HALBORN

Vesper Finance - Strategies

Smart Contract Security Audit

Prepared by: **Halborn**

Date of Engagement: **August 3rd, 2022 - September 5th, 2022**

Visit: **Halborn.com**

DOCUMENT REVISION HISTORY

VERSION	MODIFICATION	DATE	AUTHOR
0.1	Document Creation	08/03/2022	Roberto Reigada
0.2	Document Updates	09/05/2022	Roberto Reigada
0.3	Draft Review	09/06/2022	Gabi Urrutia
1.0	Remediation Plan	09/26/2022	Roberto Reigada
1.1	Remediation Plan Review	09/28/2022	Gabi Urrutia

CONTACTS

CONTACT	COMPANY	EMAIL
Rob Behnke	Halborn	Rob.Behnke@halborn.com
Steven Walbroehl	Halborn	Steven.Walbroehl@halborn.com
Gabi Urrutia	Halborn	Gabi.Urrutia@halborn.com
Roberto Reigada	Halborn	Roberto.Reigada@halborn.com

EXECUTIVE OVERVIEW

1.1 INTRODUCTION

Vesper Finance engaged Halborn to conduct a security audit on their smart contracts beginning on August 3rd, 2022 and ending on September 5th, 2022. The security assessment was scoped to the smart contracts provided in the GitHub repository bloqpriv/vesper-contracts/tree/main/vesper-strategies.

1.2 AUDIT SUMMARY

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

The purpose of this audit is to:

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

In summary, Halborn identified some security risks that were mostly addressed by the Vesper Finance team.

1.3 TEST APPROACH & METHODOLOGY

Halborn performed a combination of manual and automated security testing to balance efficiency, timeliness, practicality, and accuracy in regard to the scope of this audit. While manual testing is recommended to uncover flaws in logic, process, and implementation; automated testing techniques help enhance coverage of the code and can quickly identify items that do not follow the security best practices. The following phases and associated tools were used during the audit:

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

RISK METHODOLOGY:

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

RISK SCALE - LIKELIHOOD

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

RISK SCALE - IMPACT

5 - May cause devastating and unrecoverable impact or loss.
4 - May cause a significant level of impact or loss.

3 - May cause a partial impact or loss to many.
2 - May cause temporary impact or loss.
1 - May cause minimal or un-noticeable impact.

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

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

1.4 SCOPE

IN-SCOPE:
The security assessment was scoped to the following smart contracts:

- VesperMakerStrategy.sol
- AaveV3.sol
- AaveV3VesperXy.sol
- Convex2PlainPool.sol
- Convex3PlainPool.sol
- Convex4FactoryMetaPool.sol
- Convex4MetaPool.sol
- Curve2LendingPool.sol
- Curve3LendingPoolAave.sol
- Curve4FactoryMetaPool.sol
- Curve4PlainOr4MetaPool.sol

Initial commit ID:
- 8da59368bf95cd5ff55ed8ec40d162c27af51e77

2. ASSESSMENT SUMMARY & FINDINGS OVERVIEW

CRITICAL	HIGH	MEDIUM	LOW	INFORMATIONAL
0	2	3	2	3

LIKELIHOOD

IMPACT

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

SECURITY ANALYSIS	RISK LEVEL	REMEDIATION DATE
HAL01 - HARVESTVSP FUNCTION IS NOT CALLED WHEN MIGRATING TO A NEW STRATEGY	High	SOLVED - 09/26/2022
HAL02 - STKAAVE TOKENS ARE NOT TRANSFERRED DURING A MIGRATION IN CURVE3LENDINGPOOLAAVE STRATEGY	High	SOLVED - 09/26/2022
HAL03 - SOME STRATEGIES ARE PRONE TO SANDWICH ATTACKS	Medium	NOT SOLVED
HAL04 - CONVEX STRATEGIES MAY NOT CLAIM CURVE REWARDS WHEN MIGRATING TO A NEW STRATEGY	Medium	SOLVED - 09/26/2022
HAL05 - OLDER CURVE LENDING POOLS DO NOT IMPLEMENT ALL PLAIN POOL METHODS FOR ADDING AND REMOVING LIQUIDITY	Medium	SOLVED - 09/26/2022
HAL06 - DEPRECATED DEPOSIT FUNCTION IS USED IN AAVEV3 STRATEGY	Low	SOLVED - 09/26/2022
HAL07 - EXTREME PRICE FLUCTUATIONS OF THE COLLATERAL OR BORROWED TOKEN COULD CAUSE A LIQUIDATION	Low	SOLVED - 09/26/2022
HAL08 - WRONG COMMENT	Informational	SOLVED - 09/26/2022
HAL9 - USING POSTFIX OPERATORS IN LOOPS	Informational	SOLVED - 09/26/2022
HAL10 - STATE VARIABLES MISSING IMMUTABLE MODIFIER	Informational	SOLVED - 09/26/2022

FINDINGS & TECH DETAILS

3.1 (HAL-01) HARVESTVSP FUNCTION IS NOT CALLED WHEN MIGRATING TO A NEW STRATEGY - HIGH

Description:

The AaveV3VesperXY strategy works the following way:
1. Collateral, for example DAI, is deposited into Aave, aDAI is received.
2. That collateral deposited in Aave is used to borrow some other token, for example WBTC.
3. WBTC is deposited in a Vesper pool.
4. VSP tokens are also received as a yield from the Vesper pool.

On the other hand, in the VesperMakerStrategy:
1. Collateral is deposited into Maker vault.
2. Mint DAI.
3. Deposit DAI in another Vesper pool.
4. Yield earned from the Vesper pool is converted into collateral, increasing the pool token price and the depositor's deposited asset value.
5. VSP tokens are also received as a yield from the Vesper pool.

When a migration is done in the AaveV3VesperXY strategy, these are the steps followed:
1. `_withdrawFromVesperPool(_amount);`
2. `_aaveLendingPool.repay(borrowToken, _amount, 2, address(this));`
3. Transfer receipt token to new strategy contract.
4. Transfer collateral token to new strategy contract.

And for the VesperMakerStrategy strategy, these are the steps performed during a migration:
1. `cm.transferVaultOwnership(_newStrategy);`
2. Transfer receipt token to new strategy contract.
3. Transfer collateral token to new strategy contract.

Both strategies receive VSP tokens as yield from their deposits in the

Vesper pool, but the VSP tokens are not claimed automatically when a migration is done. If the harvestVSP() function is not called before the migration, all the unclaimed VSP tokens will be lost.

Risk Level:

Likelihood - 3
Impact - 5

Recommendation:

It is recommended to call the harvestVSP() function during the migration process in these 2 strategies mentioned.

Remediation Plan:

SOLVED: As per Vesper Finance team's response:
“Migrating a strategy is a very important operation, and we do not want to add unnecessary steps and increase the point of failure. There is a way to avoid this issue by calling manually the function.”

This issue will never appear as long as the Vesper Finance team remembers to call manually the harvestVSP() function before starting a migration.

3.2 (HAL-02) STKAAVE TOKENS ARE NOT TRANSFERRED DURING A MIGRATION IN CURVE3LENDINGPOOLAAVE STRATEGY - HIGH

Description:

The Curve3LendingPoolAave contract contains multiple functions to interact with the AAVE Safety Module. By staking AAVE in the Safety Module, users help protect the protocol from a short fall event and earn an incentive as a result.

When a user stakes AAVE in the Safety Module, the user receives an equivalent amount of stkAAVE in return, and starts accruing rewards in AAVE. The user can then claim the rewards at anytime. To withdraw their staked AAVE, the user needs to activate a coolDown() period.

Currently, during a migration, these would be steps executed by the Curve3LendingPoolAave:

1. _unstakeAllLp()
2. Transfer receipt token to new strategy contract.
3. Transfer collateral token to new strategy contract.

Although the remaining stkAAVE tokens in the contract are never transferred to the new strategy and will remain stuck forever in the old strategy contract after a migration.

Risk Level:

Likelihood - 3
Impact - 5

Recommendation:

It is recommended to implement a _beforeMigration() hook in the Curve3LendingPoolAave contract that transfers all the stkAAVE tokens to the new strategy.

Remediation Plan:

SOLVED: As per Vesper Finance team's response:
“Migrating a strategy is a very important operation and we do not want to add unnecessary steps and increase the point of failure. There is a way to avoid this issue by calling manually the function.”

This issue will never appear as long as the Vesper Finance team remembers to call manually the rebalance() function which will automatically call _claimRewardsAndConvertTo() converting the stkAAVE tokens in the contract to the collateralToken before starting a migration.

3.3 (HAL-03) SOME STRATEGIES ARE PRONE TO SANDWICH ATTACKS - MEDIUM

Description:

Some strategies use a Swapper contract to swap one asset for another. For example, in the AAVEV3 strategies the rewards are claimed and given in aTokens (for example aDAI). These aTokens received as rewards are swapped right away into the collateralToken (DAI) and deposited into Aave to earn compounded interest.

This swap is done with the _minAmountOut parameter set to 1 and hence they are prone to be sandwiched. In that case, the received amount of the swap would be lower and the APY of the strategy would decrease.

The strategies affected are:

- AaveV2Xy.sol
- AaveV3.sol
- AaveV3Xy.sol
- MakerStrategy.sol

The likelihood of this issue, as is proportional to the rewards swapped, will increase if:
- rebalance() function is called after a long period of time (hence the swapped rewards will likely be higher)
- The rewards increase for any particular reason, for example, a higher liquidity in the strategy

Code Location:

AaveV2Xy.sol

Listing 1: AaveV2Xy.sol (Lines 210-212)

```
196 function _rebalance()
197     internal
198     override
199     returns (
200         uint256 _profit,
201         uint256 _loss,
202         uint256 _payback
203     )
204 {
205     uint256 _excessDebt = IVesperPool(pool).excessDebt(address(
 ↳ this));
206     uint256 _totalDebt = IVesperPool(pool).totalDebtOf(address(
 ↳ this));
207
208     // Claim rewards and convert to collateral token
209     uint256 _aaveAmount = _claimAave();
210     if (_aaveAmount > 0) {
211         _safeSwapExactInput(rewardToken, address(collateralToken),
 ↳  _aaveAmount);
212     }
213
214     uint256 _supply = aToken.balanceOf(address(this));
215     uint256 _borrow = vdToken.balanceOf(address(this));
216
217     uint256 _investedBorrowBalance = _getInvestedBorrowBalance();
218
219     // _borrow increases every block. Convert collateral to
 ↳ borrowToken.
220     if (_borrow > _investedBorrowBalance) {
221         _swapToBorrowToken(_borrow - _investedBorrowBalance);
222     } else {
223         // When _investedBorrowBalance exceeds _borrow balance
 ↳ from Aave
224         // Customize this hook to handle the excess borrowToken
 ↳ for profit
225         _rebalanceBorrow(_investedBorrowBalance - _borrow);
226     }
227
228     uint256 _collateralHere = collateralToken.balanceOf(address(
 ↳ this));
229     uint256 _totalCollateral = _supply + _collateralHere;
```

```
230
231     if (_totalCollateral > _totalDebt) {
232         _profit = _totalCollateral - _totalDebt;
233     } else {
234         _loss = _totalDebt - _totalCollateral;
235     }
236     uint256 _profitAndExcessDebt = _profit + _excessDebt;
237     if (_collateralHere < _profitAndExcessDebt) {
238         uint256 _totalAmountToWithdraw = Math.min((
 ↳ _profitAndExcessDebt - _collateralHere), _supply);
239         if (_totalAmountToWithdraw > 0) {
240             _withdrawHere(_totalAmountToWithdraw);
241             _collateralHere = collateralToken.balanceOf(address(
 ↳ this));
242         }
243     }
244
245     // Make sure _collateralHere >= _payback + profit. set actual
 ↳ payback first and then profit
246     _payback = Math.min(_collateralHere, _excessDebt);
247     _profit = _collateralHere > _payback ? Math.min((
 ↳ _collateralHere - _payback), _profit) : 0;
248     IVesperPool(pool).reportEarning(_profit, _loss, _payback);
249     _deposit();
250 }
```

AaveV3.sol

Listing 2: AaveV3.sol (Line 70)

```
64 /// @notice Claim all rewards and convert to _toToken.
65 function _claimRewardsAndConvertTo(address _toToken) internal {
66     (address[] memory _tokens, uint256[] memory _amounts) =
 ↳ AaveV3Incentive._claimRewards(receiptToken);
67     uint256 _length = _tokens.length;
68     for (uint256 i; i < _length; ++i) {
69         if (_amounts[i] > 0) {
70             _safeSwapExactInput(_tokens[i], _toToken, _amounts[i])
 ↳ ;
71         }
72     }
73 }
```

AaveV3Xy.sol

Listing 3: AaveV3Xy.sol (Line 180)

```
175 function _claimRewardsAndConvertTo(address _toToken) internal {
176     (address[] memory _tokens, uint256[] memory _amounts) =
 ↳ AaveV3Incentive._claimRewards(receiptToken);
177     uint256 _length = _tokens.length;
178     for (uint256 i; i < _length; ++i) {
179         if (_amounts[i] > 0) {
180             _safeSwapExactInput(_tokens[i], _toToken, _amounts[i])
 ↳ ;
181         }
182     }
183 }
```

MakerStrategy.sol

Listing 4: MakerStrategy.sol (Line 202)

```
139 function _rebalance()
140     internal
141     override
142     returns (
143         uint256 _profit,
144         uint256 _loss,
145         uint256 _payback
146     )
147 {
148     _payback = IVesperPool(pool).excessDebt(address(this));
149     if (_payback == 0) {
150         // If strategy is suppose to get more fund from pool, this
 ↳  method fetch it.
151         IVesperPool(pool).reportEarning(0, 0, 0);
152     }
153
154     // Deposit available collateral to Maker vault. This will
 ↳ improve collateral ratio
155     uint256 _collateralHere = collateralToken.balanceOf(address(
 ↳ this));
156     if (_collateralHere > 0) {
157         //FIXME: if some collateral sent to this contract , it is
 ↳ not counted as profit and not return to pool. This remains in
 ↳ vault forever.
```

```
158         cm.depositCollateral(_collateralHere);
159     }
160     uint256 _totalDebt = IVesperPool(pool).totalDebtOf(address(
 ↳ this));
161     uint256 _collateralInVault = cm.getVaultBalance(address(this))
 ↳ ;
162     // When strategy was making loss in DAI and resurface() method
 ↳  called then it reduce collateral in vault. Ref _resurface();
163     if (_totalDebt > _collateralInVault) {
164         _loss = _totalDebt - _collateralInVault;
165     }
166
167     (uint256 _daiToRepay, uint256 _daiToBorrow, uint256
 ↳ _currentDaiDebt) = _calculateSafeBorrowPosition(_payback);
168
169     uint256 _daiBalance = _getDaiBalance();
170     // This contract is not suppose to hold any borrowed DAI. If
 ↳ any DAI received from rewards, donation etc is profit.
171     uint256 _profitInDai;
172     uint256 _daiToWithdraw = _daiToRepay;
173     if (_daiBalance > _currentDaiDebt) {
174         // Yield generated in DAI.
175         _profitInDai = _daiBalance - _currentDaiDebt;
176         _daiToWithdraw += _profitInDai;
177     }
178     // Contract may have some DAI here from rewards or from Maker.
 ↳  Use this DAI for profit.
179     _profitInDai += IERC20(DAI).balanceOf(address(this));
180     if (_daiToWithdraw > 0) {
181         // This can withdraw less than requested amount.  This is
 ↳ not problem as long as Dai here >= _daiToRepay. Profit earned in
 ↳ DAI can be reused for _daiToRepay.
182         _withdrawDaiFromLender(_daiToWithdraw);
183     }
184
185     if (_daiToRepay > 0) {
186         cm.payback(_daiToRepay);
187     } else if (_daiToBorrow > 100e18) {
188         cm.borrow(_daiToBorrow);
189     }
190     // Dai paid back by now. Good to withdraw excessDebt in
 ↳ collateral.
191     if (_payback > 0) {
192         cm.withdrawCollateral(_payback);
```

```
193     }
194
195     // DAI: profit DAI, borrowed DAI, repay DAI.
196     // Collateral token: excessDebt, profit in collateral if any
197
198     // All remaining dai here is not profit. some part is borrowed
 ↳  dai.
199     _profitInDai = Math.min(_profitInDai, IERC20(DAI).balanceOf(
 ↳ address(this)));
200     if (_profitInDai > 0) {
201         // calling safeSwap to not revert in case profit
 ↳ conversion to collateralToken fails. Let Dai remains here. It
 ↳ doesn't harm overall.
202         _safeSwapExactInput(DAI, address(collateralToken),
 ↳ _profitInDai);
203     }
204     // Remaining Dai here are actually borrowed DAI or leftover
 ↳ profit. Deposit it for yield generation
205     uint256 _daiBalanceHere = IERC20(DAI).balanceOf(address(this))
 ↳ ;
206
207     if (_daiBalanceHere > 0) {
208         _depositDaiToLender(_daiBalanceHere);
209     }
210
211     _collateralHere = collateralToken.balanceOf(address(this));
212     if (_collateralHere > _payback) {
213         _profit = _collateralHere - _payback;
214     }
215
216     // Adjust past loss and current profit here itself if possible
 ↳ .
217     if (_profit >= _loss) {
218         _profit = _profit - _loss;
219         _loss = 0;
220     } else {
221         // Loss > profit
222         _loss = _loss - _profit;
223         _profit = 0;
224     }
225
226     // Pool expect this contract has _profit + _payback in the
 ↳ contract. This method would revert if collateral.balanceOf(
 ↳ strategy) < (_profit + _excessDebt);
```

```
227     IVesperPool(pool).reportEarning(_profit, _loss, _payback);
228 }
```

Listing 5: Strategy.sol (Line 223)

```
218 function _safeSwapExactInput(
219     address _tokenIn,
220     address _tokenOut,
221     uint256 _amountIn
222 ) internal {
223     try swapper.swapExactInput(_tokenIn, _tokenOut, _amountIn, 1,
 ↳ address(this)) {} catch {} //solhint-disable no-empty-blocks
224 }
```

Risk Level:

Likelihood - 1
Impact - 5

Recommendation:

1. Before rebalance() is called, a view function that returns the rewards should be called. With the amount of rewards received, getMinAmountOut() function should be called.
2. Once the return value of the getMinAmountOut() is received, rebalance (_minAmountOut) can be called by passing this value as parameter. This way, the rebalance calls cannot ever be sandwiched.

Remediation Plan:

NOT SOLVED: As per Vesper Finance team's response:
“We will address this in the future as there is no huge fund risk associated with this issue, we do not want to overhaul the design for all the strategies at this moment. We will revisit this in the future.”

This issue can be prevented by calling frequently enough the rebalance() functions. Halborn has reviewed most of the Vesper pools that are using

some strategies prone to sandwich attacks and currently none of them are having this issue as the Vesper Finance team calls periodically the rebalance() function.

3.4 (HAL-04) CONVEX STRATEGIES MAY NOT CLAIM CURVE REWARDS WHEN MIGRATING TO A NEW STRATEGY - MEDIUM

Description:

When a migration is done in a Convex strategy, these are the steps followed:

1. `_unstakeAllLp();`
2. Transfer `receipt` token to new strategy contract.
3. Transfer `collateral` token to new strategy contract.

Although, when the LPs are unstaked, the rewards may not be claimed if the `isClaimRewards` state variable is set to `false`:

Listing 6: Convex2PlainPool.sol (Line 50)

```
49      function _unstakeAllLp() internal override {
50          cvxCrvRewards.withdrawAllAndUnwrap(isClaimRewards);
51      }
```

If the migration is done when the `isClaimRewards` is set to `false` all the unclaimed Curve rewards will be lost.

Risk Level:

Likelihood - 3
Impact - 3

Recommendation:

It is recommended to force the claim of the rewards when calling `_unstakeAllLp()` during a migration.

Remediation Plan:

SOLVED: As per Vesper Finance team's response:
“Migrating a strategy is a very important operation, and we do not want to add unnecessary steps and increase the point of failure. There is a way to avoid this issue by manually the function.”

This issue will never appear as long as the Vesper Finance team remembers to call manually claiming the rewards before starting a migration.

3.5 (HAL-05) OLDER CURVE LENDING POOLS DO NOT IMPLEMENT ALL PLAIN POOL METHODS FOR ADDING AND REMOVING LIQUIDITY - MEDIUM

Description:

The contracts Curve2LendingPool and Curve3LendingPool use the following functions to add and remove liquidity from the Curve lending pools:

Listing 7: Curve2LendingPool.sol (Lines 27,37)

```
20 function _depositToCurve(uint256 coinAmountIn_) internal virtual
↳ override {
21     if (coinAmountIn_ > 0) {
22         uint256[2] memory _depositAmounts;
23         _depositAmounts[collateralIdx] = coinAmountIn_;
24 
25         uint256 _lpAmountOutMin = _calculateAmountOutMin(address(
↳ collateralToken), address(crvLp), coinAmountIn_);
26         // Note: Using use_underlying = true to deposit underlying
↳  instead of IB token
27         IStableSwap2xUnderlying(crvPool).add_liquidity(
↳ _depositAmounts, _lpAmountOutMin, true);
28     }
29 }
30 
31 function _withdrawFromCurve(
32     uint256 lpAmount_,
33     uint256 minAmountOut_,
34     int128 i_
35 ) internal override {
36     // Note: Using use_underlying = true to withdraw underlying
↳ instead of IB token
37     IStableSwap2xUnderlying(crvPool).remove_liquidity_one_coin(
↳ lpAmount_, i_, minAmountOut_, true);
38 }
```

Listing 8: Curve3LendingPool.sol (Lines 33,43)

```
26 function _depositToCurve(uint256 coinAmountIn_) internal override
↳ {
27     if (coinAmountIn_ > 0) {
28         uint256[3] memory _depositAmounts;
29         _depositAmounts[collateralIdx] = coinAmountIn_;
30 
31         uint256 _lpAmountOutMin = _calculateAmountOutMin(address(
↳ collateralToken), address(crvLp), coinAmountIn_);
32         // Note: Using use_underlying = true to deposit underlying
↳  instead of IB token
33         IStableSwap3xUnderlying(crvPool).add_liquidity(
↳ _depositAmounts, _lpAmountOutMin, true);
34     }
35 }
36 
37 function _withdrawFromCurve(
38     uint256 lpAmount_,
39     uint256 minAmountOut_,
40     int128 i_
41 ) internal override {
42     // Note: Using use_underlying = true to withdraw underlying
↳ instead of IB token
43     IStableSwap3xUnderlying(crvPool).remove_liquidity_one_coin(
↳ lpAmount_, i_, minAmountOut_, true);
44 }
```

As per the Curve documentation:

> **Warning**
>
> Older Curve lending pools (e.g., Compound Pool) **do not** implement all plain pool methods for adding and removing liquidity. For instance, `remove_liquidity_one_coin` is not implemented by Compound Pool).

This means that for example, if the Compound Pool was used in the Curve2LendingPool or Curve3LendingPool strategies, the strategies would never be able to remove the liquidity from Curve as the remove_liquidity_one_coin() call would always revert.

Currently, Curve supports the following lending pools:

- Aave: Aave pool, with lending on Aave: Implements add_liquidity(uint256[], uint256, bool) and remove_liquidity_one_coin(uint256, int128, uint256, bool) functions.
- BUSD: BUSD pool, with lending on yearn.finance: Does not implement add_liquidity(uint256[], uint256, bool) nor remove_liquidity_one_coin(uint256, int128, uint256, bool) functions.
- Compound: Compound pool, with lending on Compound: Does not implement add_liquidity(uint256[], uint256, bool) nor remove_liquidity_one_coin(uint256, int128, uint256, bool) functions.
- IB: Iron Bank pool, with lending on Cream: Implements add_liquidity(uint256[], uint256, bool) and remove_liquidity_one_coin(uint256, int128, uint256, bool) functions.
- PAX: PAX pool, with lending on yearn.finance: Does not implement add_liquidity(uint256[], uint256, bool) nor remove_liquidity_one_coin(uint256, int128, uint256, bool) functions.
- USDT: USDT pool, with lending on Compound: Does not implement add_liquidity(uint256[], uint256, bool) nor remove_liquidity_one_coin(uint256, int128, uint256, bool) functions.
- Y: Y pool, with lending on yearn.finance: Does not implement add_liquidity(uint256[], uint256, bool) nor remove_liquidity_one_coin(uint256, int128, uint256, bool) functions.

Based on the information above, currently only the Aave and the Iron Bank lending pools would be compatible with the Curve2LendingPool and Curve3LendingPool strategies.

Risk Level:

Likelihood - 1
Impact - 5

Recommendation:

If the strategies are planned to be used with any Curve Lending pool, it is recommended to add multiple try/catch code blocks where the different functions to add/remove liquidity are used.

Remediation Plan:

SOLVED: The Vesper Finance team corrected this issue and now supports all the Curve Lending pool by making use of the Curve Deposit contracts

3.6 (HAL-06) DEPRECATED DEPOSIT FUNCTION IS USED IN AAVEV3 STRATEGY - LOW

Description:

All the AaveV3 strategies are calling the Pool.deposit() function:

Listing 9: AaveV3.sol (Line 112)

```
74 function _rebalance()
75     internal
76     override
77     returns (
78         uint256 _profit,
79         uint256 _loss,
80         uint256 _payback
81     )
82 {
83     uint256 _excessDebt = IVesperPool(pool).excessDebt(address(
↳ this));
84     uint256 _totalDebt = IVesperPool(pool).totalDebtOf(address(
↳ this));
85
86     // Claim any reward we have.
87     _claimRewardsAndConvertTo(address(collateralToken));
88
89     uint256 _collateralHere = collateralToken.balanceOf(address(
↳ this));
90
91     uint256 _totalCollateral = IERC20(receiptToken).balanceOf(
↳ address(this)) + _collateralHere;
92
93     if (_totalCollateral > _totalDebt) {
94         _profit = _totalCollateral - _totalDebt;
95     } else {
96         _loss = _totalDebt - _totalCollateral;
97     }
98     uint256 _profitAndExcessDebt = _profit + _excessDebt;
99     if (_profitAndExcessDebt > _collateralHere) {
100         _withdrawHere(_profitAndExcessDebt - _collateralHere);
```

```
101         _collateralHere = collateralToken.balanceOf(address(this))
 ↳ ;
102     }
103
104     // Make sure _collateralHere >= _payback + profit. set actual
 ↳ payback first and then profit
105     _payback = Math.min(_collateralHere, _excessDebt);
106     _profit = _collateralHere > _payback ? Math.min((
 ↳ _collateralHere - _payback), _profit) : 0;
107     IVesperPool(pool).reportEarning(_profit, _loss, _payback);
108
109     // strategy may get new fund. deposit to generate yield
110     _collateralHere = collateralToken.balanceOf(address(this)); //
 ↳ DAI
111     if (_collateralHere > 0) {
112         AaveLendingPool(aaveAddressProvider.getPool()).deposit(
113             address(collateralToken),
114             _collateralHere,
115             address(this),
116             0
117         );
118     }
119 }
```

Listing 10: AaveV3Xy.sol (Line 190)

```
188 function _depositToAave(uint256 _amount, AaveLendingPool
 ↳ _aaveLendingPool) internal virtual {
189     if (_amount > 0) {
190         try _aaveLendingPool.deposit(address(collateralToken),
 ↳ _amount, address(this), 0) {} catch Error(
191             string memory _reason
192         ) {
193             // Aave uses liquidityIndex and some other indexes as
 ↳ needed to normalize input.
194             // If normalized input equals to 0 then error will be
 ↳ thrown with '56' error code.
195             // CT_INVALID_MINT_AMOUNT = '56'; //invalid amount to
 ↳ mint
196             // Hence discard error where error code is '56'
197             require(bytes32(bytes(_reason)) == "56", "deposit
 ↳ failed");
198         }
199     }
```

```
200 }
```

As we can see in the picture below, the function Pool.deposit() has been deprecated in favour of Pool.supply():

C-Chain: 0xdf9e4abdbd941079322653194796 43d3b05809dc

As the Pool.deposit() function is currently deprecated, we do not have any guarantee that this function will be kept in the future.

