescueStakingToken_RescueNotAllowed() (gas: 115610)
[32m[PASS][0m test_rewardPerToken() (gas: 550328)
[32m[PASS][0m test_setRewardDuration_revertWhen_RewardsDurationCannotBeZero() (gas: 112661)
[32m[PASS][0m test_setRewardDuration_revertWhen_StakingTokenNotAdded() (gas: 39796)
[32m[PASS][0m test_setRewardsDuration() (gas: 122012)
[32m[PASS][0m test_setRewardsDuration_revertWhen_CallerIsNotTimelock() (gas: 96379)
[32m[PASS][0m test_setRewardsDuration_revertWhen_PreviousRewardsPeriodNotCompleted() (gas: 488744)
[32m[PASS][0m test_updateUserBalance() (gas: 173181)
[32m[PASS][0m test_updateUserBalance_revertWhen_CallerIsNotStakingDelegate() (gas: 35487)
Suite result: [32mok[0m. [32m33[0m passed; [31m0[0m failed; [33m0[0m skipped; finished in 550.91ms
(542.69ms CPU time)

Ran 1 test for test/forked/SwapAndLock.t.sol:SwapAndLock_ForkedTest
[32m[PASS][0m test_lockYfi() (gas: 977032)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

Suite result: [32mok[0m. [32m1[0m passed; [31m0[0m failed; [33m0[0m skipped; finished in 636.96ms
(9.12ms CPU time)

Ran 5 tests for test/unit/SwapAndLock.t.sol:SwapAndLock_Test
[32m[PASS][0m testFuzz_setDYfiRedeemer(address) (runs: 256, μ: 93401, ~: 93466)
[32m[PASS][0m testFuzz_setDYfiRedeemer_passWhen_Replacing(address,address) (runs: 256, μ: 162900, ~:
163043)
[32m[PASS][0m testFuzz_setDYfiRedeemer_revertWhen_SameAddress(address) (runs: 256, μ: 114918, ~: 115048)
[32m[PASS][0m test_lockYfi() (gas: 303896)
[32m[PASS][0m test_setDYfiRedeemer_revertWhen_ZeroAddress() (gas: 34984)
Suite result: [32mok[0m. [32m5[0m passed; [31m0[0m failed; [33m0[0m skipped; finished in 153.81ms
(149.30ms CPU time)

Ran 9 tests for test/unit/rewards/YSDRewardsGauge.t.sol:YSDRewardsGauge_Test
[32m[PASS][0m testFuzz_deposit(uint256) (runs: 256, μ: 427346, ~: 427236)
[32m[PASS][0m testFuzz_deposit_revertWhen_MaxTotalAssetsExceeded(uint256) (runs: 256, μ: 324958, ~:
324815)
[32m[PASS][0m testFuzz_initialize_revetWhen_AlreadyInitialized(address,address,address) (runs: 256, μ:
36886, ~: 37110)
[32m[PASS][0m testFuzz_withdraw(uint256) (runs: 256, μ: 488392, ~: 488103)
[32m[PASS][0m testFuzz_withdraw_revertsWhen_noSpenderAllowance(uint256) (runs: 256, μ: 779672, ~:
779267)
[32m[PASS][0m test_initialize() (gas: 24266)
[32m[PASS][0m test_initialize_revertWhen_ZeroAddress() (gas: 246462)
[32m[PASS][0m test_setStakingDelegateRewardsReceiver() (gas: 571634)
```

```
 [32m[PASS] [0m test_setStakingDelegateRewardsReceiver_revertWhen_notAdmin() (gas: 97369)
Suite result: [32mok [0m. [32m9 [0m passed; [31m0 [0m failed; [33m0 [0m skipped; finished in 1.02s (1.02s
CPU time)

Ran 9 tests for test/forked/YearnGaugeStrategy.t.sol:YearnGaugeStrategy_ForkedTest
 [32m[PASS] [0m testFuzz_deposit(uint256) (runs: 256, μ: 606862, ~: 612768)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS] [0m testFuzz_deposit_duringShutdown(uint256) (runs: 256, μ: 557291, ~: 560858)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS] [0m testFuzz_report_passWhen_noProfits(uint256) (runs: 256, μ: 738044, ~: 738031)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS] [0m testFuzz_report_staking_rewards_profit(uint256) (runs: 256, μ: 2161382, ~: 2161384)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS] [0m testFuzz_withdraw(uint256) (runs: 256, μ: 801804, ~: 815024)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS] [0m testFuzz_withdraw_duringShutdown(uint256) (runs: 256, μ: 834478, ~: 846253)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS] [0m testFuzz_withdraw_duringShutdownReport(uint256) (runs: 256, μ: 2092477, ~: 2092468)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS] [0m test_setHarvestSwapParams_nonManager() (gas: 50692)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS] [0m test_setHarvestSwapParams_validateSwapParams_revertWhen_InvalidCoinIndex() (gas: 78862)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