Risk Level:

Likelihood - 1
Impact - 3

Recommendation:

It is recommended to use the Pool.supply() function instead of Pool.deposit() in all the AaveV3 strategies.

Remediation Plan:

SOLVED: Vesper Finance team corrected the issue and now uses supply() instead of deposit().

3.7 (HAL-07) EXTREME PRICE FLUCTUATIONS OF THE COLLATERAL OR BORROWED TOKEN COULD CAUSE A LIQUIDATION - LOW

Description:

In the AaveV3VesperXy strategy, the rewards received from Aave are swapped into the collateral token and then into borrow tokens. These borrow tokens are then deposited in a Vesper pool to generate extra yield. In order to prevent a bad debt and hence, being liquidated in AAVE, a maxBorrowLimit and minBorrowLimit are set in the smart contract and used in the _calculateBorrowPosition() function:

Listing 11: AaveV3Xy.sol (Lines 156-159)

```
112 function _calculateBorrowPosition(
113     uint256 _depositAmount,
114     uint256 _withdrawAmount,
115     uint256 _borrowed,
116     uint256 _supplied
117 ) internal view returns (uint256 _borrowAmount, uint256
 ↳ _repayAmount) {
118     require(_depositAmount == 0 || _withdrawAmount == 0, "all-
 ↳ input-gt-zero");
119     // If maximum borrow limit set to 0 then repay borrow
120     if (maxBorrowLimit == 0) {
121         return (0, _borrowed);
122     }
123     // In case of withdraw, _amount can be greater than _supply
124     uint256 _hypotheticalCollateral =
125         _depositAmount > 0 ? _supplied + _depositAmount :
 ↳ _supplied > _withdrawAmount
126             ? _supplied - _withdrawAmount
127             : 0;
128     if (_hypotheticalCollateral == 0) {
129         return (0, _borrowed);
130     }
```

```
131     AaveOracle _aaveOracle = AaveOracle(aaveAddressProvider.
 ↳ getPriceOracle());
132     // Oracle prices are in 18 decimal
133     uint256 _borrowTokenPrice = _aaveOracle.getAssetPrice(
 ↳ borrowToken);
134     uint256 _collateralTokenPrice = _aaveOracle.getAssetPrice(
 ↳ address(collateralToken));
135     if (_borrowTokenPrice == 0 || _collateralTokenPrice == 0) {
136         // Oracle problem. Lets payback all
137         return (0, _borrowed);
138     }
139     // _collateralFactor in 4 decimal. 10_000 = 100%
140     (, uint256 _collateralFactor, , , , , , , ) =
141         AaveProtocolDataProvider(aaveAddressProvider.
 ↳ getPoolDataProvider()).getReserveConfigurationData(
142             address(collateralToken)
143         );
144
145     // Collateral in base currency based on oracle price and cf;
146     uint256 _actualCollateralForBorrow =
147         (_hypotheticalCollateral * _collateralFactor *
 ↳ _collateralTokenPrice) /
148             (MAX_BPS * (10**IERC20Metadata(address(collateralToken
 ↳ )).decimals()));
149     // Calculate max borrow possible in borrow token number
150     uint256 _maxBorrowPossible =
151         (_actualCollateralForBorrow * (10**IERC20Metadata(address(
 ↳ borrowToken)).decimals())) / _borrowTokenPrice;
152     if (_maxBorrowPossible == 0) {
153         return (0, _borrowed);
154     }
155     // Safe buffer to avoid liquidation due to price variations.
156     uint256 _borrowUpperBound = (_maxBorrowPossible *
 ↳ maxBorrowLimit) / MAX_BPS;
157
158     // Borrow up to _borrowLowerBound and keep buffer of
 ↳ _borrowUpperBound - _borrowLowerBound for price variation
159     uint256 _borrowLowerBound = (_maxBorrowPossible *
 ↳ minBorrowLimit) / MAX_BPS;
160
161     // If current borrow is greater than max borrow, then repay to
 ↳  achieve safe position.
162     if (_borrowed > _borrowUpperBound) {
```

```
163         // If borrow > upperBound then it is greater than
 ↳ lowerBound too.
164         _repayAmount = _borrowed - _borrowLowerBound;
165     } else if (_borrowLowerBound > _borrowed) {
166         _borrowAmount = _borrowLowerBound - _borrowed;
167         uint256 _availableLiquidity = IERC20(borrowToken).
 ↳ balanceOf(aBorrowToken);
168         if (_borrowAmount > _availableLiquidity) {
169             _borrowAmount = _availableLiquidity;
170         }
171     }
172 }
```

Although a drastic price change of the collateral or the borrowed token could cause a liquidation especially if rebalance() or withdraw() is not called for a long period of time.

Risk Level:

Likelihood - 1
Impact - 3

Recommendation:

It is recommended to monitor the prices of the collateral and the borrowed token frequently offchain and call rebalance() in case of a drastic price change.

Remediation Plan:

SOLVED: The Vesper Finance team has a monitoring system in place to identify such situations, in case they occur.

3.8 (HAL-08) WRONG COMMENT - INFORMATIONAL

Description:

In the AaveV3Xy contract, it is mentioned in the _calculateBorrowPosition() function that oracle prices are in 18 decimals, although this is not always true, especially in Avalanche:

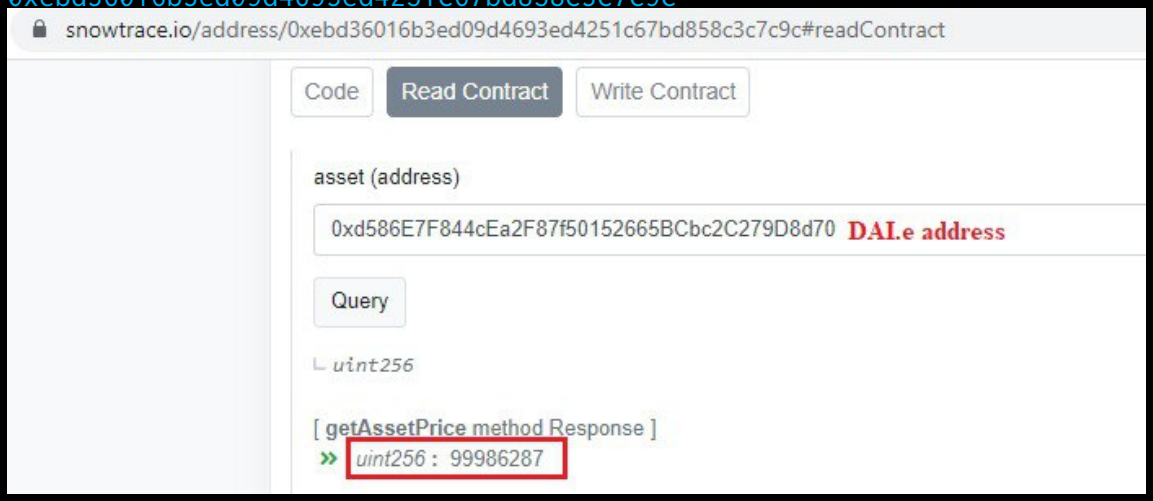
Listing 12: AaveV3Xy.sol (Line 132)

```
112 function _calculateBorrowPosition(
113     uint256 _depositAmount,
114     uint256 _withdrawAmount,
115     uint256 _borrowed,
116     uint256 _supplied
117 ) internal view returns (uint256 _borrowAmount, uint256
 ↳ _repayAmount) {
118     require(_depositAmount == 0 || _withdrawAmount == 0, "all-
 ↳ input-gt-zero");
119     // If maximum borrow limit set to 0 then repay borrow
120     if (maxBorrowLimit == 0) {
121         return (0, _borrowed);
122     }
123     // In case of withdraw, _amount can be greater than _supply
124     uint256 _hypotheticalCollateral =
125         _depositAmount > 0 ? _supplied + _depositAmount :
 ↳ _supplied > _withdrawAmount
126             ? _supplied - _withdrawAmount
127             : 0;
128     if (_hypotheticalCollateral == 0) {
129         return (0, _borrowed);
130     }
131     AaveOracle _aaveOracle = AaveOracle(aaveAddressProvider.
 ↳ getPriceOracle());
132     // Oracle prices are in 18 decimal
133     uint256 _borrowTokenPrice = _aaveOracle.getAssetPrice(
 ↳ borrowToken);
134     uint256 _collateralTokenPrice = _aaveOracle.getAssetPrice(
 ↳ address(collateralToken));
135     if (_borrowTokenPrice == 0 || _collateralTokenPrice == 0) {
136         // Oracle problem. Lets payback all
```

```
137         return (0, _borrowed);
138     }
139     // _collateralFactor in 4 decimal. 10_000 = 100%
140     (, uint256 _collateralFactor, , , , , , , ) =
141         AaveProtocolDataProvider(aaveAddressProvider.
 ↳ getPoolDataProvider()).getReserveConfigurationData(
142             address(collateralToken)
143         );
144
145     // Collateral in base currency based on oracle price and cf;
146     uint256 _actualCollateralForBorrow =
147         (_hypotheticalCollateral * _collateralFactor *
 ↳ _collateralTokenPrice) /
148             (MAX_BPS * (10**IERC20Metadata(address(collateralToken
 ↳ )).decimals()));
149     // Calculate max borrow possible in borrow token number
150     uint256 _maxBorrowPossible =
151         (_actualCollateralForBorrow * (10**IERC20Metadata(address(
 ↳ borrowToken)).decimals())) / _borrowTokenPrice;
152     if (_maxBorrowPossible == 0) {
153         return (0, _borrowed);
154     }
155     // Safe buffer to avoid liquidation due to price variations.
156     uint256 _borrowUpperBound = (_maxBorrowPossible *
 ↳ maxBorrowLimit) / MAX_BPS;
157
158     // Borrow up to _borrowLowerBound and keep buffer of
 ↳ _borrowUpperBound - _borrowLowerBound for price variation
159     uint256 _borrowLowerBound = (_maxBorrowPossible *
 ↳ minBorrowLimit) / MAX_BPS;
160
161     // If current borrow is greater than max borrow, then repay to
 ↳  achieve safe position.
162     if (_borrowed > _borrowUpperBound) {
163         // If borrow > upperBound then it is greater than
 ↳ lowerBound too.
164         _repayAmount = _borrowed - _borrowLowerBound;
165     } else if (_borrowLowerBound > _borrowed) {
166         _borrowAmount = _borrowLowerBound - _borrowed;
167         uint256 _availableLiquidity = IERC20(borrowToken).
 ↳ balanceOf(aBorrowToken);
168         if (_borrowAmount > _availableLiquidity) {
169             _borrowAmount = _availableLiquidity;
170         }
```

```
171     }
172 }
```

0xebd36016b3ed09d4693ed4251c67bd858c3c7c9c



Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

It is recommended to correct this comment as it may lead to incorrect future implementations.

Remediation Plan:

SOLVED: The Vesper Finance team removed the suggested comment.

3.9 (HAL-09) USING POSTFIX OPERATORS IN LOOPS - INFORMATIONAL

Description:

In the loops below, postfix (e.g. i++) operators were used to increment or decrement variable values. In loops, using prefix operators (e.g. ++i) costs less gas per iteration than using postfix operators.

Code Location:

ConvexBase.sol
- Line 66:
`for (uint256 i; i < _length; i++){`
- Line 80:
`for (uint256 i; i < _length; i++){`

CurvePoolBase.sol
- Line 149:
`for (uint256 i; i < _rewardTokensLength; i++){`
- Line 187:
`for (uint256 i; i < _rewardTokensLength; i++){`
- Line 350:
`for (uint256 i; i < _rewardTokensLength; i++){`

Proof of Concept:

For example, based in the following test contract:

Listing 13: Test.sol

```
1 //SPDX-License-Identifier: MIT
2 pragma solidity 0.8.9;
3
4 contract test {
5     function postiincrement(uint256 iterations) public {
6         for (uint256 i = 0; i < iterations; i++) {
```

```
        }
    }
    function preiincrement(uint256 iterations) public {
        for (uint256 i = 0; i < iterations; ++i) {
        }
    }
}
```

We can see the difference in the gas costs:

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

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

<Transaction '0x205f09a4d2268de4c1a40f35bb2ec2847bf2ab8d584909b42c71a022b047614a'>
>>> test_contract.postiincrement(10)
Transaction sent: 0x98c04430526a59balcf947c114b62666a4417165947d3lbf300cd6ae68328033
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 46
  test.postiincrement confirmed   Block: 13622337   Gas used: 22673 (0.34%)

<Transaction '0x98c04430526a59balcf947c114b62666a4417165947d3lbf300cd6ae68328033'>
>>> test_contract.preiincrement(10)
Transaction sent: 0xf060d04714eff8482a828342414d5a20be9958c822d42860e7992aba20elde05
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 47
  test.preiincrement confirmed   Block: 13622338   Gas used: 22601 (0.34%)

<Transaction '0xf060d04714eff8482a828342414d5a20be9958c822d42860e7992aba20elde05'>
```

Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

It is recommended to use ++i instead of i++ to increment the value of an uint variable inside a loop. This does not only apply to the iterator variable. It also applies to the increments/decrements done inside the loop code block.

Remediation Plan:

SOLVED: The Vesper Finance team corrected this issue and now uses ++i instead of i++ to increment the value of an uint variable inside loops.

3.10 (HAL-10) STATE VARIABLES MISSING IMMUTABLE MODIFIER - INFORMATIONAL

Description:

Some state variables can be declared as immutable to reduce the gas costs.

The immutable keyword was added to Solidity in 0.6.5. State variables can be marked immutable which causes them to be read-only, but only assignable in the constructor.

Code Location:

AaveV3.sol
- Line 14: string public NAME;

AaveV3Xy.sol
- Line 18: string public NAME;

CurvePoolBase.sol
- Line 37: address public CRV = 0xD533a949740bb3306d119CC777fa900bA034cd52
;
- Line 39: string public NAME;

MakerStrategy.sol
- Line 16: string public NAME;
- Line 24: uint256 public decimalConversionFactor;

Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

It is recommended to add the immutable modifier to the state variables described.

Remediation Plan:

SOLVED: The Vesper Finance team corrected the issue and added the immutable modifier to the state variables suggested.

AUTOMATED TESTING

4.1 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.

Slither results:

VesperMakerStrategy.sol

```
VesperMakerStrategy._withdrawDaiFromLender(uint256) (contracts/contracts/strategies/maker/VesperMakerStrategy.sol#53-59) performs a multiplication on the result of a division:
        -_share = (_amount * 1e18) / _pricePerShare (contracts/contracts/strategies/maker/VesperMakerStrategy.sol#55)
        -_amount > ((_share * _pricePerShare) / 1e18) (contracts/contracts/strategies/maker/VesperMakerStrategy.sol#57)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#divide-before-multiply

MakerStrategy._rebalance().profitInDai (contracts/contracts/strategies/maker/MakerStrategy.sol#171) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

Strategy._swapExactInput(address,address,uint256) (contracts/contracts/strategies/Strategy.sol#210-216) ignores return value by swapper.swapExactInput(_tokenIn,_tokenOut,_amountIn,1,address(this)) (contracts/contracts/strategies/Strategy.sol#215)
Strategy._safeSwapExactInput(address,address,uint256) (contracts/contracts/strategies/Strategy.sol#218-224) ignores return value by swapper.swapExactInput(_tokenIn,_tokenOut,_amountIn,1,address(this)) (contracts/contracts/strategies/Strategy.sol#223)
MakerStrategy._resurface() (contracts/contracts/strategies/maker/MakerStrategy.sol#230-240) ignores return value by swapper.swapExactOutput(address(collateralToken),DAI,_daiNeeded,_collateralNeeded,address(this)) (contracts/contracts/strategies/maker/MakerStrategy.sol#236)
MakerStrategy.createVault() (contracts/contracts/strategies/maker/MakerStrategy.sol#277-279) ignores return value by cm.createVault(collateralType) (contracts/contracts/strategies/maker/MakerStrategy.sol#278)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

Address.isContract(address) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#26-35) uses assembly
        - INLINE ASM (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#33)
Address._verifyCallResult(bool,bytes,string) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#171-188) uses assembly
        - INLINE ASM (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#180-183)
EnumerableSet.values(EnumerableSet.AddressSet) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#273-282) uses assembly
        - INLINE ASM (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#277-279)
EnumerableSet.values(EnumerableSet.UintSet) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#346-355) uses assembly
        - INLINE ASM (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#350-352)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Different versions of Solidity is used:
        - Version used: ['^0.8.9', '^0.8.0']
        - 0.8.9 (contracts/contracts/interfaces/swapper/IRoutedSwapper.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/vesper/ICollateralManager.sol#3)
        - 0.8.9 (contracts/contracts/strategies/Strategy.sol#3)
        - 0.8.9 (contracts/contracts/strategies/maker/MakerStrategy.sol#3)
        - 0.8.9 (contracts/contracts/strategies/maker/VesperMakerStrategy.sol#3)
        - 0.8.9 (vesper-commons/contracts/interfaces/vesper/IStrategy.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/IERC20.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Context.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/Math.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#3)
        - 0.8.9 (vesper-pools/contracts/interfaces/vesper/IGovernable.sol#3)
        - 0.8.9 (vesper-pools/contracts/interfaces/vesper/IPausable.sol#3)
        - 0.8.9 (vesper-pools/contracts/interfaces/vesper/IPoolRewards.sol#3)
        - 0.8.9 (vesper-pools/contracts/interfaces/vesper/IVesperPool.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

Address.functionCall(address,bytes) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#79-81) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#104-106) is never used and should be removed
Address.functionDelegateCall(address,bytes) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#153-155) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#163-169) is never used and should be removed
Address.functionStaticCall(address,bytes) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#125-131) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#139-145) is never used and should be removed
Context._msgData() (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Context.sol#20-23) is never used and should be removed
EnumerableSet._at(EnumerableSet.Set,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#125-131) is never used and should be removed
EnumerableSet._length(EnumerableSet.Set) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#115-117) is never used and should be removed
EnumerableSet.add(EnumerableSet.Bytes32Set,bytes32) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#157-159) is never used and should be removed
EnumerableSet.add(EnumerableSet.UintSet,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#296-298) is never used and should be removed
EnumerableSet.at(EnumerableSet.AddressSet,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#261-263) is never used and should be removed
EnumerableSet.at(EnumerableSet.Bytes32Set,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#195-197) is never used and should be removed
EnumerableSet.at(EnumerableSet.UintSet,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#334-336) is never used and should be removed
EnumerableSet.contains(EnumerableSet.Bytes32Set,bytes32) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#174-176) is never used and should be removed
EnumerableSet.contains(EnumerableSet.UintSet,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#313-315) is never used and should be removed
EnumerableSet.length(EnumerableSet.AddressSet) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#247-249) is never used and should be removed
EnumerableSet.length(EnumerableSet.Bytes32Set) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#181-183) is never used and should be removed
EnumerableSet.length(EnumerableSet.UintSet) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#320-322) is never used and should be removed
EnumerableSet.remove(EnumerableSet.Bytes32Set,bytes32) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#167-169) is never used and should be removed
EnumerableSet.remove(EnumerableSet.UintSet,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#306-308) is never used and should be removed
EnumerableSet.values(EnumerableSet.Bytes32Set) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#207-209) is never used and should be removed
EnumerableSet.values(EnumerableSet.UintSet) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#346-355) is never used and should be removed
MakerStrategy._depositDaiToLender(uint256) (contracts/contracts/strategies/maker/MakerStrategy.sol#128) is never used and should be removed
MakerStrategy._getDaiBalance() (contracts/contracts/strategies/maker/MakerStrategy.sol#130) is never used and should be removed
MakerStrategy._withdrawDaiFromLender(uint256) (contracts/contracts/strategies/maker/MakerStrategy.sol#249) is never used and should be removed
Math.average(uint256,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/Math.sol#27-30) is never used and should be removed
Math.max(uint256,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/Math.sol#12-14) is never used and should be removed
SafeERC20.safeDecreaseAllowance(IERC20,address,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#51-58) is never used and should be removed
SafeERC20.safeIncreaseAllowance(IERC20,address,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#46-49) is never used and should be removed
SafeERC20.safeTransferFrom(IERC20,address,address,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#24-26) is never used and should be removed
Strategy._beforeMigration(address) (contracts/contracts/strategies/Strategy.sol#199) is never used and should be removed
Strategy._rebalance() (contracts/contracts/strategies/Strategy.sol#201-208) is never used and should be removed
Strategy._withdrawHere(uint256) (contracts/contracts/strategies/Strategy.sol#227) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
```

```
Pragma version0.8.9 (contracts/contracts/interfaces/swapper/IRoutedSwapper.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/vesper/ICollateralManager.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/strategies/Strategy.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/strategies/maker/MakerStrategy.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/strategies/maker/VesperMakerStrategy.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (vesper-commons/contracts/interfaces/vesper/IStrategy.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/IERC20.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Context.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/Math.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#3) allows old versions
Pragma version0.8.9 (vesper-pools/contracts/interfaces/vesper/IGovernable.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (vesper-pools/contracts/interfaces/vesper/IPausable.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (vesper-pools/contracts/interfaces/vesper/IPoolRewards.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (vesper-pools/contracts/interfaces/vesper/IVesperPool.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
solc-0.8.9 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

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

Parameter Strategy.addKeeper(address)._keeperAddress (contracts/contracts/strategies/Strategy.sol#64) is not in mixedCase
Parameter Strategy.migrate(address)._newStrategy (contracts/contracts/strategies/Strategy.sol#87) is not in mixedCase
Parameter Strategy.removeKeeper(address)._keeperAddress (contracts/contracts/strategies/Strategy.sol#119) is not in mixedCase
Parameter Strategy.sweepERC20(address)._fromToken (contracts/contracts/strategies/Strategy.sol#127) is not in mixedCase
Parameter Strategy.updateFeeCollector(address)._feeCollector (contracts/contracts/strategies/Strategy.sol#156) is not in mixedCase
Parameter Strategy.updateSwapper(IRoutedSwapper)._swapper (contracts/contracts/strategies/Strategy.sol#167) is not in mixedCase
Parameter Strategy.withdraw(uint256)._amount (contracts/contracts/strategies/Strategy.sol#178) is not in mixedCase
Parameter MakerStrategy.convertFrom18(uint256)._amount (contracts/contracts/strategies/maker/MakerStrategy.sol#48) is not in mixedCase
Parameter MakerStrategy.isReservedToken(address)._token (contracts/contracts/strategies/maker/MakerStrategy.sol#53) is not in mixedCase
Parameter MakerStrategy.updateBalancingFactor(uint256,uint256)._highWater (contracts/contracts/strategies/maker/MakerStrategy.sol#301) is not in mixedCase
Parameter MakerStrategy.updateBalancingFactor(uint256,uint256)._lowWater (contracts/contracts/strategies/maker/MakerStrategy.sol#301) is not in mixedCase
Variable MakerStrategy.NAME (contracts/contracts/strategies/maker/MakerStrategy.sol#16) is not in mixedCase
Function IStrategy.VERSION() (vesper-commons/contracts/interfaces/vesper/IStrategy.sol#31) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

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

AaveV3.sol

```
AaveV3._withdrawHere(uint256) (contracts/contracts/strategies/aave/v3/AaveV3.sol#122-140) uses a dangerous strict equality:
        - require(bool,string)(IAaveLendingPool(aaveAddressProvider.getPool()).withdraw(address(collateralToken),_possibleWithdraw,address(this)) == _possibleWithdraw,Errors.INCORRECT_WITHDRAW_AMOUNT) (contracts/contracts/strategies/aave/v3/AaveV3.sol#131-138)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dangerous-strict-equalities

AaveV3._claimRewardsAndConvertTo(address).i (contracts/contracts/strategies/aave/v3/AaveV3.sol#67) is a local variable never initialized
AaveV3._approveToken(uint256)._aaveIncentivesController (contracts/contracts/strategies/aave/v3/AaveV3.sol#51) is a local variable never initialized
AaveV3._approveToken(uint256).i (contracts/contracts/strategies/aave/v3/AaveV3.sol#53) is a local variable never initialized
AaveV3Incentive._claimRewards(address)._aaveIncentivesController (contracts/contracts/strategies/aave/v3/AaveV3Incentive.sol#16) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

Strategy._swapExactInput(address,address,uint256) (contracts/contracts/strategies/Strategy.sol#210-216) ignores return value by swapper.swapExactInput(_tokenIn,_tokenOut,_amountIn,1,address(this)) (contracts/contracts/strategies/Strategy.sol#215)
Strategy._safeSwapExactInput(address,address,uint256) (contracts/contracts/strategies/Strategy.sol#218-224) ignores return value by swapper.swapExactInput(_tokenIn,_tokenOut,_amountIn,1,address(this)) (contracts/contracts/strategies/Strategy.sol#223)
AaveV3._approveToken(uint256) (contracts/contracts/strategies/aave/v3/AaveV3.sol#48-58) ignores return value by AToken(receiptToken).getIncentivesController() (contracts/contracts/strategies/aave/v3/AaveV3.sol#51-57)
AaveV3Incentive._claimRewards(address) (contracts/contracts/strategies/aave/v3/AaveV3Incentive.sol#11-22) ignores return value by AToken(_aToken).getIncentivesController() (contracts/contracts/strategies/aave/v3/AaveV3Incentive.sol#16-21)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

Variable 'AaveV3._approveToken(uint256)._aaveIncentivesController (contracts/contracts/strategies/aave/v3/AaveV3.sol#51)' in AaveV3._approveToken(uint256) (contracts/contracts/strategies/aave/v3/AaveV3.sol#48-58) potentially used before declaration: _rewardTokens = AaveIncentivesController(_aaveIncentivesController).getRewardsList() (contracts/contracts/strategies/aave/v3/AaveV3.sol#52)
Variable 'AaveV3Incentive._claimRewards(address)._aaveIncentivesController (contracts/contracts/strategies/aave/v3/AaveV3Incentive.sol#16)' in AaveV3Incentive._claimRewards(address) (contracts/contracts/strategies/aave/v3/AaveV3Incentive.sol#11-22) potentially used before declaration: IAaveIncentivesController(_aaveIncentivesController).claimAllRewards(assets,address(this)) (contracts/contracts/strategies/aave/v3/AaveV3Incentive.sol#19)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables

Address.isContract(address) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#26-35) uses assembly
        - INLINE ASM (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#33)
Address._verifyCallResult(bool,bytes,string) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#171-188) uses assembly
        - INLINE ASM (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#180-183)
EnumerableSet.values(EnumerableSet.AddressSet) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#273-282) uses assembly
        - INLINE ASM (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#277-279)
EnumerableSet.values(EnumerableSet.UintSet) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#346-355) uses assembly
        - INLINE ASM (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#350-352)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Different versions of Solidity is used:
        - Version used: ['0.8.9', '^0.8.0']
        - 0.8.9 (contracts/contracts/interfaces/aave/IAave.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/swapper/IRoutedSwapper.sol#3)
        - 0.8.9 (contracts/contracts/strategies/Strategy.sol#3)
        - 0.8.9 (contracts/contracts/strategies/aave/v3/AaveV3.sol#3)
        - 0.8.9 (contracts/contracts/strategies/aave/v3/AaveV3Incentive.sol#3)
        - 0.8.9 (vesper-commons/contracts/interfaces/vesper/IStrategy.sol#3)
        - 0.8.9 (vesper-pools/contracts/Errors.sol#2)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/IERC20.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Context.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/Math.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#3)
        - 0.8.9 (vesper-pools/contracts/interfaces/vesper/IGovernable.sol#3)
        - 0.8.9 (vesper-pools/contracts/interfaces/vesper/IPausable.sol#3)
        - 0.8.9 (vesper-pools/contracts/interfaces/vesper/IVesperPool.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

Address.functionCall(address,bytes) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#79-81) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#104-106) is never used and should be removed
Address.functionDelegateCall(address,bytes) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#153-155) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#163-169) is never used and should be removed
Address.functionStaticCall(address,bytes) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#129-131) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#139-145) is never used and should be removed

Context._msgData() (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Context.sol#20-23) is never used and should be removed
EnumerableSet._at(EnumerableSet.Set,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#129-131) is never used and should be removed
EnumerableSet._length(EnumerableSet.Set) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#115-117) is never used and should be removed
EnumerableSet.add(EnumerableSet.Bytes32Set,bytes32) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#157-159) is never used and should be removed
EnumerableSet.add(EnumerableSet.UintSet,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#296-298) is never used and should be removed
EnumerableSet.at(EnumerableSet.AddressSet,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#261-263) is never used and should be removed
EnumerableSet.at(EnumerableSet.Bytes32Set,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#195-197) is never used and should be removed
EnumerableSet.at(EnumerableSet.UintSet,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#334-336) is never used and should be removed
EnumerableSet.contains(EnumerableSet.Bytes32Set,bytes32) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#174-176) is never used and should be removed
EnumerableSet.contains(EnumerableSet.UintSet,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#313-315) is never used and should be removed
EnumerableSet.length(EnumerableSet.AddressSet) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#247-249) is never used and should be removed
EnumerableSet.length(EnumerableSet.Bytes32Set) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#181-183) is never used and should be removed
EnumerableSet.length(EnumerableSet.UintSet) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#320-322) is never used and should be removed
EnumerableSet.remove(EnumerableSet.Bytes32Set,bytes32) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#167-169) is never used and should be removed
EnumerableSet.remove(EnumerableSet.UintSet,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#306-308) is never used and should be removed
EnumerableSet.values(EnumerableSet.Bytes32Set) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#207-209) is never used and should be removed
EnumerableSet.values(EnumerableSet.UintSet) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#346-355) is never used and should be removed
Math.average(uint256,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/Math.sol#27-30) is never used and should be removed
Math.max(uint256,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/Math.sol#12-14) is never used and should be removed
SafeERC20.safeDecreaseAllowance(IERC20,address,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#51-58) is never used and should be removed
SafeERC20.safeIncreaseAllowance(IERC20,address,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#46-49) is never used and should be removed
SafeERC20.safeTransferFrom(IERC20,address,address,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#24-26) is never used and should be removed
Strategy._beforeMigration(address) (contracts/contracts/strategies/Strategy.sol#195) is never used and should be removed
Strategy._rebalance() (contracts/contracts/strategies/Strategy.sol#201-208) is never used and should be removed
Strategy._swapExactInput(address,address,uint256) (contracts/contracts/strategies/Strategy.sol#210-216) is never used and should be removed
Strategy._withdrawHere(uint256) (contracts/contracts/strategies/Strategy.sol#227) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version0.8.9 (contracts/contracts/interfaces/aave/IAave.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/swapper/IRoutedSwapper.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/strategies/Strategy.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/strategies/aave/v3/AaveV3.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/strategies/aave/v3/AaveV3Incentive.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (vesper-commons/contracts/interfaces/vesper/IStrategy.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (vesper-pools/contracts/Errors.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/IERC20.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Context.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/Math.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#3) allows old versions
Pragma version0.8.9 (vesper-pools/contracts/interfaces/vesper/IGovernable.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (vesper-pools/contracts/interfaces/vesper/IPausable.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (vesper-pools/contracts/interfaces/vesper/IVesperPool.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
solc-0.8.9 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
```

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

Function AToken.UNDERLYING_ASSET_ADDRESS() (contracts/contracts/interfaces/aave/IAave.sol#45) is not in mixedCase
Function StakedAave.COOLDOWN_SECONDS() (contracts/contracts/interfaces/aave/IAave.sol#180) is not in mixedCase
Function StakedAave.UNSTAKE_WINDOW() (contracts/contracts/interfaces/aave/IAave.sol#182) is not in mixedCase
Parameter Strategy.addKeeper(address)._keeperAddress (contracts/contracts/strategies/Strategy.sol#64) is not in mixedCase
Parameter Strategy.migrate(address)._newStrategy (contracts/contracts/strategies/Strategy.sol#87) is not in mixedCase
Parameter Strategy.removeKeeper(address)._keeperAddress (contracts/contracts/strategies/Strategy.sol#119) is not in mixedCase
Parameter Strategy.sweepERC20(address)._fromToken (contracts/contracts/strategies/Strategy.sol#127) is not in mixedCase
Parameter Strategy.updateFeeCollector(address)._feeCollector (contracts/contracts/strategies/Strategy.sol#156) is not in mixedCase
Parameter Strategy.updateSwapper(IRoutedSwapper)._swapper (contracts/contracts/strategies/Strategy.sol#167) is not in mixedCase
Parameter Strategy.withdraw(uint256)._amount (contracts/contracts/strategies/Strategy.sol#178) is not in mixedCase
Parameter AaveV3.isAaveV3Token(address)._token (contracts/contracts/strategies/aave/v3/AaveV3.sol#34) is not in mixedCase
Variable AaveV3.NAME (contracts/contracts/strategies/aave/v3/AaveV3.sol#14) is not in mixedCase
Function IStrategy.VERSION() (vesper-commons/contracts/interfaces/vesper/IStrategy.sol#31) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

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

tvl() should be declared external:
        - AaveV3.tvl() (contracts/contracts/strategies/aave/v3/AaveV3.sol#42-45)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

AaveV3VesperXy.sol

```
AaveV3VesperXy._withdrawFromVesperPool(uint256) (contracts/contracts/strategies/aave/v3/AaveV3VesperXy.sol#81-88) performs a multiplication on the result of a division:
        -_shares = (_amount * 1e18) / _pricePerShare (contracts/contracts/strategies/aave/v3/AaveV3VesperXy.sol#84)
        -_amount > ((_shares * _pricePerShare) / 1e18) (contracts/contracts/strategies/aave/v3/AaveV3VesperXy.sol#85)
AaveV3Xy._calculateBorrowPosition(uint256,uint256,uint256,uint256) (contracts/contracts/strategies/aave/v3/AaveV3Xy.sol#112-172) performs a multiplication on the result of a division:
        -_actualCollateralForBorrow = (_hypotheticalCollateral * _collateralFactor * _collateralTokenPrice) / (MAX_BPS * (10 ** IERC20Metadata(address(collateralToken)).decimals())) (contracts/contracts/strategies/aave/v3/AaveV3Xy.sol#146-148)
        -_maxBorrowPossible = (_actualCollateralForBorrow * (10 ** IERC20Metadata(address(borrowToken)).decimals())) / _borrowTokenPrice (contracts/contracts/strategies/aave/v3/AaveV3Xy.sol#150-151)
AaveV3Xy._calculateBorrowPosition(uint256,uint256,uint256,uint256) (contracts/contracts/strategies/aave/v3/AaveV3Xy.sol#112-172) performs a multiplication on the result of a division:
        -_maxBorrowPossible = (_actualCollateralForBorrow * (10 ** IERC20Metadata(address(borrowToken)).decimals())) / _borrowTokenPrice (contracts/contracts/strategies/aave/v3/AaveV3Xy.sol#150-151)
        -_borrowUpperBound = (_maxBorrowPossible * maxBorrowLimit) / MAX_BPS (contracts/contracts/strategies/aave/v3/AaveV3Xy.sol#156)
AaveV3Xy._calculateBorrowPosition(uint256,uint256,uint256,uint256) (contracts/contracts/strategies/aave/v3/AaveV3Xy.sol#112-172) performs a multiplication on the result of a division:
        -_maxBorrowPossible = (_actualCollateralForBorrow * (10 ** IERC20Metadata(address(borrowToken)).decimals())) / _borrowTokenPrice (contracts/contracts/strategies/aave/v3/AaveV3Xy.sol#150-151)
        -_borrowLowerBound = (_maxBorrowPossible * minBorrowLimit) / MAX_BPS (contracts/contracts/strategies/aave/v3/AaveV3Xy.sol#159)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#divide-before-multiply

AaveV3Xy._calculateBorrowPosition(uint256,uint256,uint256,uint256) (contracts/contracts/strategies/aave/v3/AaveV3Xy.sol#112-172) uses a dangerous strict equality:
        - require(bool,string)(_depositAmount == 0 || _withdrawAmount == 0,all-input-gt-zero) (contracts/contracts/strategies/aave/v3/AaveV3Xy.sol#119)
AaveV3Xy._calculateBorrowPosition(uint256,uint256,uint256,uint256) (contracts/contracts/strategies/aave/v3/AaveV3Xy.sol#112-172) uses a dangerous strict equality:
        -_hypotheticalCollateral == 0 (contracts/contracts/strategies/aave/v3/AaveV3Xy.sol#128)
AaveV3Xy._calculateBorrowPosition(uint256,uint256,uint256,uint256) (contracts/contracts/strategies/aave/v3/AaveV3Xy.sol#112-172) uses a dangerous strict equality:
        -_maxBorrowPossible == 0 (contracts/contracts/strategies/aave/v3/AaveV3Xy.sol#152)
AaveV3Xy._swapToBorrowToken(uint256,AaveLendingPool) (contracts/contracts/strategies/aave/v3/AaveV3Xy.sol#299-314) uses a dangerous strict equality:
        - require(bool,string)(_aaveLendingPool.withdraw(address(collateralToken),_amount,address(this)) == _amount,Errors.INCORRECT_WITHDRAW_AMOUNT) (contracts/contracts/strategies/aave/v3/AaveV3Xy.sol#307-310)
AaveV3Xy._withdrawHere(uint256,AaveLendingPool,uint256,uint256) (contracts/contracts/strategies/aave/v3/AaveV3Xy.sol#326-347) uses a dangerous strict equality:
        - require(bool,string)(_aaveLendingPool.withdraw(address(collateralToken),_possibleWithdraw,address(this)) == _possibleWithdraw,Errors.INCORRECT_WITHDRAW_AMOUNT) (contracts/contracts/strategies/aave/v3/AaveV3Xy.sol#343-346)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dangerous-strict-equalities

AaveV3Xy._approveToken(uint256).i (contracts/contracts/strategies/aave/v3/AaveV3Xy.sol#86) is a local variable never initialized
AaveV3Xy._claimRewardsAndConvertTo(address).i (contracts/contracts/strategies/aave/v3/AaveV3Xy.sol#178) is a local variable never initialized
AaveV3Xy._depositToAave(uint256,AaveLendingPool)._reason (contracts/contracts/strategies/aave/v3/AaveV3Xy.sol#191) is a local variable never initialized
AaveV3Incentive._claimRewards(address)._aaveIncentivesController (contracts/contracts/strategies/aave/v3/AaveV3Incentive.sol#16) is a local variable never initialized
AaveV3Xy._approveToken(uint256)._aaveIncentivesController (contracts/contracts/strategies/aave/v3/AaveV3Xy.sol#84) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

Strategy._swapExactInput(address,address,uint256) (contracts/contracts/strategies/Strategy.sol#210-216) ignores return value by swapper.swapExactInput(_tokenIn,_tokenOut,_amountIn,1,address(this)) (contracts/contracts/strategies/Strategy.sol#215)
Strategy._safeSwapExactInput(address,address,uint256) (contracts/contracts/strategies/Strategy.sol#218-224) ignores return value by swapper.swapExactInput(_tokenIn,_tokenOut,_amountIn,1,address(this)) (contracts/contracts/strategies/Strategy.sol#223)
AaveV3Incentive._claimRewards(address) (contracts/contracts/strategies/aave/v3/AaveV3Incentive.sol#11-22) ignores return value by AToken(_aToken).getIncentivesController() (contracts/contracts/strategies/aave/v3/AaveV3Incentive.sol#16-21)
AaveV3Xy._approveToken(uint256) (contracts/contracts/strategies/aave/v3/AaveV3Xy.sol#77-91) ignores return value by AToken(receiptToken).getIncentivesController() (contracts/contracts/strategies/aave/v3/AaveV3Xy.sol#84-90)
AaveV3Xy._swapToBorrowToken(uint256,AaveLendingPool) (contracts/contracts/strategies/aave/v3/AaveV3Xy.sol#299-314) ignores return value by swapper.swapExactOutput(address(collateralToken),borrowToken,_shortOnBorrow,_amountIn,address(this)) (contracts/contracts/strategies/aave/v3/AaveV3Xy.sol#312)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

AaveV3Xy.constructor(address,address,address,address,address,string)._borrowToken (contracts/contracts/strategies/aave/v3/AaveV3Xy.sol#40) lacks a zero-check on :
        - borrowToken = _borrowToken (contracts/contracts/strategies/aave/v3/AaveV3Xy.sol#54)
AaveV3Xy.constructor(address,address,address,address,address,string)._aBorrowToken (contracts/contracts/strategies/aave/v3/AaveV3Xy.sol#50) lacks a zero-check on :
        - aBorrowToken = _aBorrowToken (contracts/contracts/strategies/aave/v3/AaveV3Xy.sol#55)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation

Variable 'AaveV3Incentive._claimRewards(address)._aaveIncentivesController (contracts/contracts/strategies/aave/v3/AaveV3Incentive.sol#16)' in AaveV3Incentive._claimRewards(address) (contracts/contracts/strategies/aave/v3/AaveV3Incentive.sol#11-22) potentially used before declaration: AaveIncentivesController(_aaveIncentivesController).claimAllRewards(assets,address(this)) (contracts/contracts/strategies/aave/v3/AaveV3Incentive.sol#18)
Variable 'AaveV3Xy._approveToken(uint256)._aaveIncentivesController (contracts/contracts/strategies/aave/v3/AaveV3Xy.sol#84)' in AaveV3Xy._approveToken(uint256) (contracts/contracts/strategies/aave/v3/AaveV3Xy.sol#77-91) potentially used before declaration: _rewardTokens = AaveIncentivesController(_aaveIncentivesController).getRewardsList() (contracts/contracts/strategies/aave/v3/AaveV3Xy.sol#85)
Variable 'AaveV3Xy._depositToAave(uint256,AaveLendingPool)._reason (contracts/contracts/strategies/aave/v3/AaveV3Xy.sol#191)' in AaveV3Xy._depositToAave(uint256,AaveLendingPool) (contracts/contracts/strategies/aave/v3/AaveV3Xy.sol#188-200) potentially used before declaration: require(bool,string)(bytes32(bytes(_reason)) == 56,deposit failed) (contracts/contracts/strategies/aave/v3/AaveV3Xy.sol#197)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables

Address.isContract(address) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#26-35) uses assembly
        - INLINE ASM (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#33)
Address._verifyCallResult(bool,bytes,string) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#171-188) uses assembly
        - INLINE ASM (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#180-183)

EnumerableSet.values(EnumerableSet.AddressSet) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#273-282) uses assembly
        - INLINE ASM (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#277-279)
EnumerableSet.values(EnumerableSet.UintSet) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#346-355) uses assembly
        - INLINE ASM (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#350-352)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Different versions of Solidity is used:
        - Version used: ['0.8.9', '^0.8.0']
        - 0.8.9 (contracts/contracts/interfaces/aave/IAave.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/swapper/IRoutedSwapper.sol#3)
        - 0.8.9 (contracts/contracts/strategies/Strategy.sol#3)
        - 0.8.9 (contracts/contracts/strategies/aave/v3/AaveV3Incentive.sol#3)
        - 0.8.9 (contracts/contracts/strategies/aave/v3/AaveV3VesperXy.sol#3)
        - 0.8.9 (contracts/contracts/strategies/aave/v3/AaveV3Xy.sol#3)
        - 0.8.9 (vesper-commons/contracts/interfaces/vesper/IStrategy.sol#3)
        - 0.8.9 (vesper-pools/contracts/Errors.sol#2)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/IERC20.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Context.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/Math.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#3)
        - 0.8.9 (vesper-pools/contracts/interfaces/vesper/IGovernable.sol#3)
        - 0.8.9 (vesper-pools/contracts/interfaces/vesper/IPausable.sol#3)
        - 0.8.9 (vesper-pools/contracts/interfaces/vesper/IPoolRewards.sol#3)
        - 0.8.9 (vesper-pools/contracts/interfaces/vesper/IVesperPool.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

AaveV3Xy._afterBorrowY(uint256) (contracts/contracts/strategies/aave/v3/AaveV3Xy.sol#74) is never used and should be removed
AaveV3Xy._beforeRepayY(uint256) (contracts/contracts/strategies/aave/v3/AaveV3Xy.sol#103) is never used and should be removed
AaveV3Xy._getInvertedBorrowBalance() (contracts/contracts/strategies/aave/v3/AaveV3Xy.sol#203-208) is never used and should be removed
AaveV3Xy._rebalanceBorrow(uint256) (contracts/contracts/strategies/aave/v3/AaveV3Xy.sol#288) is never used and should be removed
Address.functionCall(address,bytes) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#79-81) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#104-106) is never used and should be removed
Address.functionDelegateCall(address,bytes) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#153-155) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#163-169) is never used and should be removed
Address.functionStaticCall(address,bytes) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#129-131) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#139-145) is never used and should be removed
Context._msgData() (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Context.sol#20-23) is never used and should be removed
EnumerableSet._at(EnumerableSet.Set,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#129-131) is never used and should be removed
EnumerableSet._length(EnumerableSet.Set) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#115-117) is never used and should be removed
EnumerableSet.add(EnumerableSet.Bytes32Set,bytes32) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#157-159) is never used and should be removed
EnumerableSet.add(EnumerableSet.UintSet,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#296-298) is never used and should be removed
EnumerableSet.at(EnumerableSet.AddressSet,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#261-263) is never used and should be removed
EnumerableSet.at(EnumerableSet.Bytes32Set,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#195-197) is never used and should be removed
EnumerableSet.at(EnumerableSet.UintSet,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#334-336) is never used and should be removed
EnumerableSet.contains(EnumerableSet.Bytes32Set,bytes32) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#176-176) is never used and should be removed
EnumerableSet.contains(EnumerableSet.UintSet,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#313-315) is never used and should be removed
EnumerableSet.length(EnumerableSet.AddressSet) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#247-249) is never used and should be removed
EnumerableSet.length(EnumerableSet.Bytes32Set) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#181-183) is never used and should be removed
EnumerableSet.length(EnumerableSet.UintSet) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#320-322) is never used and should be removed
EnumerableSet.remove(EnumerableSet.Bytes32Set,bytes32) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#167-169) is never used and should be removed
EnumerableSet.remove(EnumerableSet.UintSet,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#306-308) is never used and should be removed
EnumerableSet.values(EnumerableSet.Bytes32Set) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#207-209) is never used and should be removed
EnumerableSet.values(EnumerableSet.UintSet) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#346-355) is never used and should be removed
Math.average(uint256,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/Math.sol#27-30) is never used and should be removed
Math.max(uint256,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/Math.sol#12-14) is never used and should be removed
SafeERC20.safeDecreaseAllowance(IERC20,address,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#51-58) is never used and should be removed
SafeERC20.safeIncreaseAllowance(IERC20,address,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#46-49) is never used and should be removed
SafeERC20.safeTransferFrom(IERC20,address,address,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#24-26) is never used and should be removed
Strategy._beforeMigration(address) (contracts/contracts/strategies/Strategy.sol#199) is never used and should be removed
Strategy._rebalance() (contracts/contracts/strategies/Strategy.sol#201-208) is never used and should be removed
```

```
Strategy._withdrawWere(uint256) (contracts/contracts/strategies/Strategy.sol#227) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version0.8.9 (contracts/contracts/interfaces/aave/ILAave.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/swapper/IRoutedSwapper.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/strategies/Strategy.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/strategies/aave/v3/AaveV3Incentive.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/strategies/aave/v3/AaveV3VesperXy.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/strategies/aave/v3/AaveV3Xy.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (vesper-commons/contracts/interfaces/vesper/IStrategy.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (vesper-pools/contracts/Errors.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/IERC20.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Context.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/Math.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#3) allows old versions
Pragma version0.8.9 (vesper-pools/contracts/interfaces/vesper/IGovernable.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (vesper-pools/contracts/interfaces/vesper/IPausable.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (vesper-pools/contracts/interfaces/vesper/IPoolRewards.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (vesper-pools/contracts/interfaces/vesper/IVesperPool.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
solc-0.8.9 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

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

Function ATokenV3.UNDERLYING_ASSET_ADDRESS() (contracts/contracts/interfaces/aave/ILAave.sol#45) is not in mixedCase
Function StakedAave.COOLDOWN_SECONDS() (contracts/contracts/interfaces/aave/ILAave.sol#130) is not in mixedCase
Function StakedAave.UNSTAKE_WINDOW() (contracts/contracts/interfaces/aave/ILAave.sol#82) is not in mixedCase
Parameter Strategy.addKeeper(address)._keeperAddress (contracts/contracts/strategies/Strategy.sol#64) is not in mixedCase
Parameter Strategy.migrate(address)._newStrategy (contracts/contracts/strategies/Strategy.sol#87) is not in mixedCase
Parameter Strategy.removeKeeper(address)._keeperAddress (contracts/contracts/strategies/Strategy.sol#119) is not in mixedCase
Parameter Strategy.sweepERC20(address)._fromToken (contracts/contracts/strategies/Strategy.sol#127) is not in mixedCase
Parameter Strategy.updateFeeCollector(address)._feeCollector (contracts/contracts/strategies/Strategy.sol#156) is not in mixedCase
Parameter Strategy.updateSwapper(IRoutedSwapper)._swapper (contracts/contracts/strategies/Strategy.sol#167) is not in mixedCase
Parameter Strategy.withdraw(uint256)._amount (contracts/contracts/strategies/Strategy.sol#178) is not in mixedCase
Parameter AaveV3Xy.isReservedToken(address)._token (contracts/contracts/strategies/aave/v3/AaveV3Xy.sol#59) is not in mixedCase
Parameter AaveV3Xy.updateBorrowLimit(uint256,uint256)._minBorrowLimit (contracts/contracts/strategies/aave/v3/AaveV3Xy.sol#358) is not in mixedCase
Parameter AaveV3Xy.updateBorrowLimit(uint256,uint256)._maxBorrowLimit (contracts/contracts/strategies/aave/v3/AaveV3Xy.sol#358) is not in mixedCase
Variable AaveV3Xy.NAME (contracts/contracts/strategies/aave/v3/AaveV3Xy.sol#18) is not in mixedCase
Function IStrategy.VERSION() (vesper-commons/contracts/interfaces/vesper/IStrategy.sol#31) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

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

Variable AaveV3VesperXy.constructor(address,address,address,address,address,address,address,address,string)._borrowToken (contracts/contracts/strategies/aave/v3/AaveV3VesperXy.sol#21) is too similar to AaveV3Xy.aBorrowToken (contracts/contracts/strategies/aave/v3/AaveV3Xy.sol#28)
Variable AaveV3Xy.constructor(address,address,address,address,address,address,string)._borrowToken (contracts/contracts/strategies/aave/v3/AaveV3Xy.sol#40) is too similar to AaveV3Xy.aBorrowToken (contracts/contracts/strategies/aave/v3/AaveV3Xy.sol#28)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-are-too-similar
```