Suite result: [32mok [0m. [32m9 [0m passed; [31m0 [0m failed; [33m0 [0m skipped; finished in 7.03s (6.27s
CPU time)

Ran 20 tests for test/unit/strategies/YearnGaugeStrategy.t.sol:YearnGaugeStrategy_Test
 [32m[PASS] [0m testFuzz_deposit(uint256) (runs: 256, μ: 698611, ~: 698520)
 [32m[PASS] [0m testFuzz_deposit_passWhen_DuringShutdown(uint256) (runs: 256, μ: 366636, ~: 366604)
 [32m[PASS] [0m testFuzz_deposit_revertWhen_MaxTotalAssetsExceeded(uint256) (runs: 256, μ: 369782, ~:
370454)
 [32m[PASS] [0m testFuzz_emergencyWithdraw(uint256) (runs: 256, μ: 944188, ~: 944047)
 [32m[PASS] [0m testFuzz_report_passWhen_DuringShutdown(uint256) (runs: 256, μ: 1985439, ~: 1985352)
 [32m[PASS] [0m testFuzz_report_passWhen_noProfits(uint256) (runs: 256, μ: 839790, ~: 839699)
 [32m[PASS] [0m testFuzz_report_passWhen_stakingRewardsProfit(uint256) (runs: 256, μ: 1933394, ~: 1933311)
 [32m[PASS] [0m testFuzz_report_passWhen_stakingRewardsProfitMultipleUsers(uint256,uint256) (runs: 256, μ:
3016665, ~: 3016522)
 [32m[PASS] [0m testFuzz_setDYfiRedeemer(address) (runs: 256, μ: 97004, ~: 97087)
 [32m[PASS] [0m testFuzz_setDYfiRedeemer_passWhen_Replacing(address,address) (runs: 256, μ: 170178, ~:
170342)
 [32m[PASS] [0m testFuzz_setDYfiRedeemer_revertWhen_SameAddress(address) (runs: 256, μ: 122228, ~: 122372)
 [32m[PASS] [0m testFuzz_setHarvestSwapParams_revertWhen_CallerIsNotManagement(address) (runs: 256, μ:
56064, ~: 56064)
 [32m[PASS] [0m testFuzz_setMaxTotalAssets(uint256) (runs: 256, μ: 43392, ~: 43819)
 [32m[PASS] [0m testFuzz_setMaxTotalAssets_revert_when_non_manager(address,uint256) (runs: 256, μ: 38827,
```

```
~: 38722)
 [32m[PASS] [0m testFuzz_withdraw(uint256) (runs: 256, μ: 798629, ~: 798512)
 [32m[PASS] [0m testFuzz_withdraw_passWhen_DuringShutdown(uint256) (runs: 256, μ: 829521, ~: 829397)
 [32m[PASS] [0m test_constructor() (gas: 18393)
 [32m[PASS] [0m test_emergencyWithdraw_revertWhen_nonManager(address) (runs: 256, μ: 63572, ~: 63572)
 [32m[PASS] [0m test_setDYfiRedeemer_revertWhen_ZeroAddress() (gas: 38646)
 [32m[PASS] [0m test_setdYfiRedeemer_revertWhen_ZeroAddress() (gas: 38618)
Suite result:  [32mok [0m.  [32m20 [0m passed;  [31m0 [0m failed;  [33m0 [0m skipped; finished in 10.28s
(6.14s CPU time)