Convex2PlainPool.sol

```
CurvePoolBase._withdrawWere(uint256) (contracts/contracts/strategies/curve/CurvePoolBase.sol#306-322) uses a dangerous strict equality:
        - _lpToBurn == 0 (contracts/contracts/strategies/curve/CurvePoolBase.sol#314)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dangerous-strict-equalities

Reentrancy in Convex2PlainPool.setRewardTokens(address[]) (contracts/contracts/strategies/convex/2Pool/Convex2PlainPool.sol#60-68):
        External calls:
        - _claimRewardsAndConvertTo(address(collateralToken)) (contracts/contracts/strategies/convex/2Pool/Convex2PlainPool.sol#64)
                - require(bool,string)(cvxCrvRewards.getReward(address(this),true),reward-claim-failed) (contracts/contracts/strategies/convex/2Pool/Convex2PlainPool.sol#39)
                - swapper.swapExactInput(_rewardToken,tokenOut_,_amountIn,1,address(this)) (contracts/contracts/strategies/curve/CurvePoolBase.sol#190-193)
        State variables written after the call(s):
        - rewardTokens = _getRewardTokens() (contracts/contracts/strategies/convex/2Pool/Convex2PlainPool.sol#65)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-1

CurvePoolBase._claimRewardsAndConvertTo(address).i (contracts/contracts/strategies/curve/CurvePoolBase.sol#186) is a local variable never initialized
ConvexBase._getRewardTokens()._extraRewardCount (contracts/contracts/strategies/convex/ConvexBase.sol#63) is a local variable never initialized
CurvePoolBase.setRewardTokens(address[]).i (contracts/contracts/strategies/curve/CurvePoolBase.sol#390) is a local variable never initialized
CurvePoolBase._approveToken(uint256).i (contracts/contracts/strategies/curve/CurvePoolBase.sol#148) is a local variable never initialized
ConvexBase._getRewardTokens().i_scope_0 (contracts/contracts/strategies/convex/ConvexBase.sol#80) is a local variable never initialized
ConvexBase._getRewardTokens().i (contracts/contracts/strategies/convex/ConvexBase.sol#66) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

Strategy._swapExactInput(address,address,uint256) (contracts/contracts/strategies/Strategy.sol#210-216) ignores return value by swapper.swapExactInput(_tokenIn,_tokenOut,_amountIn,1,address(this)) (contracts/contracts/strategies/Strategy.sol#215)
Strategy._safeSwapExactInput(address,address,uint256) (contracts/contracts/strategies/Strategy.sol#218-224) ignores return value by swapper.swapExactInput(_tokenIn,_tokenOut,_amountIn,1,address(this)) (contracts/contracts/strategies/Strategy.sol#223)
CurvePoolBase._claimRewardsAndConvertTo(address) (contracts/contracts/strategies/curve/CurvePoolBase.sol#183-196) ignores return value by swapper.swapExactInput(_rewardToken,tokenOut_,_amountIn,1,address(this)) (contracts/contracts/strategies/curve/CurvePoolBase.sol#190-193)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

Address.isContract(address) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#26-35) uses assembly
        - INLINE ASM (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#33)
Address._verifyCallResult(bool,bytes,string) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#171-188) uses assembly
        - INLINE ASM (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#180-183)
EnumerableSet.values(EnumerableSet.AddressSet) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#273-282) uses assembly
        - INLINE ASM (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#277-279)
EnumerableSet.values(EnumerableSet.UintSet) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#346-355) uses assembly
        - INLINE ASM (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#350-352)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Different versions of Solidity is used:
        - Version used: ['0.8.9', '^0.8.0']
        - 0.8.9 (contracts/contracts/interfaces/convex/ICvxCrvRewards.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/convex/IConvexToken.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/curve/IAddressProvider.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/curve/IGaugeController.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/curve/IGaugeMinter.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#4)
        - 0.8.9 (contracts/contracts/interfaces/curve/IRegistry.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/curve/IStableSwap.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/curve/ITokenMinter.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/one-oracle/IMasterOracle.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/swapper/IRoutedSwapper.sol#3)
        - 0.8.9 (contracts/contracts/strategies/Strategy.sol#3)
        - 0.8.9 (contracts/contracts/strategies/convex/2Pool/Convex2PlainPool.sol#3)
        - 0.8.9 (contracts/contracts/strategies/convex/ConvexBase.sol#3)
        - 0.8.9 (contracts/contracts/strategies/curve/2Pool/Curve2PlainPool.sol#3)
        - 0.8.9 (contracts/contracts/strategies/curve/CurvePoolBase.sol#3)
        - 0.8.9 (vesper-commons/contracts/interfaces/vesper/IStrategy.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/IERC20.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Context.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/Math.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#3)
        - 0.8.9 (vesper-pools/contracts/interfaces/vesper/IGovernable.sol#3)
        - 0.8.9 (vesper-pools/contracts/interfaces/vesper/IPausable.sol#3)
        - 0.8.9 (vesper-pools/contracts/interfaces/vesper/IVesperPool.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

Address.functionCall(address,bytes) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#79-81) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#104-106) is never used and should be removed
Address.functionDelegateCall(address,bytes) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#153-155) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#163-169) is never used and should be removed
Address.functionStaticCall(address,bytes) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#129-131) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#139-145) is never used and should be removed
Context._msgData() (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Context.sol#20-23) is never used and should be removed
ConvexBase._calculateCVXRewards(uint256) (contracts/contracts/strategies/convex/ConvexBase.sol#34-59) is never used and should be removed
CurvePoolBase._claimRewards() (contracts/contracts/strategies/curve/CurvePoolBase.sol#169-178) is never used and should be removed
CurvePoolBase._depositToCurve(uint256) (contracts/contracts/strategies/curve/CurvePoolBase.sol#203) is never used and should be removed
CurvePoolBase._stakeAllLp() (contracts/contracts/strategies/curve/CurvePoolBase.sol#281-286) is never used and should be removed
CurvePoolBase._unstakeAllLp() (contracts/contracts/strategies/curve/CurvePoolBase.sol#288-290) is never used and should be removed
CurvePoolBase._unstakeLp(uint256) (contracts/contracts/strategies/curve/CurvePoolBase.sol#292-296) is never used and should be removed
EnumerableSet._at(EnumerableSet.Set,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#129-131) is never used and should be removed
EnumerableSet._length(EnumerableSet.Set) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#115-117) is never used and should be removed
EnumerableSet.add(EnumerableSet.Bytes32Set,bytes32) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#157-159) is never used and should be removed
EnumerableSet.add(EnumerableSet.UintSet,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#296-298) is never used and should be removed
EnumerableSet.at(EnumerableSet.AddressSet,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#261-263) is never used and should be removed
EnumerableSet.at(EnumerableSet.Bytes32Set,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#195-197) is never used and should be removed
EnumerableSet.at(EnumerableSet.UintSet,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#334-336) is never used and should be removed
EnumerableSet.contains(EnumerableSet.Bytes32Set,bytes32) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#174-176) is never used and should be removed
EnumerableSet.contains(EnumerableSet.UintSet,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#313-315) is never used and should be removed
EnumerableSet.length(EnumerableSet.AddressSet) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#247-249) is never used and should be removed
EnumerableSet.length(EnumerableSet.Bytes32Set) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#181-183) is never used and should be removed
EnumerableSet.length(EnumerableSet.UintSet) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#320-322) is never used and should be removed
EnumerableSet.remove(EnumerableSet.Bytes32Set,bytes32) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#167-169) is never used and should be removed
EnumerableSet.remove(EnumerableSet.UintSet,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#306-308) is never used and should be removed
EnumerableSet.values(EnumerableSet.Bytes32Set) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#207-209) is never used and should be removed
EnumerableSet.values(EnumerableSet.UintSet) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#346-355) is never used and should be removed
Math.average(uint256,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/Math.sol#27-30) is never used and should be removed
Math.max(uint256,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/Math.sol#12-14) is never used and should be removed
SafeCast.toInt16(int256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#175-178) is never used and should be removed
SafeCast.toInt256(uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#205-208) is never used and should be removed
SafeCast.toInt32(int256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#157-160) is never used and should be removed
SafeCast.toInt64(int256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#139-142) is never used and should be removed
SafeCast.toInt8(int256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#193-196) is never used and should be removed
SafeCast.toUint128(uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#31-34) is never used and should be removed
SafeCast.toUint16(uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#76-79) is never used and should be removed
```

```
SafeCast.toUint256(int256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#103-106) is never used and should be removed
SafeCast.toUint32(uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#61-64) is never used and should be removed
SafeCast.toUint64(uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#46-49) is never used and should be removed
SafeCast.toUint8(uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#91-94) is never used and should be removed
SafeERC20.safeDecreaseAllowance(IERC20,address,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#51-58) is never used and should be removed
SafeERC20.safeIncreaseAllowance(IERC20,address,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#46-49) is never used and should be removed
SafeERC20.safeTransferFrom(IERC20,address,address,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#24-26) is never used and should be removed
Strategy._beforeMigration(address) (contracts/contracts/strategies/Strategy.sol#199) is never used and should be removed
Strategy._rebalance() (contracts/contracts/strategies/Strategy.sol#201-208) is never used and should be removed
Strategy._safeSwapExactInput(address,address,uint256) (contracts/contracts/strategies/Strategy.sol#218-224) is never used and should be removed
Strategy._swapExactInput(address,address,uint256) (contracts/contracts/strategies/Strategy.sol#210-216) is never used and should be removed
Strategy._withdrawHere(uint256) (contracts/contracts/strategies/Strategy.sol#227) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version0.8.9 (contracts/contracts/interfaces/convex/IConvex.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/convex/IConvexToken.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/curve/IAddressProvider.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#4) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/curve/IRegistry.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/curve/IStableSwap.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/curve/ITokenMinter.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/one-oracle/IMasterOracle.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/swapper/IRoutedSwapper.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/strategies/Strategy.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/strategies/convex/2Pool/Convex2PlainPool.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/strategies/convex/ConvexBase.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/strategies/curve/2Pool/CurvePlainPool.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/strategies/curve/CurvePoolBase.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (vesper-commons/contracts/interfaces/vesper/IStrategy.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/IERC20.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Context.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/Math.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#3) allows old versions
Pragma version0.8.9 (vesper-pools/contracts/interfaces/vesper/IGovernable.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (vesper-pools/contracts/interfaces/vesper/IPausable.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (vesper-pools/contracts/interfaces/vesper/IVesperPool.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
solc-0.8.9 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

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

Function IAddressProvider.get_registry() (contracts/contracts/interfaces/curve/IAddressProvider.sol#6) is not in mixedCase
Function IAddressProvider.get_address(uint256) (contracts/contracts/interfaces/curve/IAddressProvider.sol#8) is not in mixedCase
Function ILiquidityGaugeV2.claim_rewards(address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#15) is not in mixedCase
Function ILiquidityGaugeV2.claim_rewards() (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#17) is not in mixedCase
Function ILiquidityGaugeV2.claimable_tokens(address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#19) is not in mixedCase
Function ILiquidityGaugeV2.claimable_reward(address,address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#21) is not in mixedCase
Function ILiquidityGaugeV2.integrate_fraction(address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#23) is not in mixedCase
Function ILiquidityGaugeV2.user_checkpoint(address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#25) is not in mixedCase
Function ILiquidityGaugeV2.reward_integral(address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#27) is not in mixedCase
Function ILiquidityGaugeV2.reward_integral_for(address,address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#29) is not in mixedCase
Function ILiquidityGaugeV2.lp_token() (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#31) is not in mixedCase
Function ILiquidityGaugeV2.reward_count() (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#33) is not in mixedCase
Function ILiquidityGaugeV2.reward_tokens(uint256) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#35) is not in mixedCase
Function ILiquidityGaugeV3.claimable_reward(address,address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#39) is not in mixedCase
Function ILiquidityGaugeV3.claimable_reward_write(address,address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#41) is not in mixedCase
Function IMetapoolFactory.get_underlying_coins(address) (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#7) is not in mixedCase
Function IMetapoolFactory.get_underlying_decimals(address) (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#9) is not in mixedCase
Function IMetapoolFactory.get_coins(address) (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#11) is not in mixedCase
Function IMetapoolFactory.get_n_coins(address) (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#13) is not in mixedCase
Function IMetapoolFactory.get_meta_n_coins(address) (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#15) is not in mixedCase
Function IMetapoolFactory.get_decimals(address) (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#17) is not in mixedCase
Function IMetapoolFactory.get_gauge(address) (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#19) is not in mixedCase
Function IMetapoolFactory.is_meta(address) (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#21) is not in mixedCase
Function IRegistry.get_pool_from_lp_token() (contracts/contracts/interfaces/curve/IRegistry.sol#6) is not in mixedCase
Function IRegistry.get_lp_token(address) (contracts/contracts/interfaces/curve/IRegistry.sol#8) is not in mixedCase
Function IRegistry.get_n_coins(address) (contracts/contracts/interfaces/curve/IRegistry.sol#10) is not in mixedCase
Function IRegistry.get_underlying_coins(address) (contracts/contracts/interfaces/curve/IRegistry.sol#12) is not in mixedCase
Function IRegistry.get_underlying_decimals(address) (contracts/contracts/interfaces/curve/IRegistry.sol#14) is not in mixedCase
Function IRegistry.get_gauges(address) (contracts/contracts/interfaces/curve/IRegistry.sol#16) is not in mixedCase
Function IStableSwap.lp_token() (contracts/contracts/interfaces/curve/IStableSwap.sol#11) is not in mixedCase
Function IStableSwap.A() (contracts/contracts/interfaces/curve/IStableSwap.sol#13) is not in mixedCase
Function IStableSwap.A_precise() (contracts/contracts/interfaces/curve/IStableSwap.sol#15) is not in mixedCase
Function IStableSwap.get_virtual_price() (contracts/contracts/interfaces/curve/IStableSwap.sol#18) is not in mixedCase
Function IStableSwap.get_dy(int128,int128,uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#21-25) is not in mixedCase
Function IStableSwap.get_dy_underlying(int128,int128,uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#27-31) is not in mixedCase
Parameter IStableSwap.exchange(int128,int128,uint256,uint256).min_dy (contracts/contracts/interfaces/curve/IStableSwap.sol#37) is not in mixedCase
Function IStableSwap.calc_withdraw_one_coin(uint256,int128) (contracts/contracts/interfaces/curve/IStableSwap.sol#40) is not in mixedCase
Parameter IStableSwap.calc_withdraw_one_coin(uint256,int128)._token_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#40) is not in mixedCase
Function IStableSwap.remove_liquidity_one_coin(uint256,int128,uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#42-46) is not in mixedCase
Parameter IStableSwap.remove_liquidity_one_coin(uint256,int128,uint256)._token_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#43) is not in mixedCase
Parameter IStableSwap.remove_liquidity_one_coin(uint256,int128,uint256)._min_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#45) is not in mixedCase
Function IStableSwapUnderlying.underlying_coins(uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#54) is not in mixedCase
Function IStableSwapUnderlying.exchange_underlying(int128,int128,uint256,uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#56-61) is not in mixedCase
Parameter IStableSwapUnderlying.exchange_underlying(int128,int128,uint256,uint256).min_dy (contracts/contracts/interfaces/curve/IStableSwap.sol#60) is not in mixedCase
Function IStableSwapUnderlying.remove_liquidity_one_coin(uint256,int128,uint256,bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#63-68) is not in mixedCase
Parameter IStableSwapUnderlying.remove_liquidity_one_coin(uint256,int128,uint256,bool)._token_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#64) is not in mixedCase
Parameter IStableSwapUnderlying.remove_liquidity_one_coin(uint256,int128,uint256,bool)._min_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#66) is not in mixedCase
Parameter IStableSwapUnderlying.remove_liquidity_one_coin(uint256,int128,uint256,bool)._use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#67) is not in mixedCase
Function IStableSwap2x.calc_token_amount(uint256[2],bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#72) is not in mixedCase
Parameter IStableSwap2x.calc_token_amount(uint256[2],bool)._is_deposit (contracts/contracts/interfaces/curve/IStableSwap.sol#72) is not in mixedCase
Function IStableSwap2x.add_liquidity(uint256[2],uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#74) is not in mixedCase
Parameter IStableSwap2x.add_liquidity(uint256[2],uint256)._min_mint_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#74) is not in mixedCase
Function IStableSwap2x.remove_liquidity(uint256,uint256[2]) (contracts/contracts/interfaces/curve/IStableSwap.sol#76) is not in mixedCase
Parameter IStableSwap2x.remove_liquidity(uint256,uint256[2])._min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#76) is not in mixedCase
Function IStableSwap2x.remove_liquidity_imbalance(uint256[2],uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#78-80) is not in mixedCase
Parameter IStableSwap2x.remove_liquidity_imbalance(uint256[2],uint256)._max_burn_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#78) is not in mixedCase
Function IStableSwap3x.calc_token_amount(uint256[3],bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#84) is not in mixedCase
Parameter IStableSwap3x.calc_token_amount(uint256[3],bool)._is_deposit (contracts/contracts/interfaces/curve/IStableSwap.sol#84) is not in mixedCase
Function IStableSwap3x.add_liquidity(uint256[3],uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#86) is not in mixedCase
Parameter IStableSwap3x.add_liquidity(uint256[3],uint256)._min_mint_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#86) is not in mixedCase
Function IStableSwap3x.remove_liquidity(uint256,uint256[3]) (contracts/contracts/interfaces/curve/IStableSwap.sol#88) is not in mixedCase
Parameter IStableSwap3x.remove_liquidity(uint256,uint256[3])._min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#88) is not in mixedCase
Function IStableSwap3x.remove_liquidity_imbalance(uint256[3],uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#90-92) is not in mixedCase
Parameter IStableSwap3x.remove_liquidity_imbalance(uint256[3],uint256)._max_burn_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#90) is not in mixedCase
Function IStableSwap4x.calc_token_amount(uint256[4],bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#96) is not in mixedCase
Parameter IStableSwap4x.calc_token_amount(uint256[4],bool)._is_deposit (contracts/contracts/interfaces/curve/IStableSwap.sol#96) is not in mixedCase
Function IStableSwap4x.add_liquidity(uint256[4],uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#98) is not in mixedCase
Parameter IStableSwap4x.add_liquidity(uint256[4],uint256)._min_mint_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#98) is not in mixedCase
Function IStableSwap4x.remove_liquidity(uint256,uint256[4]) (contracts/contracts/interfaces/curve/IStableSwap.sol#100) is not in mixedCase
Parameter IStableSwap4x.remove_liquidity(uint256,uint256[4])._min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#100) is not in mixedCase
Function IStableSwap4x.remove_liquidity_imbalance(uint256[4],uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#102-104) is not in mixedCase
Parameter IStableSwap4x.remove_liquidity_imbalance(uint256[4],uint256)._max_burn_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#102) is not in mixedCase

Function IStableSwap2xUnderlying.add_liquidity(uint256[2],uint256,bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#108-112) is not in mixedCase
Parameter IStableSwap2xUnderlying.add_liquidity(uint256[2],uint256,bool).min_mint_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#110) is not in mixedCase
Parameter IStableSwap2xUnderlying.add_liquidity(uint256[2],uint256,bool).use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#111) is not in mixedCase
Function IStableSwap2xUnderlying.remove_liquidity_imbalance(uint256[2],uint256,bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#114-119) is not in mixedCase
Parameter IStableSwap2xUnderlying.remove_liquidity_imbalance(uint256[2],uint256,bool).max_burn_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#116) is not in mixedCase
Parameter IStableSwap2xUnderlying.remove_liquidity_imbalance(uint256[2],uint256,bool).use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#117) is not in mixedCase
Function IStableSwap2xUnderlying.remove_liquidity(uint256,uint256[2],bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#120-124) is not in mixedCase
Parameter IStableSwap2xUnderlying.remove_liquidity(uint256,uint256[2],bool).min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#122) is not in mixedCase
Parameter IStableSwap2xUnderlying.remove_liquidity(uint256,uint256[2],bool).use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#123) is not in mixedCase
Function IStableSwap3xUnderlying.add_liquidity(uint256[3],uint256,bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#128-132) is not in mixedCase
Parameter IStableSwap3xUnderlying.add_liquidity(uint256[3],uint256,bool).min_mint_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#130) is not in mixedCase
Parameter IStableSwap3xUnderlying.add_liquidity(uint256[3],uint256,bool).use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#131) is not in mixedCase
Function IStableSwap3xUnderlying.remove_liquidity_imbalance(uint256[3],uint256,bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#134-138) is not in mixedCase
Parameter IStableSwap3xUnderlying.remove_liquidity_imbalance(uint256[3],uint256,bool).max_burn_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#136) is not in mixedCase
Parameter IStableSwap3xUnderlying.remove_liquidity_imbalance(uint256[3],uint256,bool).use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#137) is not in mixedCase
Function IStableSwap3xUnderlying.remove_liquidity(uint256,uint256[3],bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#140-144) is not in mixedCase
Parameter IStableSwap3xUnderlying.remove_liquidity(uint256,uint256[3],bool).min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#142) is not in mixedCase
Parameter IStableSwap3xUnderlying.remove_liquidity(uint256,uint256[3],bool).use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#143) is not in mixedCase
Function IStableSwap4xUnderlying.add_liquidity(uint256[4],uint256,bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#148-152) is not in mixedCase
Parameter IStableSwap4xUnderlying.add_liquidity(uint256[4],uint256,bool).min_mint_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#150) is not in mixedCase
Parameter IStableSwap4xUnderlying.add_liquidity(uint256[4],uint256,bool).use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#151) is not in mixedCase
Function IStableSwap4xUnderlying.remove_liquidity_imbalance(uint256[4],uint256,bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#154-158) is not in mixedCase
Parameter IStableSwap4xUnderlying.remove_liquidity_imbalance(uint256[4],uint256,bool).max_burn_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#156) is not in mixedCase
Parameter IStableSwap4xUnderlying.remove_liquidity_imbalance(uint256[4],uint256,bool).use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#157) is not in mixedCase
Function IStableSwap4xUnderlying.remove_liquidity(uint256,uint256[4],bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#160-164) is not in mixedCase
Parameter IStableSwap4xUnderlying.remove_liquidity(uint256,uint256[4],bool).min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#162) is not in mixedCase
Parameter IStableSwap4xUnderlying.remove_liquidity(uint256,uint256[4],bool).use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#163) is not in mixedCase
Parameter ITokenMinter.mint(address).gauge_addr (contracts/contracts/interfaces/curve/ITokenMinter.sol#9) is not in mixedCase
Parameter Strategy.addKeeper(address)._keeperAddress (contracts/contracts/strategies/Strategy.sol#64) is not in mixedCase
Parameter Strategy.migrate(address)._newStrategy (contracts/contracts/strategies/Strategy.sol#87) is not in mixedCase
Parameter Strategy.removeKeeper(address)._keeperAddress (contracts/contracts/strategies/Strategy.sol#119) is not in mixedCase
Parameter Strategy.sweepERC20(address)._fromToken (contracts/contracts/strategies/Strategy.sol#127) is not in mixedCase
Parameter Strategy.updateFeeCollector(address)._feeCollector (contracts/contracts/strategies/Strategy.sol#156) is not in mixedCase
Parameter Strategy.updateSwapper(IRoutedSwapper)._swapper (contracts/contracts/strategies/Strategy.sol#167) is not in mixedCase
Parameter Strategy.withdraw(uint256)._amount (contracts/contracts/strategies/Strategy.sol#178) is not in mixedCase
Variable CurvePoolBase.CRV (contracts/contracts/strategies/curve/CurvePoolBase.sol#37) is not in mixedCase
Variable CurvePoolBase.WETH (contracts/contracts/strategies/curve/CurvePoolBase.sol#39) is not in mixedCase
Function IStrategy.VERSION() (vesper-commons/contracts/interfaces/vesper/IStrategy.sol#31) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

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

```
Variable IStableSwap.remove_liquidity_one_coin(uint256,int128,uint256)._min_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#45) is too similar to IStableSwap2xUnderlying.remove_liquidity(uint256,uint256[2],bool).min_amounts
(contracts/contracts/interfaces/curve/IStableSwap.sol#122)
Variable IStableSwapUnderlying.remove_liquidity_one_coin(uint256,int128,uint256,bool)._min_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#66) is too similar to IStableSwap2xUnderlying.remove_liquidity(uint256,uint256[2],boo
l).min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#122)
Variable IStableSwap.remove_liquidity_one_coin(uint256,int128,uint256)._min_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#45) is too similar to IStableSwap3xUnderlying.remove_liquidity(uint256,uint256[3],bool).min_amounts
(contracts/contracts/interfaces/curve/IStableSwap.sol#142)
Variable IStableSwapUnderlying.remove_liquidity_one_coin(uint256,int128,uint256,bool)._min_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#66) is too similar to IStableSwap3xUnderlying.remove_liquidity(uint256,uint256[3],boo
l).min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#142)
Variable IStableSwap.remove_liquidity_one_coin(uint256,int128,uint256)._min_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#45) is too similar to IStableSwap4xUnderlying.remove_liquidity(uint256,uint256[4],bool).min_amounts
(contracts/contracts/interfaces/curve/IStableSwap.sol#162)
Variable IStableSwapUnderlying.remove_liquidity_one_coin(uint256,int128,uint256,bool)._min_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#66) is too similar to IStableSwap4xUnderlying.remove_liquidity(uint256,uint256[4],boo
l).min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#162)
Variable IStrategy.migrate(address)._newStrategy (vesper-commons/contracts/interfaces/vesper/IStrategy.sol#24) is too similar to CurvePoolBase._beforeMigration(address).newStrategy_ (contracts/contracts/strategies/curve/CurvePoolBase.sol
#155)
Variable Strategy._beforeMigration(address)._newStrategy (contracts/contracts/strategies/Strategy.sol#199) is too similar to CurvePoolBase._beforeMigration(address).newStrategy_ (contracts/contracts/strategies/curve/CurvePoolBase.sol#155
)
Variable ConvexBase._getRewardTokens()._rewardToken (contracts/contracts/strategies/convex/ConvexBase.sol#67) is too similar to CurvePoolBase.rewardTokens (contracts/contracts/strategies/curve/CurvePoolBase.sol#45)
Variable CurvePoolBase._claimRewardsAndConvertTo(address)._rewardToken (contracts/contracts/strategies/curve/CurvePoolBase.sol#187) is too similar to CurvePoolBase.rewardTokens (contracts/contracts/strategies/curve/CurvePoolBase.sol#45)
Variable Strategy.migrate(address)._newStrategy (contracts/contracts/strategies/Strategy.sol#87) is too similar to CurvePoolBase._beforeMigration(address).newStrategy_ (contracts/contracts/strategies/curve/CurvePoolBase.sol#155)
Variable ConvexBase._getRewardTokens()._rewardTokens (contracts/contracts/strategies/convex/ConvexBase.sol#62) is too similar to CurvePoolBase.setRewardTokens(address[])._rewardTokens_ (contracts/contracts/strategies/curve/CurvePoolBase.s
ol#326)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-are-too-similar

Convex2PlainPool.slitherConstructorConstantVariables() (contracts/contracts/strategies/convex/2Pool/Convex2PlainPool.sol#9-73) uses literals with too many digits:
        - ADDRESS_PROVIDER = IAddressProvider(0x0000000022D53366457F9d5E68Ec105046FC4383) (contracts/contracts/strategies/curve/CurvePoolBase.sol#28)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits
```

Convex3PlainPool.sol

```
CurvePoolBase._withdrawHere(uint256) (contracts/contracts/strategies/curve/CurvePoolBase.sol#306-322) uses a dangerous strict equality:
        - _lpToBurn == 0 (contracts/contracts/strategies/curve/CurvePoolBase.sol#314)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dangerous-strict-equalities

Reentrancy in Convex3PlainPool.setRewardTokens(address[]) (contracts/contracts/strategies/convex/3pool/Convex3PlainPool.sol#62-70):
        External calls:
        - _claimRewardsAndConvertTo(address(collateralToken)) (contracts/contracts/strategies/convex/3pool/Convex3PlainPool.sol#66)
                - require(bool,string)(cvxCrvRewards.getReward(address(this),true),reward-claim-failed) (contracts/contracts/strategies/convex/3pool/Convex3PlainPool.sol#41)
                - swapper.swapExactInput(_rewardToken,tokenOut_,_amountIn,1,address(this)) (contracts/contracts/strategies/curve/CurvePoolBase.sol#190-193)
        State variables written after the call(s):
        - rewardTokens = _getRewardTokens() (contracts/contracts/strategies/convex/3pool/Convex3PlainPool.sol#67)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-1

CurvePoolBase._claimRewardsAndConvertTo(address).i (contracts/contracts/strategies/curve/CurvePoolBase.sol#186) is a local variable never initialized
CurvePoolBase.setRewardTokens(address[]).i (contracts/contracts/strategies/curve/CurvePoolBase.sol#330) is a local variable never initialized
CurvePoolBase._approveToken(uint256).i (contracts/contracts/strategies/curve/CurvePoolBase.sol#148) is a local variable never initialized
ConvexBase._getRewardTokens()._extraRewardCount (contracts/contracts/strategies/convex/ConvexBase.sol#63) is a local variable never initialized
ConvexBase._getRewardTokens().i (contracts/contracts/strategies/convex/ConvexBase.sol#66) is a local variable never initialized
ConvexBase._getRewardTokens().i_scope_0 (contracts/contracts/strategies/convex/ConvexBase.sol#80) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

Strategy._swapExactInput(address,address,uint256) (contracts/contracts/strategies/Strategy.sol#210-216) ignores return value by swapper.swapExactInput(_tokenIn,_tokenOut,_amountIn,1,address(this)) (contracts/contracts/strategies/Strategy
.sol#215)
Strategy._safeSwapExactInput(address,address,uint256) (contracts/contracts/strategies/Strategy.sol#218-224) ignores return value by swapper.swapExactInput(_tokenIn,_tokenOut,_amountIn,1,address(this)) (contracts/contracts/strategies/Stra
tegy.sol#223)
CurvePoolBase._claimRewardsAndConvertTo(address) (contracts/contracts/strategies/curve/CurvePoolBase.sol#183-196) ignores return value by swapper.swapExactInput(_rewardToken,tokenOut_,_amountIn,1,address(this)) (contracts/contracts/strat
egies/curve/CurvePoolBase.sol#190-193)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

Address.isContract(address) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#26-35) uses assembly
        - INLINE ASM (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#33)
Address._verifyCallResult(bool,bytes,string) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#171-188) uses assembly
        - INLINE ASM (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#180-183)
EnumerableSet.values(EnumerableSet.AddressSet) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#273-282) uses assembly
        - INLINE ASM (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#277-279)
EnumerableSet.values(EnumerableSet.UintSet) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#346-355) uses assembly
        - INLINE ASM (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#350-352)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Different versions of Solidity is used:
        - Version used: ['0.8.9', '^0.8.0']
        - 0.8.9 (contracts/contracts/interfaces/convex/IConvex.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/convex/IConvexToken.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/curve/IAddressProvider.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#4)
        - 0.8.9 (contracts/contracts/interfaces/curve/IRegistry.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/curve/IStableSwap.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/curve/ITokenMinter.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/one-oracle/IMasterOracle.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/swapper/IRoutedSwapper.sol#3)
        - 0.8.9 (contracts/contracts/strategies/Strategy.sol#3)
        - 0.8.9 (contracts/contracts/strategies/convex/3pool/Convex3PlainPool.sol#3)
        - 0.8.9 (contracts/contracts/strategies/convex/ConvexBase.sol#3)
        - 0.8.9 (contracts/contracts/strategies/curve/3Pool/Curve3PlainPool.sol#3)
        - 0.8.9 (contracts/contracts/strategies/curve/CurvePoolBase.sol#3)
        - 0.8.9 (vesper-commons/contracts/interfaces/vesper/IStrategy.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/IERC20.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Context.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/Math.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#3)
        - 0.8.9 (vesper-pools/contracts/interfaces/vesper/IGovernable.sol#3)
        - 0.8.9 (vesper-pools/contracts/interfaces/vesper/IPausable.sol#3)
        - 0.8.9 (vesper-pools/contracts/interfaces/vesper/IVesperPool.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

Address.functionCall(address,bytes) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#79-81) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#104-106) is never used and should be removed
Address.functionDelegateCall(address,bytes) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#153-155) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#163-169) is never used and should be removed
Address.functionStaticCall(address,bytes) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#129-131) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#139-145) is never used and should be removed
Context._msgData() (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Context.sol#20-23) is never used and should be removed
ConvexBase._calculateCVXRewards(uint256) (contracts/contracts/strategies/convex/ConvexBase.sol#34-59) is never used and should be removed
CurvePoolBase._claimRewards() (contracts/contracts/strategies/curve/CurvePoolBase.sol#169-178) is never used and should be removed
CurvePoolBase._depositToCurve(uint256) (contracts/contracts/strategies/curve/CurvePoolBase.sol#203) is never used and should be removed
CurvePoolBase._stakeAllLp() (contracts/contracts/strategies/curve/CurvePoolBase.sol#281-286) is never used and should be removed
CurvePoolBase._unstakeAllLp() (contracts/contracts/strategies/curve/CurvePoolBase.sol#288-290) is never used and should be removed
CurvePoolBase._unstakeLp(uint256) (contracts/contracts/strategies/curve/CurvePoolBase.sol#292-296) is never used and should be removed
EnumerableSet._at(EnumerableSet.Set,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#129-131) is never used and should be removed
EnumerableSet._length(EnumerableSet.Set) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#115-117) is never used and should be removed
EnumerableSet.add(EnumerableSet.Bytes32Set,bytes32) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#157-159) is never used and should be removed
EnumerableSet.add(EnumerableSet.UintSet,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#296-298) is never used and should be removed
EnumerableSet.at(EnumerableSet.AddressSet,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#261-263) is never used and should be removed
EnumerableSet.at(EnumerableSet.Bytes32Set,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#195-197) is never used and should be removed
EnumerableSet.at(EnumerableSet.UintSet,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#334-336) is never used and should be removed
EnumerableSet.contains(EnumerableSet.Bytes32Set,bytes32) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#174-176) is never used and should be removed
EnumerableSet.contains(EnumerableSet.UintSet,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#313-315) is never used and should be removed
EnumerableSet.length(EnumerableSet.AddressSet) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#247-249) is never used and should be removed
EnumerableSet.length(EnumerableSet.Bytes32Set) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#181-183) is never used and should be removed
EnumerableSet.length(EnumerableSet.UintSet) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#320-322) is never used and should be removed
EnumerableSet.remove(EnumerableSet.Bytes32Set,bytes32) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#167-169) is never used and should be removed
EnumerableSet.remove(EnumerableSet.UintSet,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#306-308) is never used and should be removed
EnumerableSet.values(EnumerableSet.Bytes32Set) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#207-209) is never used and should be removed
EnumerableSet.values(EnumerableSet.UintSet) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#344-355) is never used and should be removed
Math.average(uint256,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/Math.sol#27-30) is never used and should be removed
Math.max(uint256,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/Math.sol#12-14) is never used and should be removed
SafeCast.toInt16(int256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#175-178) is never used and should be removed
SafeCast.toInt256(uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#205-208) is never used and should be removed
SafeCast.toInt32(int256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#157-160) is never used and should be removed
SafeCast.toInt64(int256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#139-142) is never used and should be removed
SafeCast.toInt8(int256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#193-196) is never used and should be removed
SafeCast.toUint128(uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#31-34) is never used and should be removed
SafeCast.toUint16(uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#76-79) is never used and should be removed
SafeCast.toUint256(int256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#103-106) is never used and should be removed
SafeCast.toUint32(uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#61-64) is never used and should be removed
SafeCast.toUint64(uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#46-49) is never used and should be removed
SafeCast.toUint8(uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#91-94) is never used and should be removed
SafeERC20.safeDecreaseAllowance(IERC20,address,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#51-58) is never used and should be removed
SafeERC20.safeIncreaseAllowance(IERC20,address,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#46-49) is never used and should be removed
SafeERC20.safeTransferFrom(IERC20,address,address,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#24-26) is never used and should be removed
Strategy._beforeMigration(address) (contracts/contracts/strategies/Strategy.sol#199) is never used and should be removed
Strategy._rebalance() (contracts/contracts/strategies/Strategy.sol#201-208) is never used and should be removed
Strategy._safeSwapExactInput(address,address,uint256) (contracts/contracts/strategies/Strategy.sol#218-224) is never used and should be removed
Strategy._swapExactInput(address,address,uint256) (contracts/contracts/strategies/Strategy.sol#210-216) is never used and should be removed
Strategy._withdrawHere(uint256) (contracts/contracts/strategies/Strategy.sol#227) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version0.8.9 (contracts/contracts/interfaces/convex/IConvex.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/convex/IConvexToken.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/curve/IAddressProvider.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7

Pragma version0.8.9 (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#4) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/curve/IRegistry.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/curve/IStableSwap.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/curve/ITokenMinter.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/one-oracle/IMasterOracle.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/swapper/IRoutedSwapper.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/strategies/Strategy.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/strategies/convex/3pool/Convex3PlainPool.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/strategies/convex/ConvexBase.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/strategies/curve/3Pool/Curve3PlainPool.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/strategies/curve/CurvePoolBase.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (vesper-commons/contracts/interfaces/vesper/IStrategy.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
```

```
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/IERC20.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Context.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/Math.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#3) allows old versions
Pragma version0.8.9 (vesper-pools/contracts/interfaces/vesper/IGovernable.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (vesper-pools/contracts/interfaces/vesper/IPausable.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (vesper-pools/contracts/interfaces/vesper/IVesperPool.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
solc-0.8.5 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

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

Function IAddressProvider.get_registry() (contracts/contracts/interfaces/curve/IAddressProvider.sol#6) is not in mixedCase
Function IAddressProvider.get_address(uint256) (contracts/contracts/interfaces/curve/IAddressProvider.sol#8) is not in mixedCase
Function ILiquidityGaugeV2.claim_rewards(address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#15) is not in mixedCase
Function ILiquidityGaugeV2.claim_rewards() (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#17) is not in mixedCase
Function ILiquidityGaugeV2.claimable_tokens(address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#19) is not in mixedCase
Function ILiquidityGaugeV2.claimable_reward(address,address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#21) is not in mixedCase
Function ILiquidityGaugeV2.integrate_fraction(address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#23) is not in mixedCase
Function ILiquidityGaugeV2.user_checkpoint(address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#25) is not in mixedCase
Function ILiquidityGaugeV2.reward_integral(address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#27) is not in mixedCase
Function ILiquidityGaugeV2.reward_integral_for(address,address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#29) is not in mixedCase
Function ILiquidityGaugeV2.lp_token() (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#31) is not in mixedCase
Function ILiquidityGaugeV2.reward_count() (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#33) is not in mixedCase
Function ILiquidityGaugeV2.reward_tokens(uint256) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#35) is not in mixedCase
Function ILiquidityGaugeV3.claimable_reward(address,address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#39) is not in mixedCase
Function ILiquidityGaugeV3.claimable_reward_write(address,address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#41) is not in mixedCase
Function IMetapoolFactory.get_underlying_coins(address) (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#7) is not in mixedCase
Function IMetapoolFactory.get_underlying_decimals(address) (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#9) is not in mixedCase
Function IMetapoolFactory.get_coins(address) (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#11) is not in mixedCase
Function IMetapoolFactory.get_n_coins(address) (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#13) is not in mixedCase
Function IMetapoolFactory.get_meta_n_coins(address) (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#15) is not in mixedCase
Function IMetapoolFactory.get_decimals(address) (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#17) is not in mixedCase
Function IMetapoolFactory.get_gauge(address) (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#19) is not in mixedCase
Function IMetapoolFactory.is_meta(address) (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#21) is not in mixedCase
Function IRegistry.get_pool_from_lp_token() (contracts/contracts/interfaces/curve/IRegistry.sol#6) is not in mixedCase
Function IRegistry.get_lp_token(address) (contracts/contracts/interfaces/curve/IRegistry.sol#8) is not in mixedCase
Function IRegistry.get_n_coins(address) (contracts/contracts/interfaces/curve/IRegistry.sol#10) is not in mixedCase
Function IRegistry.get_underlying_coins(address) (contracts/contracts/interfaces/curve/IRegistry.sol#12) is not in mixedCase
Function IRegistry.get_underlying_decimals(address) (contracts/contracts/interfaces/curve/IRegistry.sol#14) is not in mixedCase
Function IRegistry.get_gauges(address) (contracts/contracts/interfaces/curve/IRegistry.sol#16) is not in mixedCase
Function IStableSwap.lp_token() (contracts/contracts/interfaces/curve/IStableSwap.sol#11) is not in mixedCase
Function IStableSwap.A() (contracts/contracts/interfaces/curve/IStableSwap.sol#13) is not in mixedCase
Function IStableSwap.A_precise() (contracts/contracts/interfaces/curve/IStableSwap.sol#15) is not in mixedCase
Function IStableSwap.get_virtual_price() (contracts/contracts/interfaces/curve/IStableSwap.sol#19) is not in mixedCase
Function IStableSwap.get_dy(int128,int128,uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#21-25) is not in mixedCase
Function IStableSwap.get_dy_underlying(int128,int128,uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#27-31) is not in mixedCase
Parameter IStableSwap.exchange(int128,int128,uint256,uint256).min_dy (contracts/contracts/interfaces/curve/IStableSwap.sol#37) is not in mixedCase
Function IStableSwap.calc_withdraw_one_coin(uint256,int128) (contracts/contracts/interfaces/curve/IStableSwap.sol#40) is not in mixedCase
Parameter IStableSwap.calc_withdraw_one_coin(uint256,int128)._token_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#40) is not in mixedCase
Function IStableSwap.remove_liquidity_one_coin(uint256,int128,uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#42-46) is not in mixedCase
Parameter IStableSwap.remove_liquidity_one_coin(uint256,int128,uint256)._token_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#43) is not in mixedCase
Parameter IStableSwap.remove_liquidity_one_coin(uint256,int128,uint256)._min_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#45) is not in mixedCase
Function IStableSwapUnderlying.underlying_coins(uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#54) is not in mixedCase
Function IStableSwapUnderlying.exchange_underlying(int128,int128,uint256,uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#56-61) is not in mixedCase
Parameter IStableSwapUnderlying.exchange_underlying(int128,int128,uint256,uint256).min_dy (contracts/contracts/interfaces/curve/IStableSwap.sol#60) is not in mixedCase
Function IStableSwapUnderlying.remove_liquidity_one_coin(uint256,int128,uint256,bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#63-68) is not in mixedCase
Parameter IStableSwapUnderlying.remove_liquidity_one_coin(uint256,int128,uint256,bool)._token_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#64) is not in mixedCase
Parameter IStableSwapUnderlying.remove_liquidity_one_coin(uint256,int128,uint256,bool)._min_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#66) is not in mixedCase
Parameter IStableSwapUnderlying.remove_liquidity_one_coin(uint256,int128,uint256,bool)._use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#67) is not in mixedCase
Function IStableSwap2x.calc_token_amount(uint256[2],bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#72) is not in mixedCase
Parameter IStableSwap2x.calc_token_amount(uint256[2],bool)._is_deposit (contracts/contracts/interfaces/curve/IStableSwap.sol#72) is not in mixedCase
Function IStableSwap2x.add_liquidity(uint256[2],uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#74) is not in mixedCase
Parameter IStableSwap2x.add_liquidity(uint256[2],uint256)._min_mint_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#74) is not in mixedCase
Function IStableSwap2x.remove_liquidity(uint256,uint256[2]) (contracts/contracts/interfaces/curve/IStableSwap.sol#76) is not in mixedCase
Parameter IStableSwap2x.remove_liquidity(uint256,uint256[2])._min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#76) is not in mixedCase
Function IStableSwap2x.remove_liquidity_imbalance(uint256[2],uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#78-80) is not in mixedCase
Parameter IStableSwap2x.remove_liquidity_imbalance(uint256[2],uint256)._max_burn_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#78) is not in mixedCase
Function IStableSwap3x.calc_token_amount(uint256[3],bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#86) is not in mixedCase
Parameter IStableSwap3x.calc_token_amount(uint256[3],bool)._is_deposit (contracts/contracts/interfaces/curve/IStableSwap.sol#86) is not in mixedCase
Function IStableSwap3x.add_liquidity(uint256[3],uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#86) is not in mixedCase
Parameter IStableSwap3x.add_liquidity(uint256[3],uint256)._min_mint_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#86) is not in mixedCase
Function IStableSwap3x.remove_liquidity(uint256,uint256[3]) (contracts/contracts/interfaces/curve/IStableSwap.sol#88) is not in mixedCase
Parameter IStableSwap3x.remove_liquidity(uint256,uint256[3])._min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#88) is not in mixedCase
Function IStableSwap3x.remove_liquidity_imbalance(uint256[3],uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#90-92) is not in mixedCase
Parameter IStableSwap3x.remove_liquidity_imbalance(uint256[3],uint256)._max_burn_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#90) is not in mixedCase
Function IStableSwap4x.calc_token_amount(uint256[4],bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#96) is not in mixedCase
Parameter IStableSwap4x.calc_token_amount(uint256[4],bool)._is_deposit (contracts/contracts/interfaces/curve/IStableSwap.sol#96) is not in mixedCase
Function IStableSwap4x.add_liquidity(uint256[4],uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#98) is not in mixedCase
Parameter IStableSwap4x.add_liquidity(uint256[4],uint256)._min_mint_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#98) is not in mixedCase
Function IStableSwap4x.remove_liquidity(uint256,uint256[4]) (contracts/contracts/interfaces/curve/IStableSwap.sol#100) is not in mixedCase
Parameter IStableSwap4x.remove_liquidity(uint256,uint256[4])._min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#100) is not in mixedCase
Function IStableSwap4x.remove_liquidity_imbalance(uint256[4],uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#102-104) is not in mixedCase
Parameter IStableSwap4x.remove_liquidity_imbalance(uint256[4],uint256)._max_burn_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#102) is not in mixedCase
Function IStableSwap2xUnderlying.add_liquidity(uint256[2],uint256,bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#108-112) is not in mixedCase
Parameter IStableSwap2xUnderlying.add_liquidity(uint256[2],uint256,bool)._min_mint_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#110) is not in mixedCase
Parameter IStableSwap2xUnderlying.add_liquidity(uint256[2],uint256,bool)._use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#111) is not in mixedCase
Function IStableSwap2xUnderlying.remove_liquidity_imbalance(uint256[2],uint256,bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#114-118) is not in mixedCase
Parameter IStableSwap2xUnderlying.remove_liquidity_imbalance(uint256[2],uint256,bool)._max_burn_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#116) is not in mixedCase
Parameter IStableSwap2xUnderlying.remove_liquidity_imbalance(uint256[2],uint256,bool)._use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#117) is not in mixedCase
Function IStableSwap2xUnderlying.remove_liquidity(uint256,uint256[2],bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#120-124) is not in mixedCase
Parameter IStableSwap2xUnderlying.remove_liquidity(uint256,uint256[2],bool)._min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#122) is not in mixedCase
Parameter IStableSwap2xUnderlying.remove_liquidity(uint256,uint256[2],bool)._use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#123) is not in mixedCase
Function IStableSwap3xUnderlying.add_liquidity(uint256[3],uint256,bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#128-132) is not in mixedCase
Parameter IStableSwap3xUnderlying.add_liquidity(uint256[3],uint256,bool)._min_mint_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#130) is not in mixedCase
Parameter IStableSwap3xUnderlying.add_liquidity(uint256[3],uint256,bool)._use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#131) is not in mixedCase
Function IStableSwap3xUnderlying.remove_liquidity_imbalance(uint256[3],uint256,bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#134-138) is not in mixedCase
Parameter IStableSwap3xUnderlying.remove_liquidity_imbalance(uint256[3],uint256,bool)._max_burn_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#136) is not in mixedCase
Parameter IStableSwap3xUnderlying.remove_liquidity_imbalance(uint256[3],uint256,bool)._use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#137) is not in mixedCase
Function IStableSwap3xUnderlying.remove_liquidity(uint256,uint256[3],bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#140-144) is not in mixedCase
Parameter IStableSwap3xUnderlying.remove_liquidity(uint256,uint256[3],bool)._min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#142) is not in mixedCase
Parameter IStableSwap3xUnderlying.remove_liquidity(uint256,uint256[3],bool)._use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#143) is not in mixedCase
Function IStableSwap4xUnderlying.add_liquidity(uint256[4],uint256,bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#148-152) is not in mixedCase
Parameter IStableSwap4xUnderlying.add_liquidity(uint256[4],uint256,bool)._min_mint_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#150) is not in mixedCase
Parameter IStableSwap4xUnderlying.add_liquidity(uint256[4],uint256,bool)._use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#151) is not in mixedCase
Function IStableSwap4xUnderlying.remove_liquidity_imbalance(uint256[4],uint256,bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#154-158) is not in mixedCase
Parameter IStableSwap4xUnderlying.remove_liquidity_imbalance(uint256[4],uint256,bool)._max_burn_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#156) is not in mixedCase
Parameter IStableSwap4xUnderlying.remove_liquidity_imbalance(uint256[4],uint256,bool)._use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#157) is not in mixedCase
Function IStableSwap4xUnderlying.remove_liquidity(uint256,uint256[4],bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#160-164) is not in mixedCase
Parameter IStableSwap4xUnderlying.remove_liquidity(uint256,uint256[4],bool)._min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#162) is not in mixedCase
Parameter IStableSwap4xUnderlying.remove_liquidity(uint256,uint256[4],bool)._use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#163) is not in mixedCase
Parameter ITokenMinter.mint(address).gauge_addr (contracts/contracts/interfaces/curve/ITokenMinter.sol#9) is not in mixedCase
Parameter Strategy.addKeeper(address)._keeperAddress (contracts/contracts/strategies/Strategy.sol#64) is not in mixedCase
Parameter Strategy.migrate(address)._newStrategy (contracts/contracts/strategies/Strategy.sol#87) is not in mixedCase
Parameter Strategy.removeKeeper(address)._keeperAddress (contracts/contracts/strategies/Strategy.sol#119) is not in mixedCase
Parameter Strategy.sweepERC20(address)._fromToken (contracts/contracts/strategies/Strategy.sol#127) is not in mixedCase
Parameter Strategy.updateFeeCollector(address)._feeCollector (contracts/contracts/strategies/Strategy.sol#156) is not in mixedCase
Parameter Strategy.updateSwapper(IRoutedSwapper)._swapper (contracts/contracts/strategies/Strategy.sol#167) is not in mixedCase
Parameter Strategy.withdraw(uint256)._amount (contracts/contracts/strategies/Strategy.sol#178) is not in mixedCase
Variable CurvePoolBase.CRV (contracts/contracts/strategies/curve/CurvePoolBase.sol#37) is not in mixedCase
Variable CurvePoolBase.NAME (contracts/contracts/strategies/curve/CurvePoolBase.sol#39) is not in mixedCase
Function IStrategy.VERSION() (vesper-commons/contracts/interfaces/vesper/IStrategy.sol#31) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

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

Variable IStableSwap.remove_liquidity_one_coin(uint256,int128,uint256)._min_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#45) is too similar to IStableSwap2xUnderlying.remove_liquidity(uint256,uint256[2],bool).min_amounts
(contracts/contracts/interfaces/curve/IStableSwap.sol#122)
Variable IStableSwapUnderlying.remove_liquidity_one_coin(uint256,int128,uint256,bool)._min_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#66) is too similar to IStableSwap2xUnderlying.remove_liquidity(uint256,uint256[2],boo
l).min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#122)
Variable IStableSwap.remove_liquidity_one_coin(uint256,int128,uint256)._min_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#45) is too similar to IStableSwap3xUnderlying.remove_liquidity(uint256,uint256[3],bool).min_amounts
(contracts/contracts/interfaces/curve/IStableSwap.sol#142)
Variable IStableSwapUnderlying.remove_liquidity_one_coin(uint256,int128,uint256,bool)._min_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#66) is too similar to IStableSwap3xUnderlying.remove_liquidity(uint256,uint256[3],boo
l).min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#142)
Variable IStableSwap.remove_liquidity_one_coin(uint256,int128,uint256)._min_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#45) is too similar to IStableSwap4xUnderlying.remove_liquidity(uint256,uint256[4],bool).min_amounts
(contracts/contracts/interfaces/curve/IStableSwap.sol#162)
Variable IStableSwapUnderlying.remove_liquidity_one_coin(uint256,int128,uint256,bool)._min_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#66) is too similar to IStableSwap4xUnderlying.remove_liquidity(uint256,uint256[4],boo
l).min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#162)
Variable Strategy._beforeMigration(address)._newStrategy (contracts/contracts/strategies/Strategy.sol#199) is too similar to CurvePoolBase._beforeMigration(address).newStrategy_ (contracts/contracts/strategies/curve/CurvePoolBase.sol#155
)
Variable Strategy.migrate(address)._newStrategy (contracts/contracts/strategies/Strategy.sol#87) is too similar to CurvePoolBase._beforeMigration(address).newStrategy_ (contracts/contracts/strategies/curve/CurvePoolBase.sol#155)
Variable IStrategy.migrate(address)._newStrategy (vesper-commons/contracts/interfaces/vesper/IStrategy.sol#24) is too similar to CurvePoolBase._beforeMigration(address).newStrategy_ (contracts/contracts/strategies/curve/CurvePoolBase.sol
#155)
Variable CurvePoolBase._claimRewardsAndConvertTo(address)._rewardToken (contracts/contracts/strategies/curve/CurvePoolBase.sol#187) is too similar to CurvePoolBase.rewardTokens (contracts/contracts/strategies/curve/CurvePoolBase.sol#45)
Variable ConvexBase._getRewardTokens()._rewardToken (contracts/contracts/strategies/convex/ConvexBase.sol#67) is too similar to CurvePoolBase.rewardTokens (contracts/contracts/strategies/curve/CurvePoolBase.sol#45)
Variable ConvexBase._getRewardTokens()._rewardTokens (contracts/contracts/strategies/convex/ConvexBase.sol#62) is too similar to CurvePoolBase.setRewardTokens(address[])._rewardTokens_ (contracts/contracts/strategies/curve/CurvePoolBase.s
ol#326)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-are-too-similar

Convex3PlainPool.slitherConstructorConstantVariables() (contracts/contracts/strategies/convex/3pool/Convex3PlainPool.sol#11-75) uses literals with too many digits:
        - ADDRESS_PROVIDER = IAddressProvider(0x0000000022D53366457F9d5E68Ec105046FC4383) (contracts/contracts/strategies/curve/CurvePoolBase.sol#28)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits
```

Convex4FactoryMetaPool.sol

```
CurvePoolBase._withdrawWere(uint256) (contracts/contracts/strategies/curve/CurvePoolBase.sol#306-322) uses a dangerous strict equality:
        - _lpToBurn == 0 (contracts/contracts/strategies/curve/CurvePoolBase.sol#314)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dangerous-strict-equalities

Reentrancy in Convex4FactoryMetaPool.setRewardTokens(address[]) (contracts/contracts/strategies/convex/4Pool/Convex4FactoryMetaPool.sol#70-78):
        External calls:
        - _claimRewardsAndConvertTo(address(collateralToken)) (contracts/contracts/strategies/convex/4Pool/Convex4FactoryMetaPool.sol#74)
                - require(bool,string)(cvxCrvRewards.getReward(address(this),true),reward-claim-failed) (contracts/contracts/strategies/convex/4Pool/Convex4FactoryMetaPool.sol#49)
                - swapper.swapExactInput(_rewardToken,tokenOut,_amountIn,1,address(this)) (contracts/contracts/strategies/curve/CurvePoolBase.sol#190-193)
        State variables written after the call(s):
        - rewardTokens = _getRewardTokens() (contracts/contracts/strategies/convex/4Pool/Convex4FactoryMetaPool.sol#75)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-1

CurvePoolBase._approveToken(uint256).i (contracts/contracts/strategies/curve/CurvePoolBase.sol#148) is a local variable never initialized
CurvePoolBase._claimRewardsAndConvertTo(address).i (contracts/contracts/strategies/curve/CurvePoolBase.sol#186) is a local variable never initialized
ConvexBase._getRewardTokens()._extraRewardCount (contracts/contracts/strategies/convex/ConvexBase.sol#63) is a local variable never initialized
ConvexBase._getRewardTokens().i (contracts/contracts/strategies/convex/ConvexBase.sol#66) is a local variable never initialized
ConvexBase._getRewardTokens().i_scope_0 (contracts/contracts/strategies/convex/ConvexBase.sol#80) is a local variable never initialized
CurvePoolBase.setRewardTokens(address[]).i (contracts/contracts/strategies/curve/CurvePoolBase.sol#330) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

Strategy._swapExactInput(address,address,uint256) (contracts/contracts/strategies/Strategy.sol#210-216) ignores return value by swapper.swapExactInput(_tokenIn,_tokenOut,_amountIn,1,address(this)) (contracts/contracts/strategies/Strategy.sol#215)
Strategy._safeSwapExactInput(address,address,uint256) (contracts/contracts/strategies/Strategy.sol#218-224) ignores return value by swapper.swapExactInput(_tokenIn,_tokenOut,_amountIn,1,address(this)) (contracts/contracts/strategies/Strategy.sol#223)
CurvePoolBase._claimRewardsAndConvertTo(address) (contracts/contracts/strategies/curve/CurvePoolBase.sol#183-196) ignores return value by swapper.swapExactInput(_rewardToken,tokenOut,_amountIn,1,address(this)) (contracts/contracts/strategies/curve/CurvePoolBase.sol#190-193)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

Address.isContract(address) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#26-35) uses assembly
        - INLINE ASM (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#33)
Address._verifyCallResult(bool,bytes,string) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#171-188) uses assembly
        - INLINE ASM (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#180-183)
EnumerableSet.values(EnumerableSet.AddressSet) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#273-282) uses assembly
        - INLINE ASM (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#277-279)
EnumerableSet.values(EnumerableSet.UintSet) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#346-355) uses assembly
        - INLINE ASM (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#350-352)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Different versions of Solidity is used:
        - Version used: ['0.8.9', '^0.8.0']
        - 0.8.9 (contracts/contracts/interfaces/convex/IConvex.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/convex/IConvexToken.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/curve/IAddressProvider.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/curve/IDepositZap.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#4)
        - 0.8.9 (contracts/contracts/interfaces/curve/IRegistry.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/curve/IStableSwap.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/curve/ITokenMinter.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/one-oracle/IMasterOracle.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/swapper/IRoutedSwapper.sol#3)
        - 0.8.9 (contracts/contracts/strategies/Strategy.sol#3)
        - 0.8.9 (contracts/contracts/strategies/convex/4Pool/Convex4FactoryMetaPool.sol#3)
        - 0.8.9 (contracts/contracts/strategies/convex/ConvexBase.sol#3)
        - 0.8.9 (contracts/contracts/strategies/curve/4Pool/Curve4FactoryMetaPool.sol#3)
        - 0.8.9 (contracts/contracts/strategies/curve/CurvePoolBase.sol#3)
        - 0.8.9 (vesper-commons/contracts/interfaces/vesper/IStrategy.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/IERC20.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Context.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/Math.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#3)
        - 0.8.9 (vesper-pools/contracts/interfaces/vesper/IGovernable.sol#3)
        - 0.8.9 (vesper-pools/contracts/interfaces/vesper/IPausable.sol#3)
        - 0.8.9 (vesper-pools/contracts/interfaces/vesper/IVesperPool.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