Ran 32 tests for test/forked/YearnStakingDelegate.t.sol:YearnStakingDelegate_ForkedTest
 [32m[PASS] [0m testFuzz_constructor(address) (runs: 256, μ: 37528, ~: 37528)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS] [0m testFuzz_deposit(uint256) (runs: 256, μ: 1184917, ~: 1194575)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS] [0m testFuzz_deposit_revertWhen_GaugeRewardsNotYetAdded(uint256) (runs: 256, μ: 1004292, ~:
1004105)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS] [0m testFuzz_earlyUnlock(uint256) (runs: 256, μ: 815968, ~: 815971)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS] [0m testFuzz_lockYFI(uint256) (runs: 256, μ: 577323, ~: 577325)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS] [0m testFuzz_lockYFI_revertWhen_CreatingLockWithLessThanMinAmount(uint256) (runs: 256, μ:
318386, ~: 318384)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS] [0m testFuzz_setGaugeRewardSplit(uint80,uint80) (runs: 256, μ: 65507, ~: 65492)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS] [0m testFuzz_setGaugeRewardSplit_revertWhen_InvalidRewardSplit(uint80,uint80,uint80) (runs:
256, μ: 39954, ~: 39944)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS] [0m testFuzz_setTreasury(address) (runs: 256, μ: 42270, ~: 42339)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS] [0m testFuzz_withdraw(uint256) (runs: 256, μ: 1330540, ~: 1346929)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS] [0m testFuzz_withdraw_toReceiver(uint256) (runs: 256, μ: 1349836, ~: 1369345)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS] [0m testFuzz_withdraw_toReceiver_revertsWhen_zeroAmount(uint256) (runs: 256, μ: 1206318, ~:
1216758)
Logs:
```

```
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS][0m test_claimBoostRewards() (gas: 2028598)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS][0m test_claimExitRewards() (gas: 1940445)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS][0m test_earlyUnlock() (gas: 815126)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS][0m test_earlyUnlock_revertWhen_CallerIsNotTimelock() (gas: 656051)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS][0m test_earlyUnlock_revertWhen_PerpeutalLockEnabled() (gas: 599444)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS][0m test_harvest_passWhen_LargeVeYFI() (gas: 2026142)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS][0m test_harvest_passWhen_NoVeYFI() (gas: 1505384)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS][0m test_harvest_passWhen_SomeVeYFI() (gas: 2527537)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS][0m test_harvest_passWhen_WithVeYfiSplit() (gas: 2117751)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS][0m test_harvest_revertWhen_GaugeRewardsNotYetAdded() (gas: 91930)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS][0m test_harvest_revertWhen_SwapAndLockNotSet() (gas: 513084)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS][0m test_lockYFI() (gas: 573660)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS][0m test_lockYFI_revertWhen_PerpetualLockDisabled() (gas: 303286)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS][0m test_lockYFI_revertWhen_WithZeroAmount() (gas: 35928)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0
```

```
 [32m[PASS] [0m test_rescueDYfi() (gas: 253920)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS] [0m test_rescueYfi() (gas: 304291)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS] [0m test_setSnapshotDelegate() (gas: 89093)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS] [0m test_setSnapshotDelegate_revertWhen_ZeroAddress() (gas: 35876)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS] [0m test_setTreasury_revertWhen_CallerIsNotTimelock() (gas: 94317)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

 [32m[PASS] [0m test_setTreasury_revertWhen_ZeroAddress() (gas: 35523)
Logs:
  Started fork  mainnet  at block  18748116
  with id 0

Suite result:  [32mok [0m.  [32m32 [0m passed;  [31m0 [0m failed;  [33m0 [0m skipped; finished in 4.65s
(4.00s CPU time)

Ran 61 tests for test/unit/YearnStakingDelegate.t.sol:YearnStakingDelegate_Test
 [32m[PASS] [0m testFuzz_constructor(address) (runs: 256, μ: 37536, ~: 37536)
 [32m[PASS] [0m testFuzz_deposit(uint256) (runs: 256, μ: 721726, ~: 721034)
 [32m[PASS] [0m testFuzz_deposit_passWhen_unpaused(uint256) (runs: 256, μ: 791979, ~: 791233)
 [32m[PASS] [0m testFuzz_deposit_revertWhen_GaugeRewardsNotYetAdded(uint256) (runs: 256, μ: 1004161, ~:
1003959)
 [32m[PASS] [0m testFuzz_earlyUnlock(uint256) (runs: 256, μ: 409308, ~: 409263)
 [32m[PASS] [0m testFuzz_execute(bytes4,bytes32,address,uint256) (runs: 256, μ: 185141, ~: 193348)
 [32m[PASS] [0m testFuzz_setGaugeRewardSplit(uint80,uint80) (runs: 256, μ: 68211, ~: 68197)
 [32m[PASS] [0m testFuzz_setGaugeRewardSplit_revertWhen_InvalidRewardSplit(uint80,uint80,uint80) (runs:
256, μ: 40044, ~: 40031)
 [32m[PASS] [0m testFuzz_setSwapAndLock(address) (runs: 256, μ: 59278, ~: 59354)
 [32m[PASS] [0m testFuzz_setTreasury(address) (runs: 256, μ: 43759, ~: 43838)
 [32m[PASS] [0m testFuzz_withdraw(uint256) (runs: 256, μ: 775078, ~: 774205)
 [32m[PASS] [0m testFuzz_withdraw_passWhen_Paused(uint256) (runs: 256, μ: 825135, ~: 824325)
 [32m[PASS] [0m testGauge() (gas: 2471)
 [32m[PASS] [0m testVault() (gas: 2450)
 [32m[PASS] [0m test_addGaugeRewards() (gas: 248720)
 [32m[PASS] [0m test_addGaugeRewards_revertWhen_CallerIsNotAdmin() (gas: 103929)
 [32m[PASS] [0m test_addGaugeRewards_revertWhen_GaugeRewardsAlreadyAdded() (gas: 281308)
 [32m[PASS] [0m test_addGaugeRewards_revertWhen_GaugeZeroAddress() (gas: 42744)
 [32m[PASS] [0m test_addGaugeRewards_revertWhen_StakingDelegateRewardsZeroAddress() (gas: 42776)
 [32m[PASS] [0m test_claimBoostRewards() (gas: 315460)
 [32m[PASS] [0m test_claimExitRewards() (gas: 315444)
 [32m[PASS] [0m test_deposit_revertWhen_Paused() (gas: 85607)
 [32m[PASS] [0m test_deposit_revertWhen_ZeroAmount() (gas: 37318)
 [32m[PASS] [0m test_earlyUnlock() (gas: 408952)
 [32m[PASS] [0m test_earlyUnlock_revertWhen_CallerIsNotTimelock() (gas: 385642)
 [32m[PASS] [0m test_earlyUnlock_revertWhen_PerpeutalLockEnabled() (gas: 340252)
 [32m[PASS] [0m test_execute() (gas: 193927)
 [32m[PASS] [0m test_execute_passWhen_ZeroValue() (gas: 160555)
 [32m[PASS] [0m test_execute_revertWhen_CallerIsNotTimelock() (gas: 109342)
 [32m[PASS] [0m test_execute_revertWhen_ExecutionFailed() (gas: 59418)
 [32m[PASS] [0m test_execute_revertWhen_TargetIsDYFIRewardPool_ExecutionNotAllowed() (gas: 52805)
 [32m[PASS] [0m test_execute_revertWhen_TargetIsDYFI_ExecutionNotAllowed() (gas: 52805)
 [32m[PASS] [0m test_execute_revertWhen_TargetIsGaugeToken_ExecutionNotAllowed() (gas: 290720)
 [32m[PASS] [0m test_execute_revertWhen_TargetIsRewardReceiver_ExecutionNotAllowed() (gas: 291922)
 [32m[PASS] [0m test_execute_revertWhen_TargetIsStakingRewards_ExecutionNotAllowed() (gas: 291910)
```