Address.functionCall(address,bytes) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#79-81) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#104-106) is never used and should be removed
Address.functionDelegateCall(address,bytes) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#153-155) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#163-169) is never used and should be removed
Address.functionStaticCall(address,bytes) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#125-131) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#139-145) is never used and should be removed
Context._msgData() (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Context.sol#20-23) is never used and should be removed
ConvexBase._calculateCVXRewards(uint256) (contracts/contracts/strategies/convex/ConvexBase.sol#54-59) is never used and should be removed
CurvePoolBase._claimRewards() (contracts/contracts/strategies/curve/CurvePoolBase.sol#169-178) is never used and should be removed
CurvePoolBase._depositToCurve(uint256) (contracts/contracts/strategies/curve/CurvePoolBase.sol#203) is never used and should be removed
CurvePoolBase._quoteLpToCoin(uint256,int128) (contracts/contracts/strategies/curve/CurvePoolBase.sol#259-263) is never used and should be removed
CurvePoolBase._stakeAllLp() (contracts/contracts/strategies/curve/CurvePoolBase.sol#281-286) is never used and should be removed
CurvePoolBase._unstakeAllLp() (contracts/contracts/strategies/curve/CurvePoolBase.sol#288-290) is never used and should be removed
CurvePoolBase._unstakeLp(uint256) (contracts/contracts/strategies/curve/CurvePoolBase.sol#292-296) is never used and should be removed
CurvePoolBase._withdrawFromCurve(uint256,uint256,int128) (contracts/contracts/strategies/curve/CurvePoolBase.sol#298-304) is never used and should be removed
EnumerableSet._at(EnumerableSet.Set,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#129-131) is never used and should be removed
EnumerableSet._length(EnumerableSet.Set) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#115-117) is never used and should be removed
EnumerableSet.add(EnumerableSet.Bytes32Set,bytes32) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#157-159) is never used and should be removed
EnumerableSet.add(EnumerableSet.UintSet,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#296-298) is never used and should be removed
EnumerableSet.at(EnumerableSet.AddressSet,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#261-263) is never used and should be removed
EnumerableSet.at(EnumerableSet.Bytes32Set,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#195-197) is never used and should be removed
EnumerableSet.at(EnumerableSet.UintSet,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#334-336) is never used and should be removed
EnumerableSet.contains(EnumerableSet.Bytes32Set,bytes32) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#174-176) is never used and should be removed
EnumerableSet.contains(EnumerableSet.UintSet,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#313-315) is never used and should be removed
EnumerableSet.length(EnumerableSet.AddressSet) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#247-249) is never used and should be removed
EnumerableSet.length(EnumerableSet.Bytes32Set) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#181-183) is never used and should be removed
EnumerableSet.length(EnumerableSet.UintSet) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#320-322) is never used and should be removed
EnumerableSet.remove(EnumerableSet.Bytes32Set,bytes32) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#167-169) is never used and should be removed
EnumerableSet.remove(EnumerableSet.UintSet,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#306-308) is never used and should be removed
EnumerableSet.values(EnumerableSet.Bytes32Set) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#207-209) is never used and should be removed
EnumerableSet.values(EnumerableSet.UintSet) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#346-355) is never used and should be removed
Math.average(uint256,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/Math.sol#27-30) is never used and should be removed
Math.max(uint256,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/Math.sol#12-14) is never used and should be removed
SafeCast.toInt16(int256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#175-178) is never used and should be removed
SafeCast.toInt256(uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#205-208) is never used and should be removed
SafeCast.toInt32(int256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#157-160) is never used and should be removed
SafeCast.toInt64(int256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#139-142) is never used and should be removed
SafeCast.toInt8(int256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#193-196) is never used and should be removed
SafeCast.toUint128(uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#32-34) is never used and should be removed
SafeCast.toUint16(uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#76-79) is never used and should be removed
SafeCast.toUint256(int256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#103-106) is never used and should be removed
SafeCast.toUint32(uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#61-64) is never used and should be removed
SafeCast.toUint64(uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#46-49) is never used and should be removed
SafeCast.toUint8(uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#91-94) is never used and should be removed
SafeERC20.safeDecreaseAllowance(IERC20,address,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#51-58) is never used and should be removed
SafeERC20.safeIncreaseAllowance(IERC20,address,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#46-49) is never used and should be removed
SafeERC20.safeTransferFrom(IERC20,address,address,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#24-26) is never used and should be removed
Strategy._beforeMigration(address) (contracts/contracts/strategies/Strategy.sol#199) is never used and should be removed
Strategy._rebalance() (contracts/contracts/strategies/Strategy.sol#201-208) is never used and should be removed
Strategy._safeSwapExactInput(address,address,uint256) (contracts/contracts/strategies/Strategy.sol#218-224) is never used and should be removed
Strategy._swapExactInput(address,address,uint256) (contracts/contracts/strategies/Strategy.sol#210-216) is never used and should be removed
Strategy._withdrawWere(uint256) (contracts/contracts/strategies/Strategy.sol#227) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version0.8.9 (contracts/contracts/interfaces/convex/IConvex.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/convex/IConvexToken.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/curve/IAddressProvider.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/curve/IDepositZap.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#4) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/curve/IRegistry.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/curve/IStableSwap.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/curve/ITokenMinter.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/one-oracle/IMasterOracle.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/swapper/IRoutedSwapper.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/strategies/Strategy.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/strategies/convex/4Pool/Convex4FactoryMetaPool.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/strategies/convex/ConvexBase.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/strategies/curve/4Pool/Curve4FactoryMetaPool.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/strategies/curve/CurvePoolBase.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (vesper-commons/contracts/interfaces/vesper/IStrategy.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/IERC20.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Context.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/Math.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#3) allows old versions
Pragma version0.8.9 (vesper-pools/contracts/interfaces/vesper/IGovernable.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (vesper-pools/contracts/interfaces/vesper/IPausable.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (vesper-pools/contracts/interfaces/vesper/IVesperPool.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
solc-0.8.9 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

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

```
Function IAddressProvider.get_registry() (contracts/contracts/interfaces/curve/IAddressProvider.sol#6) is not in mixedCase
Function IAddressProvider.get_address(uint256) (contracts/contracts/interfaces/curve/IAddressProvider.sol#8) is not in mixedCase
Function IDepositZap4x.calc_token_amount(address,uint256[4],bool) (contracts/contracts/interfaces/curve/IDepositZap.sol#6-10) is not in mixedCase
Parameter IDepositZap4x.calc_token_amount(address,uint256[4],bool).is_deposit (contracts/contracts/interfaces/curve/IDepositZap.sol#9) is not in mixedCase
Function IDepositZap4x.add_liquidity(address,uint256[4],uint256) (contracts/contracts/interfaces/curve/IDepositZap.sol#12-16) is not in mixedCase
Parameter IDepositZap4x.add_liquidity(address,uint256[4],uint256)._min_mint_amount (contracts/contracts/interfaces/curve/IDepositZap.sol#15) is not in mixedCase
Function IDepositZap4x.remove_liquidity(address,uint256,uint256[4]) (contracts/contracts/interfaces/curve/IDepositZap.sol#18-22) is not in mixedCase
Parameter IDepositZap4x.remove_liquidity(address,uint256,uint256[4])._min_amounts (contracts/contracts/interfaces/curve/IDepositZap.sol#21) is not in mixedCase
Function IDepositZap4x.remove_liquidity_fee_coin(address,uint256,int128,uint256) (contracts/contracts/interfaces/curve/IDepositZap.sol#24-29) is not in mixedCase
Parameter IDepositZap4x.remove_liquidity_one_coin(address,uint256,int128,uint256)._token_amount (contracts/contracts/interfaces/curve/IDepositZap.sol#26) is not in mixedCase
Parameter IDepositZap4x.remove_liquidity_one_coin(address,uint256,int128,uint256)._min_amount (contracts/contracts/interfaces/curve/IDepositZap.sol#28) is not in mixedCase
Function IDepositZap4x.calc_withdraw_one_coin(address,uint256,int128) (contracts/contracts/interfaces/curve/IDepositZap.sol#31-35) is not in mixedCase
Parameter IDepositZap4x.calc_withdraw_one_coin(address,uint256,int128)._token_amount (contracts/contracts/interfaces/curve/IDepositZap.sol#33) is not in mixedCase
Function ILiquidityGaugeV2.claim_rewards(address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#15) is not in mixedCase
Function ILiquidityGaugeV2.claim_rewards() (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#17) is not in mixedCase
Function ILiquidityGaugeV2.claimable_tokens(address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#19) is not in mixedCase
Function ILiquidityGaugeV2.claimable_reward(address,address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#21) is not in mixedCase
Function ILiquidityGaugeV2.integrate_fraction(address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#23) is not in mixedCase
Function ILiquidityGaugeV2.user_checkpoint(address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#25) is not in mixedCase
Function ILiquidityGaugeV2.reward_integral(address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#27) is not in mixedCase
Function ILiquidityGaugeV2.reward_integral_for(address,address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#29) is not in mixedCase
Function ILiquidityGaugeV2.lp_token() (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#31) is not in mixedCase
Function ILiquidityGaugeV2.reward_count() (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#33) is not in mixedCase
Function ILiquidityGaugeV2.reward_tokens(uint256) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#35) is not in mixedCase
Function ILiquidityGaugeV3.claimable_reward(address,address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#39) is not in mixedCase
Function ILiquidityGaugeV3.claimable_reward_write(address,address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#41) is not in mixedCase
Function IMetapoolFactory.get_underlying_coins(address) (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#7) is not in mixedCase
Function IMetapoolFactory.get_underlying_decimals(address) (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#9) is not in mixedCase
Function IMetapoolFactory.get_coins(address) (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#11) is not in mixedCase
Function IMetapoolFactory.get_n_coins(address) (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#13) is not in mixedCase
Function IMetapoolFactory.get_meta_n_coins(address) (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#15) is not in mixedCase
Function IMetapoolFactory.get_decimals(address) (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#17) is not in mixedCase
Function IMetapoolFactory.get_gauge(address) (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#19) is not in mixedCase
Function IMetapoolFactory.is_meta(address) (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#21) is not in mixedCase
Function IRegistry.get_pool_from_lp_token() (contracts/contracts/interfaces/curve/IRegistry.sol#6) is not in mixedCase
Function IRegistry.get_lp_token(address) (contracts/contracts/interfaces/curve/IRegistry.sol#8) is not in mixedCase
Function IRegistry.get_n_coins(address) (contracts/contracts/interfaces/curve/IRegistry.sol#10) is not in mixedCase
Function IRegistry.get_underlying_coins(address) (contracts/contracts/interfaces/curve/IRegistry.sol#12) is not in mixedCase
Function IRegistry.get_underlying_decimals(address) (contracts/contracts/interfaces/curve/IRegistry.sol#14) is not in mixedCase
Function IRegistry.get_gauges(address) (contracts/contracts/interfaces/curve/IRegistry.sol#16) is not in mixedCase
Function ISTableSwap.lp_token() (contracts/contracts/interfaces/curve/ISTableSwap.sol#11) is not in mixedCase
Function ISTableSwap.A() (contracts/contracts/interfaces/curve/ISTableSwap.sol#13) is not in mixedCase
Function ISTableSwap.A_precise() (contracts/contracts/interfaces/curve/ISTableSwap.sol#15) is not in mixedCase
Function ISTableSwap.get_virtual_price() (contracts/contracts/interfaces/curve/ISTableSwap.sol#19) is not in mixedCase
Function ISTableSwap.get_dy(int128,int128,uint256) (contracts/contracts/interfaces/curve/ISTableSwap.sol#21-25) is not in mixedCase
Function ISTableSwap.get_dy_underlying(int128,int128,uint256) (contracts/contracts/interfaces/curve/ISTableSwap.sol#27-31) is not in mixedCase
Parameter ISTableSwap.exchange(int128,int128,uint256,uint256).min_dy (contracts/contracts/interfaces/curve/ISTableSwap.sol#37) is not in mixedCase
Function ISTableSwap.calc_withdraw_one_coin(uint256,int128) (contracts/contracts/interfaces/curve/ISTableSwap.sol#40) is not in mixedCase
Parameter ISTableSwap.calc_withdraw_one_coin(uint256,int128)._token_amount (contracts/contracts/interfaces/curve/ISTableSwap.sol#40) is not in mixedCase
Function ISTableSwap.remove_liquidity_one_coin(uint256,int128,uint256) (contracts/contracts/interfaces/curve/ISTableSwap.sol#42-46) is not in mixedCase
Parameter ISTableSwap.remove_liquidity_one_coin(uint256,int128,uint256)._token_amount (contracts/contracts/interfaces/curve/ISTableSwap.sol#43) is not in mixedCase
Parameter ISTableSwap.remove_liquidity_one_coin(uint256,int128,uint256)._min_amount (contracts/contracts/interfaces/curve/ISTableSwap.sol#45) is not in mixedCase
Function ISTableSwapUnderlying.underlying_coins(uint256) (contracts/contracts/interfaces/curve/ISTableSwap.sol#54) is not in mixedCase
Function ISTableSwapUnderlying.exchange_underlying(int128,int128,uint256,uint256) (contracts/contracts/interfaces/curve/ISTableSwap.sol#56-61) is not in mixedCase
Parameter ISTableSwapUnderlying.exchange_underlying(int128,int128,uint256,uint256).min_dy (contracts/contracts/interfaces/curve/ISTableSwap.sol#60) is not in mixedCase
Function ISTableSwapUnderlying.remove_liquidity_one_coin(uint256,int128,uint256,bool) (contracts/contracts/interfaces/curve/ISTableSwap.sol#63-68) is not in mixedCase
Parameter ISTableSwapUnderlying.remove_liquidity_one_coin(uint256,int128,uint256,bool)._token_amount (contracts/contracts/interfaces/curve/ISTableSwap.sol#64) is not in mixedCase
Parameter ISTableSwapUnderlying.remove_liquidity_one_coin(uint256,int128,uint256,bool)._min_amount (contracts/contracts/interfaces/curve/ISTableSwap.sol#66) is not in mixedCase
Parameter ISTableSwapUnderlying.remove_liquidity_one_coin(uint256,int128,uint256,bool)._use_underlying (contracts/contracts/interfaces/curve/ISTableSwap.sol#67) is not in mixedCase
Function ISTableSwap2x.calc_token_amount(uint256[2],bool) (contracts/contracts/interfaces/curve/ISTableSwap.sol#72) is not in mixedCase
Parameter ISTableSwap2x.calc_token_amount(uint256[2],bool).is_deposit (contracts/contracts/interfaces/curve/ISTableSwap.sol#72) is not in mixedCase
Function ISTableSwap2x.add_liquidity(uint256[2],uint256) (contracts/contracts/interfaces/curve/ISTableSwap.sol#74) is not in mixedCase
Parameter ISTableSwap2x.add_liquidity(uint256[2],uint256)._min_mint_amount (contracts/contracts/interfaces/curve/ISTableSwap.sol#74) is not in mixedCase
Function ISTableSwap2x.remove_liquidity(uint256,uint256[2]) (contracts/contracts/interfaces/curve/ISTableSwap.sol#76) is not in mixedCase
Parameter ISTableSwap2x.remove_liquidity(uint256,uint256[2])._min_amounts (contracts/contracts/interfaces/curve/ISTableSwap.sol#76) is not in mixedCase
Function ISTableSwap2x.remove_liquidity_imbalance(uint256[2],uint256) (contracts/contracts/interfaces/curve/ISTableSwap.sol#78-80) is not in mixedCase
Parameter ISTableSwap2x.remove_liquidity_imbalance(uint256[2],uint256)._max_burn_amount (contracts/contracts/interfaces/curve/ISTableSwap.sol#78) is not in mixedCase
Function ISTableSwap3x.calc_token_amount(uint256[3],bool) (contracts/contracts/interfaces/curve/ISTableSwap.sol#84) is not in mixedCase
Parameter ISTableSwap3x.calc_token_amount(uint256[3],bool).is_deposit (contracts/contracts/interfaces/curve/ISTableSwap.sol#84) is not in mixedCase
Function ISTableSwap3x.add_liquidity(uint256[3],uint256) (contracts/contracts/interfaces/curve/ISTableSwap.sol#86) is not in mixedCase
Parameter ISTableSwap3x.add_liquidity(uint256[3],uint256)._min_mint_amount (contracts/contracts/interfaces/curve/ISTableSwap.sol#86) is not in mixedCase
Function ISTableSwap3x.remove_liquidity(uint256,uint256[3]) (contracts/contracts/interfaces/curve/ISTableSwap.sol#88) is not in mixedCase
Parameter ISTableSwap3x.remove_liquidity(uint256,uint256[3])._min_amounts (contracts/contracts/interfaces/curve/ISTableSwap.sol#88) is not in mixedCase
Function ISTableSwap3x.remove_liquidity_imbalance(uint256[3],uint256) (contracts/contracts/interfaces/curve/ISTableSwap.sol#90-92) is not in mixedCase
Parameter ISTableSwap3x.remove_liquidity_imbalance(uint256[3],uint256)._max_burn_amount (contracts/contracts/interfaces/curve/ISTableSwap.sol#90) is not in mixedCase
Function ISTableSwap4x.calc_token_amount(uint256[4],bool) (contracts/contracts/interfaces/curve/ISTableSwap.sol#96) is not in mixedCase
Parameter ISTableSwap4x.calc_token_amount(uint256[4],bool).is_deposit (contracts/contracts/interfaces/curve/ISTableSwap.sol#96) is not in mixedCase
Function ISTableSwap4x.add_liquidity(uint256[4],uint256) (contracts/contracts/interfaces/curve/ISTableSwap.sol#98) is not in mixedCase
Parameter ISTableSwap4x.add_liquidity(uint256[4],uint256)._min_mint_amount (contracts/contracts/interfaces/curve/ISTableSwap.sol#98) is not in mixedCase
Function ISTableSwap4x.remove_liquidity(uint256,uint256[4]) (contracts/contracts/interfaces/curve/ISTableSwap.sol#100) is not in mixedCase
Parameter ISTableSwap4x.remove_liquidity(uint256,uint256[4])._min_amounts (contracts/contracts/interfaces/curve/ISTableSwap.sol#100) is not in mixedCase
Function ISTableSwap4x.remove_liquidity_imbalance(uint256[4],uint256) (contracts/contracts/interfaces/curve/ISTableSwap.sol#102-104) is not in mixedCase
Parameter ISTableSwap4x.remove_liquidity_imbalance(uint256[4],uint256)._max_burn_amount (contracts/contracts/interfaces/curve/ISTableSwap.sol#102) is not in mixedCase
Function ISTableSwap2xUnderlying.add_liquidity(uint256[2],uint256,bool) (contracts/contracts/interfaces/curve/ISTableSwap.sol#108-112) is not in mixedCase
Parameter ISTableSwap2xUnderlying.add_liquidity(uint256[2],uint256,bool).min_mint_amount (contracts/contracts/interfaces/curve/ISTableSwap.sol#110) is not in mixedCase
Parameter ISTableSwap2xUnderlying.add_liquidity(uint256[2],uint256,bool).use_underlying (contracts/contracts/interfaces/curve/ISTableSwap.sol#111) is not in mixedCase
Function ISTableSwap2xUnderlying.remove_liquidity_imbalance(uint256[2],uint256,bool) (contracts/contracts/interfaces/curve/ISTableSwap.sol#114-118) is not in mixedCase
Parameter ISTableSwap2xUnderlying.remove_liquidity_imbalance(uint256[2],uint256,bool).max_burn_amount (contracts/contracts/interfaces/curve/ISTableSwap.sol#116) is not in mixedCase
Parameter ISTableSwap2xUnderlying.remove_liquidity_imbalance(uint256[2],uint256,bool).use_underlying (contracts/contracts/interfaces/curve/ISTableSwap.sol#117) is not in mixedCase
Function ISTableSwap2xUnderlying.remove_liquidity(uint256,uint256[2],bool) (contracts/contracts/interfaces/curve/ISTableSwap.sol#120-124) is not in mixedCase
Parameter ISTableSwap2xUnderlying.remove_liquidity(uint256,uint256[2],bool).min_amounts (contracts/contracts/interfaces/curve/ISTableSwap.sol#122) is not in mixedCase
Parameter ISTableSwap2xUnderlying.remove_liquidity(uint256,uint256[2],bool).use_underlying (contracts/contracts/interfaces/curve/ISTableSwap.sol#123) is not in mixedCase
Function ISTableSwap3xUnderlying.add_liquidity(uint256[3],uint256,bool) (contracts/contracts/interfaces/curve/ISTableSwap.sol#128-132) is not in mixedCase
Parameter ISTableSwap3xUnderlying.add_liquidity(uint256[3],uint256,bool).min_mint_amount (contracts/contracts/interfaces/curve/ISTableSwap.sol#130) is not in mixedCase
Parameter ISTableSwap3xUnderlying.add_liquidity(uint256[3],uint256,bool).use_underlying (contracts/contracts/interfaces/curve/ISTableSwap.sol#131) is not in mixedCase
Function ISTableSwap3xUnderlying.remove_liquidity_imbalance(uint256[3],uint256,bool) (contracts/contracts/interfaces/curve/ISTableSwap.sol#134-138) is not in mixedCase
Parameter ISTableSwap3xUnderlying.remove_liquidity_imbalance(uint256[3],uint256,bool).max_burn_amount (contracts/contracts/interfaces/curve/ISTableSwap.sol#136) is not in mixedCase
Parameter ISTableSwap3xUnderlying.remove_liquidity_imbalance(uint256[3],uint256,bool).use_underlying (contracts/contracts/interfaces/curve/ISTableSwap.sol#137) is not in mixedCase
Function ISTableSwap3xUnderlying.remove_liquidity(uint256,uint256[3],bool) (contracts/contracts/interfaces/curve/ISTableSwap.sol#140-144) is not in mixedCase
Parameter ISTableSwap3xUnderlying.remove_liquidity(uint256,uint256[3],bool).min_amounts (contracts/contracts/interfaces/curve/ISTableSwap.sol#142) is not in mixedCase
Parameter ISTableSwap3xUnderlying.remove_liquidity(uint256,uint256[3],bool).use_underlying (contracts/contracts/interfaces/curve/ISTableSwap.sol#143) is not in mixedCase
Function ISTableSwap4xUnderlying.add_liquidity(uint256[4],uint256,bool) (contracts/contracts/interfaces/curve/ISTableSwap.sol#148-152) is not in mixedCase
Parameter ISTableSwap4xUnderlying.add_liquidity(uint256[4],uint256,bool).min_mint_amount (contracts/contracts/interfaces/curve/ISTableSwap.sol#150) is not in mixedCase
Parameter ISTableSwap4xUnderlying.add_liquidity(uint256[4],uint256,bool).use_underlying (contracts/contracts/interfaces/curve/ISTableSwap.sol#151) is not in mixedCase
Function ISTableSwap4xUnderlying.remove_liquidity_imbalance(uint256[4],uint256,bool) (contracts/contracts/interfaces/curve/ISTableSwap.sol#154-158) is not in mixedCase
Parameter ISTableSwap4xUnderlying.remove_liquidity_imbalance(uint256[4],uint256,bool).max_burn_amount (contracts/contracts/interfaces/curve/ISTableSwap.sol#156) is not in mixedCase
Parameter ISTableSwap4xUnderlying.remove_liquidity_imbalance(uint256[4],uint256,bool).use_underlying (contracts/contracts/interfaces/curve/ISTableSwap.sol#157) is not in mixedCase
Function ISTableSwap4xUnderlying.remove_liquidity(uint256,uint256[4],bool) (contracts/contracts/interfaces/curve/ISTableSwap.sol#160-164) is not in mixedCase
Parameter ISTableSwap4xUnderlying.remove_liquidity(uint256,uint256[4],bool).min_amounts (contracts/contracts/interfaces/curve/ISTableSwap.sol#162) is not in mixedCase
Parameter ISTableSwap4xUnderlying.remove_liquidity(uint256,uint256[4],bool).use_underlying (contracts/contracts/interfaces/curve/ISTableSwap.sol#163) is not in mixedCase
Parameter ITokenMinter.mint(address).gauge_addr (contracts/contracts/interfaces/curve/ITokenMinter.sol#9) is not in mixedCase
Parameter Strategy.addKeeper(address)._keeperAddress (contracts/contracts/strategies/Strategy.sol#64) is not in mixedCase
Parameter Strategy.migrate(address)._newStrategy (contracts/contracts/strategies/Strategy.sol#87) is not in mixedCase
Parameter Strategy.removeKeeper(address)._keeperAddress (contracts/contracts/strategies/Strategy.sol#119) is not in mixedCase
Parameter Strategy.sweepERC20(address)._fromToken (contracts/contracts/strategies/Strategy.sol#127) is not in mixedCase
Parameter Strategy.updateFeeCollector(address)._feeCollector (contracts/contracts/strategies/Strategy.sol#156) is not in mixedCase
Parameter Strategy.updateSwapper(ISRouterSwapper)._swapper (contracts/contracts/strategies/Strategy.sol#167) is not in mixedCase
Parameter Strategy.withdraw(uint256)._amount (contracts/contracts/strategies/Strategy.sol#178) is not in mixedCase
Variable CurvePoolBase.CRV (contracts/contracts/strategies/curve/CurvePoolBase.sol#37) is not in mixedCase
Variable CurvePoolBase.XAMT (contracts/contracts/strategies/curve/CurvePoolBase.sol#39) is not in mixedCase
Function ISStrategy.VERSION() (vesper-commons/contracts/interfaces/vesper/IStrategy.sol#31) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

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

Variable ISTableSwap.remove_liquidity_one_coin(uint256,int128,uint256)._min_amount (contracts/contracts/interfaces/curve/ISTableSwap.sol#45) is too similar to ISTableSwap2xUnderlying.remove_liquidity(uint256,uint256[2],bool).min_amounts (contracts/contracts/interfaces/curve/ISTableSwap.sol#121)
Variable ISTableSwapUnderlying.remove_liquidity_one_coin(uint256,int128,uint256,bool)._min_amount (contracts/contracts/interfaces/curve/ISTableSwap.sol#66) is too similar to ISTableSwap2xUnderlying.remove_liquidity(uint256,uint256[2],bool).min_amounts (contracts/contracts/interfaces/curve/ISTableSwap.sol#122)
Variable ISTableSwap.remove_liquidity_one_coin(uint256,int128,uint256)._min_amount (contracts/contracts/interfaces/curve/ISTableSwap.sol#45) is too similar to ISTableSwap3xUnderlying.remove_liquidity(uint256,uint256[3],bool).min_amounts (contracts/contracts/interfaces/curve/ISTableSwap.sol#142)
Variable ISTableSwapUnderlying.remove_liquidity_one_coin(uint256,int128,uint256,bool)._min_amount (contracts/contracts/interfaces/curve/ISTableSwap.sol#66) is too similar to ISTableSwap3xUnderlying.remove_liquidity(uint256,uint256[3],bool).min_amounts (contracts/contracts/interfaces/curve/ISTableSwap.sol#142)
Variable ISTableSwapUnderlying.remove_liquidity_one_coin(uint256,int128,uint256,bool)._min_amount (contracts/contracts/interfaces/curve/ISTableSwap.sol#66) is too similar to ISTableSwap4xUnderlying.remove_liquidity(uint256,uint256[4],bool).min_amounts (contracts/contracts/interfaces/curve/ISTableSwap.sol#162)
Variable ISTableSwap.remove_liquidity_one_coin(uint256,int128,uint256)._min_amount (contracts/contracts/interfaces/curve/ISTableSwap.sol#45) is too similar to ISTableSwap4xUnderlying.remove_liquidity(uint256,uint256[4],bool).min_amounts (contracts/contracts/interfaces/curve/ISTableSwap.sol#162)
Variable Strategy.migrate(address)._newStrategy (contracts/contracts/strategies/Strategy.sol#87) is too similar to CurvePoolBase._beforeMigration(address).newStrategy_ (contracts/contracts/strategies/curve/CurvePoolBase.sol#155)
Variable CurvePoolBase._claimRewardsAndConvertTo(address)._rewardToken (contracts/contracts/strategies/curve/CurvePoolBase.sol#187) is too similar to CurvePoolBase.rewardToken (contracts/contracts/strategies/curve/CurvePoolBase.sol#45)
Variable ConvexBase._getRewardTokens()._rewardToken (contracts/contracts/strategies/convex/ConvexBase.sol#67) is too similar to CurvePoolBase.rewardTokens (contracts/contracts/strategies/curve/CurvePoolBase.sol#45)
Variable Strategy._beforeMigration(address)._newStrategy (contracts/contracts/strategies/Strategy.sol#199) is too similar to CurvePoolBase._beforeMigration(address).newStrategy_ (contracts/contracts/strategies/curve/CurvePoolBase.sol#155)
Variable IStrategy.migrate(address)._newStrategy (vesper-commons/contracts/interfaces/vesper/IStrategy.sol#24) is too similar to CurvePoolBase._beforeMigration(address).newStrategy_ (contracts/contracts/strategies/curve/CurvePoolBase.sol#155)
Variable ConvexBase._getRewardTokens()._rewardTokens (contracts/contracts/strategies/convex/ConvexBase.sol#62) is too similar to CurvePoolBase.setRewardTokens(address[]).rewardTokens_ (contracts/contracts/strategies/curve/CurvePoolBase.sol#326)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-are-too-similar

Convex4FactoryMetaPool.slitherConstructorConstantVariables() (contracts/contracts/strategies/convex/4Pool/Convex4FactoryMetaPool.sol#9-83) uses literals with too many digits:
        - ADDRESS_PROVIDER = IAddressProvider(0x0000000022D53366457F9d5E68Ec105046FC4383) (contracts/contracts/strategies/curve/CurvePoolBase.sol#28)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits
```