```
[32m[PASS][0m test_execute_revertWhen_TargetIsVeYFI_ExecutionNotAllowed() (gas: 52805)
[32m[PASS][0m test_execute_revertWhen_TargetIsYFIRewardPool_ExecutionNotAllowed() (gas: 52738)
[32m[PASS][0m test_execute_revertWhen_TargetIsYFI_ExecutionNotAllowed() (gas: 52738)
[32m[PASS][0m test_grantRole_TimelockRole_revertWhen_CallerIsNotTimelock() (gas: 200861)
[32m[PASS][0m test_harvest_revertWhen_GaugeRewardsNotYetAdded() (gas: 91016)
[32m[PASS][0m test_harvest_revertWhen_SwapAndLockNotSet() (gas: 272551)
[32m[PASS][0m test_lockYFI() (gas: 318099)
[32m[PASS][0m test_lockYFI_revertWhen_PerpetualLockDisabled() (gas: 304195)
[32m[PASS][0m test_lockYFI_revertWhen_WithZeroAmount() (gas: 35772)
[32m[PASS][0m test_lockYfi_revertWhen_Paused() (gas: 86312)
[32m[PASS][0m test_pause_revertWhen_notPauser() (gas: 36987)
[32m[PASS][0m test_setGaugeRewardSplit_revertWhen_CallerIsNotTimelock() (gas: 97165)
[32m[PASS][0m test_setPerpetualLock() (gas: 76624)
[32m[PASS][0m test_setSwapAndLock_revertWhen_CallerIsNotTimelock() (gas: 94222)
[32m[PASS][0m test_setSwapAndLock_revertWhen_ZeroAddress() (gas: 35479)
[32m[PASS][0m test_setTreasury_revertWhen_CallerIsNotTimelock() (gas: 94200)
[32m[PASS][0m test_setTreasury_revertWhen_ZeroAddress() (gas: 35532)
[32m[PASS][0m test_unpause() (gas: 82742)
[32m[PASS][0m test_unpause_revertWhen_notAdmin() (gas: 91744)
[32m[PASS][0m test_updateGaugeRewards() (gas: 855427)
[32m[PASS][0m test_updateGaugeRewards_revertWhen_CallerIsNotTimelock() (gas: 103940)
[32m[PASS][0m test_updateGaugeRewards_revertWhen_GaugeRewardsAlreadyAdded() (gas: 283344)
[32m[PASS][0m test_updateGaugeRewards_revertWhen_GaugeRewardsNotYetAdded() (gas: 47417)
[32m[PASS][0m test_updateGaugeRewards_revertWhen_GaugeZeroAddress() (gas: 42775)
[32m[PASS][0m test_updateGaugeRewards_revertWhen_StakingDelegateRewardsZeroAddress() (gas: 42764)
[32m[PASS][0m test_withdraw_revertWhen_ZeroAmount() (gas: 35278)
Suite result: [32mok[0m. [32m61[0m passed; [31m0[0m failed; [33m0[0m skipped; finished in 1.88s
(1.87s CPU time)
```

Code Coverage

The coverage shown below only includes the contracts in scope. However, the coverage for most of the codebase is similarly good.

File	% Lines	% Statements	% Branches	% Funcs
src/Yearn4626RouterExt.sol	100.00% (**3/**3)	100.00% (**6/**6)	100.00% (**4/**4)	100.00% (**2/**2)
src/governance/CoveToken.sol	100.00% (**29/**29)	100.00% (**31/**31)	100.00% (**12/**12)	100.00% (**12/**12)
src/registries/CoveYearnGaugeFactory.sol	100.00% (**71/**71)	98.88% (**88/**89)	100.00% (**16/**16)	100.00% (**14/**14)
src/rewards/BaseRewardsGauge.sol	100.00% (**102/**102)	100.00% (**133/**133)	93.48% (**43/**46)	100.00% (**14/**14)
src/rewards/MiniChefV3.sol	100.00% (**113/**113)	100.00% (**137/**137)	82.50% (**33/**40)	100.00% (**16/**16)
src/rewards/RewardForwarder.sol	100.00% (**19/**19)	100.00% (**24/**24)	87.50% (**7/**8)	100.00% (**7/**7)
src/rewards/YSDRewardsGauge.sol	100.00% (**22/**22)	100.00% (**33/**33)	100.00% (**8/**8)	80.00% (**4/**5)

Code coverage after the fix review:

File	% Lines	% Statements	% Branches	% Funcs
src/Yearn4626RouterExt.sol	100.00% (**3/**3)	100.00% (**6/**6)	100.00% (**4/**4)	100.00% (**2/**2)
src/governance/CoveToken.sol	100.00% (**37/**37)	100.00% (**37/**37)	100.00% (**16/**16)	100.00% (**12/**12)

File	% Lines	% Statements	% Branches	% Funcs
src/registries/CoveYearnGaugeFactory.sol	100.00% (**92/**92)	99.12% (**112/**113)	100.00% (**26/**26)	100.00% (**18/**18)
src/rewards/BaseRewardsGauge.sol	100.00% (**117/**117)	100.00% (**147/**147)	93.75% (**45/**48)	100.00% (**18/**18)
src/rewards/MiniChefV3.sol	100.00% (**130/**130)	100.00% (**156/**156)	88.89% (**48/**54)	100.00% (**18/**18)
src/rewards/RewardForwarder.sol	100.00% (**20/**20)	100.00% (**25/**25)	87.50% (**7/**8)	100.00% (**7/**7)
src/rewards/YSDRewardsGauge.sol	100.00% (**22/**22)	100.00% (**33/**33)	100.00% (**8/**8)	80.00% (**4/**5)

Changelog

- 2024-02-29 - Initial report
- 2024-03-08 - Final report

About Quantstamp

Quantstamp is a global leader in blockchain security. Founded in 2017, Quantstamp's mission is to securely onboard the next billion users to Web3 through its best-in-class Web3 security products and services.

Quantstamp's team consists of cybersecurity experts hailing from globally recognized organizations including Microsoft, AWS, BMW, Meta, and the Ethereum Foundation. Quantstamp engineers hold PhDs or advanced computer science degrees, with decades of combined experience in formal verification, static analysis, blockchain audits, penetration testing, and original leading-edge research.

To date, Quantstamp has performed more than 500 audits and secured over $200 billion in digital asset risk from hackers. Quantstamp has worked with a diverse range of customers, including startups, category leaders and financial institutions. Brands that Quantstamp has worked with include Ethereum 2.0, Binance, Visa, PayPal, Polygon, Avalanche, Curve, Solana, Compound, Lido, MakerDAO, Arbitrum, OpenSea and the World Economic Forum.

Quantstamp's collaborations and partnerships showcase our commitment to world-class research, development and security. We're honored to work with some of the top names in the industry and proud to secure the future of web3.

Notable Collaborations & Customers:

- Blockchains: Ethereum 2.0, Near, Flow, Avalanche, Solana, Cardano, Binance Smart Chain, Hedera Hashgraph, Tezos
- DeFi: Curve, Compound, Maker, Lido, Polygon, Arbitrum, SushiSwap
- NFT: OpenSea, Parallel, Dapper Labs, Decentraland, Sandbox, Axie Infinity, Illuvium, NBA Top Shot, Zora
- Academic institutions: National University of Singapore, MIT

Timeliness of content

The content contained in the report is current as of the date appearing on the report and is subject to change without notice, unless indicated otherwise by Quantstamp; however, Quantstamp does not guarantee or warrant the accuracy, timeliness, or completeness of any report you access using the internet or other means, and assumes no obligation to update any information following publication or other making available of the report to you by Quantstamp.

Notice of confidentiality

This report, including the content, data, and underlying methodologies, are subject to the confidentiality and feedback provisions in your agreement with Quantstamp. These materials are not to be disclosed, extracted, copied, or distributed except to the extent expressly authorized by Quantstamp.

Links to other websites

You may, through hypertext or other computer links, gain access to web sites operated by persons other than Quantstamp. Such hyperlinks are provided for your reference and convenience only, and are the exclusive responsibility of such web sites&aspo; owners. You agree that Quantstamp are not responsible for the content or operation of such web sites, and that Quantstamp shall have no liability to you or any other person or entity for the use of third-party web sites. Except as described below, a hyperlink from this web site to another web site does not imply or mean that Quantstamp endorses the content on that web site or the operator or operations of that site. You are solely responsible for determining the extent to which you may use any content at any other web sites to which you link from the report. Quantstamp assumes no responsibility for the use of third-party software on any website and shall have no liability whatsoever to any person or entity for the accuracy or completeness of any output generated by such software.

Disclaimer

The review and this report are provided on an as-is, where-is, and as-available basis. To the fullest extent permitted by law, Quantstamp disclaims all warranties, expressed or implied, in connection with this report, its content, and the related services and products and your use thereof, including, without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement. You agree that your access and/or use of the report and other results of the review, including but not limited to any associated services, products, protocols, platforms, content, and materials, will be at your sole risk. FOR AVOIDANCE OF DOUBT, THE REPORT, ITS CONTENT, ACCESS, AND/OR USAGE THEREOF, INCLUDING ANY ASSOCIATED SERVICES OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, INVESTMENT, TAX, LEGAL, REGULATORY, OR OTHER ADVICE. This report is based on the scope of materials and documentation provided for a limited review at the time provided. You acknowledge that Blockchain technology remains under development and is subject to unknown risks and flaws and, as such, the report may not be complete or inclusive of all vulnerabilities. The review is limited to the materials identified in the report and does not extend to the compiler layer, or any other areas beyond the programming language, or other programming aspects that could present security risks. The report does not indicate the endorsement by Quantstamp of any particular project or team, nor guarantee its security, and may not be represented as such. No third party is entitled to rely on the report in any way, including for the purpose of making any decisions to buy or sell a product, service or any other asset. Quantstamp does not warrant, endorse, guarantee, or assume responsibility for any product or service advertised or offered by a third party, or any open source or third-party software, code, libraries, materials, or information linked to, called by, referenced by or accessible through the report, its content, or any related services and products, any hyperlinked websites, or any other websites or mobile applications, and we will not be a party to or in any way be responsible for monitoring any transaction between you and any third party. As with the purchase or use of a product or service through any medium or in any environment, you should use your best judgment and exercise caution where appropriate.

© 2024 – Quantstamp, Inc.