AUTOMATED TESTING

Convex4MetaPool.sol

```
CurvePoolBase._withdrawWere(uint256) (contracts/contracts/strategies/curve/CurvePoolBase.sol#306-322) uses a dangerous strict equality:
        - _lpToBurn == 0 (contracts/contracts/strategies/curve/CurvePoolBase.sol#314)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dangerous-strict-equalities

Reentrancy in Convex4MetaPool.setRewardTokens(address[]) (contracts/contracts/strategies/convex/4Pool/Convex4MetaPool.sol#70-78):
        External calls:
        - _claimRewardsAndConvertTo(address(collateralToken)) (contracts/contracts/strategies/convex/4Pool/Convex4MetaPool.sol#74)
                - require(bool,string)(cvxCrvRewards.getReward(address(this),true),reward-claim-failed) (contracts/contracts/strategies/convex/4Pool/Convex4MetaPool.sol#49)
                - swapper.swapExactInput(_rewardToken,tokenOut,_amountIn,1,address(this)) (contracts/contracts/strategies/curve/CurvePoolBase.sol#190-193)
        State variables written after the call(s):
        - rewardTokens = _getRewardTokens() (contracts/contracts/strategies/convex/4Pool/Convex4MetaPool.sol#75)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-1

ConvexBase._getRewardTokens().i_scope_0 (contracts/contracts/strategies/convex/ConvexBase.sol#80) is a local variable never initialized
CurvePoolBase.setRewardTokens(address[]).i (contracts/contracts/strategies/curve/CurvePoolBase.sol#330) is a local variable never initialized
ConvexBase._getRewardTokens()._extraRewardCount (contracts/contracts/strategies/convex/ConvexBase.sol#63) is a local variable never initialized
CurvePoolBase._claimRewardsAndConvertTo(address).i (contracts/contracts/strategies/curve/CurvePoolBase.sol#186) is a local variable never initialized
ConvexBase._getRewardTokens().i (contracts/contracts/strategies/convex/ConvexBase.sol#66) is a local variable never initialized
CurvePoolBase._approveToken(uint256).i (contracts/contracts/strategies/curve/CurvePoolBase.sol#148) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

Strategy._swapExactInput(address,address,uint256) (contracts/contracts/strategies/Strategy.sol#210-216) ignores return value by swapper.swapExactInput(_tokenIn,_tokenOut,_amountIn,1,address(this)) (contracts/contracts/strategies/Strategy.sol#215)
Strategy._safeSwapExactInput(address,address,uint256) (contracts/contracts/strategies/Strategy.sol#218-224) ignores return value by swapper.swapExactInput(_tokenIn,_tokenOut,_amountIn,1,address(this)) (contracts/contracts/strategies/Strategy.sol#223)
CurvePoolBase._claimRewardsAndConvertTo(address) (contracts/contracts/strategies/curve/CurvePoolBase.sol#183-196) ignores return value by swapper.swapExactInput(_rewardToken,tokenOut,_amountIn,1,address(this)) (contracts/contracts/strategies/curve/CurvePoolBase.sol#190-193)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

Address.isContract(address) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#26-35) uses assembly
        - INLINE ASM (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#33)
Address._verifyCallResult(bool,bytes,string) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#171-188) uses assembly
        - INLINE ASM (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#180-183)
EnumerableSet.values(EnumerableSet.AddressSet) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#273-282) uses assembly
        - INLINE ASM (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#277-279)
EnumerableSet.values(EnumerableSet.UintSet) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#346-355) uses assembly
        - INLINE ASM (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#350-352)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Different versions of Solidity is used:
        - Version used: ['^0.8.9', '^0.8.0']
        - 0.8.9 (contracts/contracts/interfaces/convex/IConvex.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/convex/IConvexToken.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/curve/IAddressProvider.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/curve/IDeposit.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#4)
        - 0.8.9 (contracts/contracts/interfaces/curve/IRegistry.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/curve/IStableSwap.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/curve/ITokenMinter.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/one-oracle/IMasterOracle.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/swapper/IRoutedSwapper.sol#3)
        - 0.8.9 (contracts/contracts/strategies/Strategy.sol#3)
        - 0.8.9 (contracts/contracts/strategies/convex/4Pool/Convex4MetaPool.sol#3)
        - 0.8.9 (contracts/contracts/strategies/convex/ConvexBase.sol#3)
        - 0.8.9 (contracts/contracts/strategies/curve/4Pool/Curve4PlainOr4MetaPool.sol#3)
        - 0.8.9 (contracts/contracts/strategies/curve/CurvePoolBase.sol#3)
        - 0.8.9 (vesper-commons/contracts/interfaces/vesper/IStrategy.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/IERC20.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Context.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/Math.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#3)
        - 0.8.9 (vesper-pools/contracts/interfaces/vesper/IGovernable.sol#3)
        - 0.8.9 (vesper-pools/contracts/interfaces/vesper/IPausable.sol#3)
        - 0.8.9 (vesper-pools/contracts/interfaces/vesper/IVesperPool.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

Address.functionCall(address,bytes) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#79-81) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#104-106) is never used and should be removed
Address.functionDelegateCall(address,bytes) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#153-155) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#163-169) is never used and should be removed
Address.functionStaticCall(address,bytes) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#129-131) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#139-145) is never used and should be removed
Context._msgData() (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Context.sol#20-23) is never used and should be removed
ConvexBase._calculateCVXRewards(uint256) (contracts/contracts/strategies/convex/ConvexBase.sol#34-59) is never used and should be removed
CurvePoolBase._claimRewards() (contracts/contracts/strategies/curve/CurvePoolBase.sol#168-170) is never used and should be removed
CurvePoolBase._depositToCurve(uint256) (contracts/contracts/strategies/curve/CurvePoolBase.sol#203) is never used and should be removed
CurvePoolBase._quoteLpToCoin(uint256,int128) (contracts/contracts/strategies/curve/CurvePoolBase.sol#259-263) is never used and should be removed
CurvePoolBase._stakeAllLp() (contracts/contracts/strategies/curve/CurvePoolBase.sol#281-286) is never used and should be removed
CurvePoolBase._unstakeAllLp() (contracts/contracts/strategies/curve/CurvePoolBase.sol#288-290) is never used and should be removed
CurvePoolBase._unstakeLp(uint256) (contracts/contracts/strategies/curve/CurvePoolBase.sol#292-296) is never used and should be removed
CurvePoolBase._withdrawFromCurve(uint256,uint256,int128) (contracts/contracts/strategies/curve/CurvePoolBase.sol#298-304) is never used and should be removed
EnumerableSet._at(EnumerableSet.Set,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#129-131) is never used and should be removed
EnumerableSet._length(EnumerableSet.Set) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#115-117) is never used and should be removed
EnumerableSet.add(EnumerableSet.Bytes32Set,bytes32) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#157-159) is never used and should be removed
EnumerableSet.add(EnumerableSet.UintSet,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#296-298) is never used and should be removed
EnumerableSet.at(EnumerableSet.AddressSet,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#261-263) is never used and should be removed
EnumerableSet.at(EnumerableSet.Bytes32Set,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#195-197) is never used and should be removed
EnumerableSet.at(EnumerableSet.UintSet,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#334-336) is never used and should be removed
EnumerableSet.contains(EnumerableSet.Bytes32Set,bytes32) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#174-176) is never used and should be removed
EnumerableSet.contains(EnumerableSet.UintSet,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#313-315) is never used and should be removed
EnumerableSet.length(EnumerableSet.AddressSet) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#247-249) is never used and should be removed
EnumerableSet.length(EnumerableSet.Bytes32Set) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#181-183) is never used and should be removed
EnumerableSet.length(EnumerableSet.UintSet) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#320-322) is never used and should be removed
EnumerableSet.remove(EnumerableSet.Bytes32Set,bytes32) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#167-169) is never used and should be removed
EnumerableSet.remove(EnumerableSet.UintSet,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#306-308) is never used and should be removed
EnumerableSet.values(EnumerableSet.Bytes32Set) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#207-209) is never used and should be removed
EnumerableSet.values(EnumerableSet.UintSet) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#346-355) is never used and should be removed
Math.average(uint256,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/Math.sol#27-30) is never used and should be removed
Math.max(uint256,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/Math.sol#12-14) is never used and should be removed
SafeCast.toInt16(int256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#175-178) is never used and should be removed
SafeCast.toInt256(uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#205-208) is never used and should be removed
SafeCast.toInt32(int256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#157-160) is never used and should be removed
SafeCast.toInt64(int256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#139-142) is never used and should be removed
SafeCast.toInt8(int256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#193-196) is never used and should be removed
SafeCast.toUint128(uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#31-34) is never used and should be removed
SafeCast.toUint16(uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#76-79) is never used and should be removed
SafeCast.toUint256(int256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#103-106) is never used and should be removed
SafeCast.toUint32(uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#61-64) is never used and should be removed
SafeCast.toUint64(uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#46-49) is never used and should be removed
SafeCast.toUint8(uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#91-94) is never used and should be removed
SafeERC20.safeDecreaseAllowance(IERC20,address,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#51-58) is never used and should be removed
SafeERC20.safeIncreaseAllowance(IERC20,address,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#46-49) is never used and should be removed
SafeERC20.safeTransferFrom(IERC20,address,address,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#24-26) is never used and should be removed
Strategy._beforeMigration(address) (contracts/contracts/strategies/Strategy.sol#199) is never used and should be removed
Strategy._rebalance() (contracts/contracts/strategies/Strategy.sol#201-208) is never used and should be removed
Strategy._safeSwapExactInput(address,address,uint256) (contracts/contracts/strategies/Strategy.sol#218-224) is never used and should be removed
Strategy._swapExactInput(address,address,uint256) (contracts/contracts/strategies/Strategy.sol#210-216) is never used and should be removed
Strategy._withdrawWere(uint256) (contracts/contracts/strategies/Strategy.sol#227) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version0.8.9 (contracts/contracts/interfaces/convex/IConvex.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/convex/IConvexToken.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/curve/IAddressProvider.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/curve/IDeposit.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#4) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/curve/IRegistry.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/curve/IStableSwap.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/curve/ITokenMinter.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/one-oracle/IMasterOracle.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/swapper/IRoutedSwapper.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/strategies/Strategy.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/strategies/convex/4Pool/Convex4MetaPool.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/strategies/convex/ConvexBase.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/strategies/curve/4Pool/Curve4PlainOr4MetaPool.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/strategies/curve/CurvePoolBase.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (vesper-commons/contracts/interfaces/vesper/IStrategy.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/IERC20.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Context.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/Math.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#3) allows old versions
Pragma version0.8.9 (vesper-pools/contracts/interfaces/vesper/IGovernable.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (vesper-pools/contracts/interfaces/vesper/IPausable.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (vesper-pools/contracts/interfaces/vesper/IVesperPool.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
solc-0.8.9 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

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

AUTOMATED TESTING

```
Function IAddressProvider.get_registry() (contracts/contracts/interfaces/curve/IAddressProvider.sol#6) is not in mixedCase
Function IAddressProvider.get_address(uint256) (contracts/contracts/interfaces/curve/IAddressProvider.sol#8) is not in mixedCase
Function IDeposit4x.add_liquidity(uint256[4],uint256) (contracts/contracts/interfaces/curve/IDeposit.sol#6) is not in mixedCase
Parameter IDeposit4x.add_liquidity(uint256[4],uint256)._min_mint_amount (contracts/contracts/interfaces/curve/IDeposit.sol#6) is not in mixedCase
Function IDeposit4x.remove_liquidity(uint256,uint256[4]) (contracts/contracts/interfaces/curve/IDeposit.sol#8) is not in mixedCase
Parameter IDeposit4x.remove_liquidity(uint256,uint256[4])._min_amounts (contracts/contracts/interfaces/curve/IDeposit.sol#8) is not in mixedCase
Function IDeposit4x.remove_liquidity_one_coin(uint256,int128,uint256) (contracts/contracts/interfaces/curve/IDeposit.sol#10-14) is not in mixedCase
Parameter IDeposit4x.remove_liquidity_one_coin(uint256,int128,uint256)._token_amount (contracts/contracts/interfaces/curve/IDeposit.sol#11) is not in mixedCase
Parameter IDeposit4x.remove_liquidity_one_coin(uint256,int128,uint256)._min_amount (contracts/contracts/interfaces/curve/IDeposit.sol#13) is not in mixedCase
Function IDeposit4x.calc_withdraw_one_coin(uint256,int128) (contracts/contracts/interfaces/curve/IDeposit.sol#16) is not in mixedCase
Parameter IDeposit4x.calc_withdraw_one_coin(uint256,int128)._token_amount (contracts/contracts/interfaces/curve/IDeposit.sol#16) is not in mixedCase
Function IDeposit4x.calc_token_amount(uint256[4],bool) (contracts/contracts/interfaces/curve/IDeposit.sol#18) is not in mixedCase
Parameter IDeposit4x.calc_token_amount(uint256[4],bool)._is_deposit (contracts/contracts/interfaces/curve/IDeposit.sol#18) is not in mixedCase
Function ILiquidityGaugeV2.claim_rewards(address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#15) is not in mixedCase
Function ILiquidityGaugeV2.claim_rewards() (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#17) is not in mixedCase
Function ILiquidityGaugeV2.claimable_tokens(address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#19) is not in mixedCase
Function ILiquidityGaugeV2.claimable_reward(address,address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#21) is not in mixedCase
Function ILiquidityGaugeV2.integrate_fraction(address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#23) is not in mixedCase
Function ILiquidityGaugeV2.user_checkpoint(address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#25) is not in mixedCase
Function ILiquidityGaugeV2.reward_integral(address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#27) is not in mixedCase
Function ILiquidityGaugeV2.reward_integral_for(address,address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#29) is not in mixedCase
Function ILiquidityGaugeV2.lp_token() (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#31) is not in mixedCase
Function ILiquidityGaugeV2.reward_count() (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#33) is not in mixedCase
Function ILiquidityGaugeV2.reward_tokens(uint256) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#35) is not in mixedCase
Function ILiquidityGaugeV3.claimable_reward(address,address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#39) is not in mixedCase
Function ILiquidityGaugeV3.claimable_reward_write(address,address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#41) is not in mixedCase
Function IMetapoolFactory.get_underlying_coins(address) (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#7) is not in mixedCase
Function IMetapoolFactory.get_underlying_decimals(address) (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#9) is not in mixedCase
Function IMetapoolFactory.get_coins(address) (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#11) is not in mixedCase
Function IMetapoolFactory.get_n_coins(address) (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#13) is not in mixedCase
Function IMetapoolFactory.get_meta_n_coins(address) (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#15) is not in mixedCase
Function IMetapoolFactory.get_decimals(address) (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#17) is not in mixedCase
Function IMetapoolFactory.get_gauge(address) (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#19) is not in mixedCase
Function IMetapoolFactory.is_meta(address) (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#21) is not in mixedCase
Function IRegistry.get_pool_from_lp_token() (contracts/contracts/interfaces/curve/IRegistry.sol#6) is not in mixedCase
Function IRegistry.get_lp_token(address) (contracts/contracts/interfaces/curve/IRegistry.sol#8) is not in mixedCase
Function IRegistry.get_n_coins(address) (contracts/contracts/interfaces/curve/IRegistry.sol#10) is not in mixedCase
Function IRegistry.get_underlying_coins(address) (contracts/contracts/interfaces/curve/IRegistry.sol#12) is not in mixedCase
Function IRegistry.get_underlying_decimals(address) (contracts/contracts/interfaces/curve/IRegistry.sol#14) is not in mixedCase
Function IRegistry.get_gauges(address) (contracts/contracts/interfaces/curve/IRegistry.sol#16) is not in mixedCase
Function IStableSwap.lp_token() (contracts/contracts/interfaces/curve/IStableSwap.sol#11) is not in mixedCase
Function IStableSwap.A() (contracts/contracts/interfaces/curve/IStableSwap.sol#13) is not in mixedCase
Function IStableSwap.A_precise() (contracts/contracts/interfaces/curve/IStableSwap.sol#15) is not in mixedCase
Function IStableSwap.get_virtual_price() (contracts/contracts/interfaces/curve/IStableSwap.sol#19) is not in mixedCase
Function IStableSwap.get_dy(int128,int128,uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#21-25) is not in mixedCase
Function IStableSwap.get_dy_underlying(int128,int128,uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#27-31) is not in mixedCase
Parameter IStableSwap.exchange(int128,int128,uint256,uint256).min_dy (contracts/contracts/interfaces/curve/IStableSwap.sol#37) is not in mixedCase
Function IStableSwap.calc_withdraw_one_coin(uint256,int128) (contracts/contracts/interfaces/curve/IStableSwap.sol#40) is not in mixedCase
Parameter IStableSwap.calc_withdraw_one_coin(uint256,int128)._token_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#40) is not in mixedCase
Function IStableSwap.remove_liquidity_one_coin(uint256,int128,uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#42-46) is not in mixedCase
Parameter IStableSwap.remove_liquidity_one_coin(uint256,int128,uint256)._token_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#43) is not in mixedCase
Parameter IStableSwap.remove_liquidity_one_coin(uint256,int128,uint256)._min_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#45) is not in mixedCase
Function IStableSwapUnderlying.underlying_coins(uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#54) is not in mixedCase
Function IStableSwapUnderlying.exchange_underlying(int128,int128,uint256,uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#56-61) is not in mixedCase
Parameter IStableSwapUnderlying.exchange_underlying(int128,int128,uint256,uint256).min_dy (contracts/contracts/interfaces/curve/IStableSwap.sol#60) is not in mixedCase
Function IStableSwapUnderlying.remove_liquidity_one_coin(uint256,int128,uint256,bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#63-68) is not in mixedCase
Parameter IStableSwapUnderlying.remove_liquidity_one_coin(uint256,int128,uint256,bool)._token_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#64) is not in mixedCase
Parameter IStableSwapUnderlying.remove_liquidity_one_coin(uint256,int128,uint256,bool)._min_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#66) is not in mixedCase
Parameter IStableSwapUnderlying.remove_liquidity_one_coin(uint256,int128,uint256,bool)._use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#67) is not in mixedCase
Function IStableSwap2x.calc_token_amount(uint256[2],bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#72) is not in mixedCase
Parameter IStableSwap2x.calc_token_amount(uint256[2],bool)._is_deposit (contracts/contracts/interfaces/curve/IStableSwap.sol#72) is not in mixedCase
Function IStableSwap2x.add_liquidity(uint256[2],uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#74) is not in mixedCase
Parameter IStableSwap2x.add_liquidity(uint256[2],uint256)._min_mint_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#74) is not in mixedCase
Function IStableSwap2x.remove_liquidity(uint256,uint256[2]) (contracts/contracts/interfaces/curve/IStableSwap.sol#76) is not in mixedCase
Parameter IStableSwap2x.remove_liquidity(uint256,uint256[2])._min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#76) is not in mixedCase
Function IStableSwap2x.remove_liquidity_imbalance(uint256[2],uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#78-80) is not in mixedCase
Parameter IStableSwap2x.remove_liquidity_imbalance(uint256[2],uint256)._max_burn_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#78) is not in mixedCase
Function IStableSwap3x.calc_token_amount(uint256[3],bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#84) is not in mixedCase
Parameter IStableSwap3x.calc_token_amount(uint256[3],bool)._is_deposit (contracts/contracts/interfaces/curve/IStableSwap.sol#84) is not in mixedCase
Function IStableSwap3x.add_liquidity(uint256[3],uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#86) is not in mixedCase
Parameter IStableSwap3x.add_liquidity(uint256[3],uint256)._min_mint_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#86) is not in mixedCase
Function IStableSwap3x.remove_liquidity(uint256,uint256[3]) (contracts/contracts/interfaces/curve/IStableSwap.sol#88) is not in mixedCase
Parameter IStableSwap3x.remove_liquidity(uint256,uint256[3])._min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#88) is not in mixedCase
Function IStableSwap3x.remove_liquidity_imbalance(uint256[3],uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#90-92) is not in mixedCase
Parameter IStableSwap3x.remove_liquidity_imbalance(uint256[3],uint256)._max_burn_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#90) is not in mixedCase
Function IStableSwap4x.calc_token_amount(uint256[4],bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#96) is not in mixedCase
Parameter IStableSwap4x.calc_token_amount(uint256[4],bool)._is_deposit (contracts/contracts/interfaces/curve/IStableSwap.sol#96) is not in mixedCase
Function IStableSwap4x.add_liquidity(uint256[4],uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#98) is not in mixedCase
Parameter IStableSwap4x.add_liquidity(uint256[4],uint256)._min_mint_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#98) is not in mixedCase
Function IStableSwap4x.remove_liquidity(uint256,uint256[4]) (contracts/contracts/interfaces/curve/IStableSwap.sol#100) is not in mixedCase
Parameter IStableSwap4x.remove_liquidity(uint256,uint256[4])._min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#100) is not in mixedCase
Function IStableSwap4x.remove_liquidity_imbalance(uint256[4],uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#102-104) is not in mixedCase
Parameter IStableSwap4x.remove_liquidity_imbalance(uint256[4],uint256)._max_burn_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#102) is not in mixedCase
Function IStableSwap2xUnderlying.add_liquidity(uint256[2],uint256,bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#108-112) is not in mixedCase
Parameter IStableSwap2xUnderlying.add_liquidity(uint256[2],uint256,bool)._min_mint_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#110) is not in mixedCase
Parameter IStableSwap2xUnderlying.add_liquidity(uint256[2],uint256,bool)._use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#111) is not in mixedCase
Function IStableSwap2xUnderlying.remove_liquidity_imbalance(uint256[2],uint256,bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#114-118) is not in mixedCase
Parameter IStableSwap2xUnderlying.remove_liquidity_imbalance(uint256[2],uint256,bool)._max_burn_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#116) is not in mixedCase
Parameter IStableSwap2xUnderlying.remove_liquidity_imbalance(uint256[2],uint256,bool)._use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#117) is not in mixedCase
Function IStableSwap2xUnderlying.remove_liquidity(uint256,uint256[2],bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#120-124) is not in mixedCase
Parameter IStableSwap2xUnderlying.remove_liquidity(uint256,uint256[2],bool)._min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#122) is not in mixedCase
Parameter IStableSwap2xUnderlying.remove_liquidity(uint256,uint256[2],bool)._use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#123) is not in mixedCase
Function IStableSwap3xUnderlying.add_liquidity(uint256[3],uint256,bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#128-132) is not in mixedCase
Parameter IStableSwap3xUnderlying.add_liquidity(uint256[3],uint256,bool)._min_mint_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#130) is not in mixedCase
Parameter IStableSwap3xUnderlying.add_liquidity(uint256[3],uint256,bool)._use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#131) is not in mixedCase
Function IStableSwap3xUnderlying.remove_liquidity_imbalance(uint256[3],uint256,bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#134-138) is not in mixedCase
Parameter IStableSwap3xUnderlying.remove_liquidity_imbalance(uint256[3],uint256,bool)._max_burn_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#136) is not in mixedCase
Parameter IStableSwap3xUnderlying.remove_liquidity_imbalance(uint256[3],uint256,bool)._use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#137) is not in mixedCase
Function IStableSwap3xUnderlying.remove_liquidity(uint256,uint256[3],bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#140-144) is not in mixedCase
Parameter IStableSwap3xUnderlying.remove_liquidity(uint256,uint256[3],bool)._min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#142) is not in mixedCase
Parameter IStableSwap3xUnderlying.remove_liquidity(uint256,uint256[3],bool)._use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#143) is not in mixedCase
Function IStableSwap4xUnderlying.add_liquidity(uint256[4],uint256,bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#146-152) is not in mixedCase
Parameter IStableSwap4xUnderlying.add_liquidity(uint256[4],uint256,bool)._min_mint_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#150) is not in mixedCase
Parameter IStableSwap4xUnderlying.add_liquidity(uint256[4],uint256,bool)._use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#151) is not in mixedCase
Function IStableSwap4xUnderlying.remove_liquidity_imbalance(uint256[4],uint256,bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#154-158) is not in mixedCase
Parameter IStableSwap4xUnderlying.remove_liquidity_imbalance(uint256[4],uint256,bool)._max_burn_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#156) is not in mixedCase
Parameter IStableSwap4xUnderlying.remove_liquidity_imbalance(uint256[4],uint256,bool)._use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#157) is not in mixedCase
Function IStableSwap4xUnderlying.remove_liquidity(uint256,uint256[4],bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#160-164) is not in mixedCase
Parameter IStableSwap4xUnderlying.remove_liquidity(uint256,uint256[4],bool)._min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#162) is not in mixedCase
Parameter IStableSwap4xUnderlying.remove_liquidity(uint256,uint256[4],bool)._use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#163) is not in mixedCase
Parameter ITokenMinter.mint(address).gauge_addr (contracts/contracts/interfaces/curve/ITokenMinter.sol#9) is not in mixedCase
Parameter Strategy.addKeeper(address)._keeperAddress (contracts/contracts/strategies/Strategy.sol#64) is not in mixedCase
Parameter Strategy.migrate(address)._newStrategy (contracts/contracts/strategies/Strategy.sol#87) is not in mixedCase
Parameter Strategy.removeKeeper(address)._keeperAddress (contracts/contracts/strategies/Strategy.sol#119) is not in mixedCase
Parameter Strategy.sweepERC20(address)._fromToken (contracts/contracts/strategies/Strategy.sol#127) is not in mixedCase
Parameter Strategy.updateFeeCollector(address)._feeCollector (contracts/contracts/strategies/Strategy.sol#156) is not in mixedCase
Parameter Strategy.updateSwapperIfRouterSwapper()._swapper (contracts/contracts/strategies/Strategy.sol#167) is not in mixedCase
Parameter Strategy.withdraw(uint256)._amount (contracts/contracts/strategies/Strategy.sol#178) is not in mixedCase
Variable CurvePoolBase.CRV (contracts/contracts/strategies/curve/CurvePoolBase.sol#37) is not in mixedCase
Variable CurvePoolBase.KEEP_CRV (contracts/contracts/strategies/curve/CurvePoolBase.sol#39) is not in mixedCase
Function IStrategy.VERSION() (vesper-commons/contracts/interfaces/vesper/IStrategy.sol#31) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

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

Variable IStableSwapUnderlying.remove_liquidity_one_coin(uint256,int128,uint256,bool)._min_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#66) is too similar to IStableSwap2xUnderlying.remove_liquidity(uint256,uint256[2],boo
l).min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#122)
Variable IStableSwap.remove_liquidity_one_coin(uint256,int128,uint256)._min_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#45) is too similar to IStableSwap2xUnderlying.remove_liquidity(uint256,uint256[2],bool).min_amounts
(contracts/contracts/interfaces/curve/IStableSwap.sol#122)
Variable IStableSwap.remove_liquidity_one_coin(uint256,int128,uint256)._min_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#45) is too similar to IStableSwap3xUnderlying.remove_liquidity(uint256,uint256[3],bool).min_amounts
(contracts/contracts/interfaces/curve/IStableSwap.sol#142)
Variable IStableSwapUnderlying.remove_liquidity_one_coin(uint256,int128,uint256,bool)._min_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#66) is too similar to IStableSwap3xUnderlying.remove_liquidity(uint256,uint256[3],boo
l).min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#142)
Variable IStableSwap.remove_liquidity_one_coin(uint256,int128,uint256)._min_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#45) is too similar to IStableSwap4xUnderlying.remove_liquidity(uint256,uint256[4],bool).min_amounts
(contracts/contracts/interfaces/curve/IStableSwap.sol#162)
Variable IStableSwapUnderlying.remove_liquidity_one_coin(uint256,int128,uint256,bool)._min_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#66) is too similar to IStableSwap4xUnderlying.remove_liquidity(uint256,uint256[4],boo
l).min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#162)
Variable ConvexBase._getRewardTokens()._rewardToken (contracts/contracts/strategies/convex/ConvexBase.sol#67) is too similar to CurvePoolBase.rewardTokens (contracts/contracts/strategies/curve/CurvePoolBase.sol#45)
Variable Strategy._beforeMigration(address)._newStrategy (contracts/contracts/strategies/Strategy.sol#195) is too similar to CurvePoolBase._beforeMigration(address).newStrategy_ (contracts/contracts/strategies/curve/CurvePoolBase.sol#155
)
Variable IStrategy.migrate(address)._newStrategy (vesper-commons/contracts/interfaces/vesper/IStrategy.sol#24) is too similar to CurvePoolBase._beforeMigration(address).newStrategy_ (contracts/contracts/strategies/curve/CurvePoolBase.sol
#155)
Variable Strategy.migrate(address)._newStrategy (contracts/contracts/strategies/Strategy.sol#87) is too similar to CurvePoolBase._beforeMigration(address).newStrategy_ (contracts/contracts/strategies/curve/CurvePoolBase.sol#155)
Variable CurvePoolBase._claimRewardsAndConvertTo(address)._rewardToken (contracts/contracts/strategies/curve/CurvePoolBase.sol#187) is too similar to CurvePoolBase.rewardTokens (contracts/contracts/strategies/curve/CurvePoolBase.sol#45)
Variable ConvexBase._getRewardTokens()._rewardTokens (contracts/contracts/strategies/convex/ConvexBase.sol#62) is too similar to CurvePoolBase.setRewardTokens(address[])._rewardTokens_ (contracts/contracts/strategies/curve/CurvePoolBase.s
ol#326)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-are-too-similar

Convex4MetaPool.slitherConstructorConstantVariables() (contracts/contracts/strategies/convex/4Pool/Convex4MetaPool.sol#9-83) uses literals with too many digits:
        - ADDRESS_PROVIDER = IAddressProvider(0x0000000022D53366457F9d5E68Ec105046FC4383) (contracts/contracts/strategies/curve/CurvePoolBase.sol#28)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits
```

Curve2LendingPool.sol

```
CurvePoolBase._withdrawHere(uint256) (contracts/contracts/strategies/curve/CurvePoolBase.sol#306-322) uses a dangerous strict equality:
        - _lpToBurn == 0 (contracts/contracts/strategies/curve/CurvePoolBase.sol#314)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dangerous-strict-equalities

CurvePoolBase._claimRewardsAndConvertTo(address).i (contracts/contracts/strategies/curve/CurvePoolBase.sol#186) is a local variable never initialized
CurvePoolBase._approveToken(uint256).i (contracts/contracts/strategies/curve/CurvePoolBase.sol#48) is a local variable never initialized
CurvePoolBase.setRewardTokens(address[]).i (contracts/contracts/strategies/curve/CurvePoolBase.sol#330) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables
```

```
Strategy._swapExactInput(address,address,uint256) (contracts/contracts/strategies/Strategy.sol#210-216) ignores return value by swapper.swapExactInput(_tokenIn,_tokenOut,_amountIn,1,address(this)) (contracts/contracts/strategies/Strategy.sol#215)
Strategy._safeSwapExactInput(address,address,uint256) (contracts/contracts/strategies/Strategy.sol#218-224) ignores return value by swapper.swapExactInput(_tokenIn,_tokenOut,_amountIn,1,address(this)) (contracts/contracts/strategies/Strategy.sol#223)
CurvePoolBase._claimRewardsAndConvertTo(address) (contracts/contracts/strategies/curve/CurvePoolBase.sol#183-196) ignores return value by swapper.swapExactInput(_rewardToken,tokenOut_,_amountIn,1,address(this)) (contracts/contracts/strategies/curve/CurvePoolBase.sol#190-193)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

Address.isContract(address) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#26-35) uses assembly
        - INLINE ASM (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#33)
Address._verifyCallResult(bool,bytes,string) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#171-188) uses assembly
        - INLINE ASM (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#180-183)
EnumerableSet.values(EnumerableSet.AddressSet) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#273-282) uses assembly
        - INLINE ASM (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#277-279)
EnumerableSet.values(EnumerableSet.UintSet) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#346-355) uses assembly
        - INLINE ASM (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#350-352)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Different versions of Solidity is used:
        - Version used: ['0.8.9', '^0.8.0']
        - 0.8.9 (contracts/contracts/interfaces/curve/IAddressProvider.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#4)
        - 0.8.9 (contracts/contracts/interfaces/curve/IRegistry.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/curve/IStableSwap.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/curve/ITokenMinter.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/one-oracle/IMasterOracle.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/swapper/IRoutedSwapper.sol#3)
        - 0.8.9 (contracts/contracts/strategies/Strategy.sol#3)
        - 0.8.9 (contracts/contracts/strategies/curve/2Pool/Curve2LendingPool.sol#3)
        - 0.8.9 (contracts/contracts/strategies/curve/2Pool/Curve2PlainPool.sol#3)
        - 0.8.9 (contracts/contracts/strategies/curve/CurvePoolBase.sol#3)
        - 0.8.9 (vesper-commons/contracts/interfaces/vesper/IStrategy.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/IERC20.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Context.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/Math.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#3)
        - 0.8.9 (vesper-pools/contracts/interfaces/vesper/IGovernable.sol#3)
        - 0.8.9 (vesper-pools/contracts/interfaces/vesper/IPausable.sol#3)
        - 0.8.9 (vesper-pools/contracts/interfaces/vesper/IVesperPool.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

Address.functionCall(address,bytes) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#79-81) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#104-106) is never used and should be removed
Address.functionDelegateCall(address,bytes) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#153-155) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#163-169) is never used and should be removed
Address.functionStaticCall(address,bytes) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#129-131) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#139-145) is never used and should be removed
Context._msgData() (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Context.sol#20-23) is never used and should be removed
CurvePlainPool._depositToCurve(uint256) (contracts/contracts/strategies/curve/2Pool/CurvePlainPool.sol#21-29) is never used and should be removed
CurvePoolBase._depositToCurve(uint256) (contracts/contracts/strategies/curve/CurvePoolBase.sol#203) is never used and should be removed
CurvePoolBase._withdrawFromCurve(uint256,uint256,int128) (contracts/contracts/strategies/curve/CurvePoolBase.sol#290-304) is never used and should be removed
EnumerableSet._at(EnumerableSet.Set,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#129-131) is never used and should be removed
EnumerableSet._length(EnumerableSet.Set) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#115-117) is never used and should be removed
EnumerableSet.add(EnumerableSet.Bytes32Set,bytes32) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#157-159) is never used and should be removed
EnumerableSet.add(EnumerableSet.UintSet,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#296-298) is never used and should be removed
EnumerableSet.at(EnumerableSet.AddressSet,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#261-263) is never used and should be removed
EnumerableSet.at(EnumerableSet.Bytes32Set,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#195-197) is never used and should be removed
EnumerableSet.at(EnumerableSet.UintSet,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#334-336) is never used and should be removed
EnumerableSet.contains(EnumerableSet.Bytes32Set,bytes32) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#174-176) is never used and should be removed
EnumerableSet.contains(EnumerableSet.UintSet,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#313-315) is never used and should be removed
EnumerableSet.length(EnumerableSet.AddressSet) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#247-249) is never used and should be removed
EnumerableSet.length(EnumerableSet.Bytes32Set) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#181-183) is never used and should be removed
EnumerableSet.length(EnumerableSet.UintSet) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#320-322) is never used and should be removed
EnumerableSet.remove(EnumerableSet.Bytes32Set,bytes32) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#167-169) is never used and should be removed
EnumerableSet.remove(EnumerableSet.UintSet,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#306-308) is never used and should be removed
EnumerableSet.values(EnumerableSet.Bytes32Set) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#207-209) is never used and should be removed
EnumerableSet.values(EnumerableSet.UintSet) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#346-355) is never used and should be removed
Math.average(uint256,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/Math.sol#27-30) is never used and should be removed
Math.max(uint256,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/Math.sol#12-14) is never used and should be removed
SafeCast.toInt16(int256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#175-178) is never used and should be removed
SafeCast.toInt256(uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#205-208) is never used and should be removed
SafeCast.toInt32(int256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#157-160) is never used and should be removed
SafeCast.toInt64(int256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#139-142) is never used and should be removed
SafeCast.toInt8(int256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#193-196) is never used and should be removed
SafeCast.toUint128(uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#31-34) is never used and should be removed
SafeCast.toUint16(uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#76-79) is never used and should be removed
SafeCast.toUint256(int256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#103-106) is never used and should be removed
SafeCast.toUint32(uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#61-64) is never used and should be removed
SafeCast.toUint64(uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#46-49) is never used and should be removed
SafeCast.toUint8(uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#91-94) is never used and should be removed
SafeERC20.safeDecreaseAllowance(IERC20,address,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#51-58) is never used and should be removed
SafeERC20.safeIncreaseAllowance(IERC20,address,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#46-49) is never used and should be removed
SafeERC20.safeTransferFrom(IERC20,address,address,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#24-26) is never used and should be removed
Strategy._beforeWithdraw(address) (contracts/contracts/strategies/Strategy.sol#159) is never used and should be removed
Strategy._rebalance() (contracts/contracts/strategies/Strategy.sol#201-208) is never used and should be removed
Strategy._safeSwapExactInput(address,address,uint256) (contracts/contracts/strategies/Strategy.sol#218-224) is never used and should be removed
Strategy._swapExactInput(address,address,uint256) (contracts/contracts/strategies/Strategy.sol#210-216) is never used and should be removed
Strategy._withdrawHere(uint256) (contracts/contracts/strategies/Strategy.sol#227) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version0.8.9 (contracts/contracts/interfaces/curve/IAddressProvider.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#4) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/curve/IRegistry.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/curve/IStableSwap.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/curve/ITokenMinter.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/one-oracle/IMasterOracle.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/swapper/IRoutedSwapper.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/strategies/Strategy.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/strategies/curve/2Pool/Curve2LendingPool.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/strategies/curve/2Pool/Curve2PlainPool.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/strategies/curve/CurvePoolBase.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (vesper-commons/contracts/interfaces/vesper/IStrategy.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/IERC20.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Context.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/Math.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#3) allows old versions
Pragma version0.8.9 (vesper-pools/contracts/interfaces/vesper/IGovernable.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (vesper-pools/contracts/interfaces/vesper/IPausable.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (vesper-pools/contracts/interfaces/vesper/IVesperPool.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
solc-0.8.9 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#53-59):
        - (success) = recipient.call{value: amount}() (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#57)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#114-121):
        - (success,returndata) = target.call{value: value}(data) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#119)

Low level call in Address.functionStaticCall(address,bytes,string) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#139-145):
        - (success,returndata) = target.staticcall(data) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#143)
Low level call in Address.functionDelegateCall(address,bytes,string) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#163-169):
        - (success,returndata) = target.delegatecall(data) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#167)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function IAddressProvider.get_registry() (contracts/contracts/interfaces/curve/IAddressProvider.sol#6) is not in mixedCase
Function IAddressProvider.get_address(uint256) (contracts/contracts/interfaces/curve/IAddressProvider.sol#8) is not in mixedCase
Function ILiquidityGaugeV2.claim_rewards(address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#15) is not in mixedCase
Function ILiquidityGaugeV2.claim_rewards() (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#17) is not in mixedCase
Function ILiquidityGaugeV2.claimable_tokens(address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#19) is not in mixedCase
Function ILiquidityGaugeV2.claimable_reward(address,address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#21) is not in mixedCase
Function ILiquidityGaugeV2.integrate_fraction(address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#23) is not in mixedCase
Function ILiquidityGaugeV2.user_checkpoint(address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#25) is not in mixedCase
Function ILiquidityGaugeV2.reward_integral(address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#27) is not in mixedCase
Function ILiquidityGaugeV2.reward_integral_for(address,address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#29) is not in mixedCase
Function ILiquidityGaugeV2.lp_token() (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#31) is not in mixedCase
Function ILiquidityGaugeV2.reward_count() (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#33) is not in mixedCase
Function ILiquidityGaugeV2.reward_tokens(uint256) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#35) is not in mixedCase
Function ILiquidityGaugeV3.claimable_reward(address,address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#39) is not in mixedCase
Function ILiquidityGaugeV3.claimable_reward_write(address,address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#41) is not in mixedCase
Function IMetapoolFactory.get_underlying_coins(address) (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#7) is not in mixedCase
Function IMetapoolFactory.get_underlying_decimals(address) (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#9) is not in mixedCase
Function IMetapoolFactory.get_coins(address) (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#11) is not in mixedCase
Function IMetapoolFactory.get_n_coins(address) (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#13) is not in mixedCase
Function IMetapoolFactory.get_meta_n_coins(address) (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#15) is not in mixedCase
Function IMetapoolFactory.get_decimals(address) (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#17) is not in mixedCase
Function IMetapoolFactory.get_gauge(address) (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#19) is not in mixedCase
Function IMetapoolFactory.is_meta(address) (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#21) is not in mixedCase
Function IRegistry.get_pool_from_lp_token() (contracts/contracts/interfaces/curve/IRegistry.sol#6) is not in mixedCase
Function IRegistry.get_lp_token(address) (contracts/contracts/interfaces/curve/IRegistry.sol#8) is not in mixedCase
Function IRegistry.get_n_coins(address) (contracts/contracts/interfaces/curve/IRegistry.sol#10) is not in mixedCase
Function IRegistry.get_underlying_coins(address) (contracts/contracts/interfaces/curve/IRegistry.sol#12) is not in mixedCase
Function IRegistry.get_underlying_decimals(address) (contracts/contracts/interfaces/curve/IRegistry.sol#14) is not in mixedCase
Function IRegistry.get_gauges(address) (contracts/contracts/interfaces/curve/IRegistry.sol#16) is not in mixedCase
Function IStableSwap.lp_token() (contracts/contracts/interfaces/curve/IStableSwap.sol#11) is not in mixedCase
Function IStableSwap.A() (contracts/contracts/interfaces/curve/IStableSwap.sol#13) is not in mixedCase
```

```
Function IStableSwap.A_precise() (contracts/contracts/interfaces/curve/IStableSwap.sol#15) is not in mixedCase
Function IStableSwap.get_virtual_price() (contracts/contracts/interfaces/curve/IStableSwap.sol#19) is not in mixedCase
Function IStableSwap.get_dy(int128,int128,uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#21-25) is not in mixedCase
Function IStableSwap.get_dy_underlying(int128,int128,uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#27-31) is not in mixedCase
Function IStableSwap.exchange(int128,int128,uint256,uint256,min_dy (contracts/contracts/interfaces/curve/IStableSwap.sol#37) is not in mixedCase
Function IStableSwap.calc_withdraw_one_coin(uint256,int128) (contracts/contracts/interfaces/curve/IStableSwap.sol#40) is not in mixedCase
Parameter IStableSwap.calc_withdraw_one_coin(uint256,int128)._token_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#40) is not in mixedCase
Function IStableSwap.remove_liquidity_one_coin(uint256,int128,uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#42-46) is not in mixedCase
Parameter IStableSwap.remove_liquidity_one_coin(uint256,int128,uint256)._token_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#43) is not in mixedCase
Parameter IStableSwap.remove_liquidity_one_coin(uint256,int128,uint256)._min_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#45) is not in mixedCase
Function IStableSwapUnderlying.underlying_coins(uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#54) is not in mixedCase
Function IStableSwapUnderlying.exchange_underlying(int128,int128,uint256,uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#56-61) is not in mixedCase
Parameter IStableSwapUnderlying.exchange_underlying(int128,int128,uint256,uint256).min_dy (contracts/contracts/interfaces/curve/IStableSwap.sol#60) is not in mixedCase
Function IStableSwapUnderlying.remove_liquidity_one_coin(uint256,int128,uint256,bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#63-68) is not in mixedCase
Parameter IStableSwapUnderlying.remove_liquidity_one_coin(uint256,int128,uint256,bool)._token_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#64) is not in mixedCase
Parameter IStableSwapUnderlying.remove_liquidity_one_coin(uint256,int128,uint256,bool)._min_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#66) is not in mixedCase
Parameter IStableSwapUnderlying.remove_liquidity_one_coin(uint256,int128,uint256,bool)._use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#67) is not in mixedCase
Function IStableSwap2x.calc_token_amount(uint256[2],bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#72) is not in mixedCase
Parameter IStableSwap2x.calc_token_amount(uint256[2],bool).is_deposit (contracts/contracts/interfaces/curve/IStableSwap.sol#72) is not in mixedCase
Function IStableSwap2x.add_liquidity(uint256[2],uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#74) is not in mixedCase
Parameter IStableSwap2x.add_liquidity(uint256[2],uint256)._min_mint_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#74) is not in mixedCase
Function IStableSwap2x.remove_liquidity(uint256,uint256[2]) (contracts/contracts/interfaces/curve/IStableSwap.sol#76) is not in mixedCase
Parameter IStableSwap2x.remove_liquidity(uint256,uint256[2])._min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#76) is not in mixedCase
Function IStableSwap2x.remove_liquidity_imbalance(uint256[2],uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#78-80) is not in mixedCase
Parameter IStableSwap2x.remove_liquidity_imbalance(uint256[2],uint256)._max_burn_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#78) is not in mixedCase
Function IStableSwap3x.calc_token_amount(uint256[3],bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#84) is not in mixedCase
Parameter IStableSwap3x.calc_token_amount(uint256[3],bool).is_deposit (contracts/contracts/interfaces/curve/IStableSwap.sol#84) is not in mixedCase
Function IStableSwap3x.add_liquidity(uint256[3],uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#86) is not in mixedCase
Parameter IStableSwap3x.add_liquidity(uint256[3],uint256)._min_mint_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#86) is not in mixedCase
Function IStableSwap3x.remove_liquidity(uint256,uint256[3]) (contracts/contracts/interfaces/curve/IStableSwap.sol#88) is not in mixedCase
Parameter IStableSwap3x.remove_liquidity(uint256,uint256[3])._min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#88) is not in mixedCase
Function IStableSwap3x.remove_liquidity_imbalance(uint256[3],uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#90-92) is not in mixedCase
Parameter IStableSwap3x.remove_liquidity_imbalance(uint256[3],uint256)._max_burn_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#90) is not in mixedCase
Function IStableSwap4x.calc_token_amount(uint256[4],bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#96) is not in mixedCase
Parameter IStableSwap4x.calc_token_amount(uint256[4],bool).is_deposit (contracts/contracts/interfaces/curve/IStableSwap.sol#96) is not in mixedCase
Function IStableSwap4x.add_liquidity(uint256[4],uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#98) is not in mixedCase
Parameter IStableSwap4x.add_liquidity(uint256[4],uint256)._min_mint_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#98) is not in mixedCase
Function IStableSwap4x.remove_liquidity(uint256,uint256[4]) (contracts/contracts/interfaces/curve/IStableSwap.sol#100) is not in mixedCase
Parameter IStableSwap4x.remove_liquidity(uint256,uint256[4])._min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#100) is not in mixedCase
Function IStableSwap4x.remove_liquidity_imbalance(uint256[4],uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#102-104) is not in mixedCase
Parameter IStableSwap4x.remove_liquidity_imbalance(uint256[4],uint256)._max_burn_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#102) is not in mixedCase
Function IStableSwap2xUnderlying.add_liquidity(uint256[2],uint256,bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#108-112) is not in mixedCase
Parameter IStableSwap2xUnderlying.add_liquidity(uint256[2],uint256,bool).min_mint_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#110) is not in mixedCase
Parameter IStableSwap2xUnderlying.add_liquidity(uint256[2],uint256,bool).use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#111) is not in mixedCase
Function IStableSwap2xUnderlying.remove_liquidity_imbalance(uint256[2],uint256,bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#114-118) is not in mixedCase
Parameter IStableSwap2xUnderlying.remove_liquidity_imbalance(uint256[2],uint256,bool)._max_burn_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#116) is not in mixedCase
Parameter IStableSwap2xUnderlying.remove_liquidity_imbalance(uint256[2],uint256,bool).use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#117) is not in mixedCase
Function IStableSwap3xUnderlying.remove_liquidity(uint256,uint256[2],bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#120-124) is not in mixedCase
Parameter IStableSwap2xUnderlying.remove_liquidity(uint256,uint256[2],bool).min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#122) is not in mixedCase
Parameter IStableSwap2xUnderlying.remove_liquidity(uint256,uint256[2],bool).use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#123) is not in mixedCase
Function IStableSwap3xUnderlying.add_liquidity(uint256[3],uint256,bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#128-132) is not in mixedCase
Parameter IStableSwap3xUnderlying.add_liquidity(uint256[3],uint256,bool).min_mint_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#130) is not in mixedCase
Parameter IStableSwap3xUnderlying.add_liquidity(uint256[3],uint256,bool).use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#131) is not in mixedCase
Function IStableSwap3xUnderlying.remove_liquidity_imbalance(uint256[3],uint256,bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#134-138) is not in mixedCase
Parameter IStableSwap3xUnderlying.remove_liquidity_imbalance(uint256[3],uint256,bool).max_burn_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#136) is not in mixedCase
Parameter IStableSwap3xUnderlying.remove_liquidity_imbalance(uint256[3],uint256,bool).use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#137) is not in mixedCase
Function IStableSwap3xUnderlying.remove_liquidity(uint256,uint256[3],bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#140-144) is not in mixedCase
Parameter IStableSwap3xUnderlying.remove_liquidity(uint256,uint256[3],bool).min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#142) is not in mixedCase
Parameter IStableSwap3xUnderlying.remove_liquidity(uint256,uint256[3],bool).use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#143) is not in mixedCase
Function IStableSwap4xUnderlying.add_liquidity(uint256[4],uint256,bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#148-152) is not in mixedCase
Parameter IStableSwap4xUnderlying.add_liquidity(uint256[4],uint256,bool).min_mint_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#150) is not in mixedCase
Parameter IStableSwap4xUnderlying.add_liquidity(uint256[4],uint256,bool).use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#151) is not in mixedCase
Function IStableSwap4xUnderlying.remove_liquidity_imbalance(uint256[4],uint256,bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#154-158) is not in mixedCase
Parameter IStableSwap4xUnderlying.remove_liquidity_imbalance(uint256[4],uint256,bool).max_burn_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#156) is not in mixedCase
Parameter IStableSwap4xUnderlying.remove_liquidity_imbalance(uint256[4],uint256,bool).use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#157) is not in mixedCase
Function IStableSwap4xUnderlying.remove_liquidity(uint256,uint256[4],bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#160-164) is not in mixedCase
Parameter IStableSwap4xUnderlying.remove_liquidity(uint256,uint256[4],bool).min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#162) is not in mixedCase
Parameter IStableSwap4xUnderlying.remove_liquidity(uint256,uint256[4],bool).use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#163) is not in mixedCase
Parameter ITokenMinter.mint(address).gauge_addr (contracts/contracts/interfaces/curve/ITokenMinter.sol#9) is not in mixedCase
Parameter Strategy.addKeeper(address)._keeperAddress (contracts/contracts/strategies/Strategy.sol#64) is not in mixedCase
Parameter Strategy.migrate(address)._newStrategy (contracts/contracts/strategies/Strategy.sol#87) is not in mixedCase
Parameter Strategy.removeKeeper(address)._keeperAddress (contracts/contracts/strategies/Strategy.sol#119) is not in mixedCase
Parameter Strategy.sweepERC20(address)._fromToken (contracts/contracts/strategies/Strategy.sol#127) is not in mixedCase
Parameter Strategy.updateFeeCollector(address)._feeCollector (contracts/contracts/strategies/Strategy.sol#156) is not in mixedCase
Parameter Strategy.updateSwapper(IRoutedSwapper)._swapper (contracts/contracts/strategies/Strategy.sol#167) is not in mixedCase
Parameter Strategy.withdraw(uint256)._amount (contracts/contracts/strategies/Strategy.sol#178) is not in mixedCase
Variable CurvePoolBase.CRV (contracts/contracts/strategies/curve/CurvePoolBase.sol#37) is not in mixedCase
Variable CurvePoolBase.NAME (contracts/contracts/strategies/curve/CurvePoolBase.sol#39) is not in mixedCase
Function IStableSwap.VERSION() (vesper-commons/contracts/interfaces/vesper/IStrategy.sol#31) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

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

Variable IStableSwapUnderlying.remove_liquidity_one_coin(uint256,int128,uint256,bool)._min_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#66) is too similar to IStableSwap2xUnderlying.remove_liquidity(uint256,uint256[2],boo
l).min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#122)
Variable IStableSwap.remove_liquidity_one_coin(uint256,int128,uint256)._min_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#45) is too similar to IStableSwap2xUnderlying.remove_liquidity(uint256,uint256[2],bool).min_amounts
(contracts/contracts/interfaces/curve/IStableSwap.sol#122)
Variable IStableSwap.remove_liquidity_one_coin(uint256,int128,uint256)._min_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#45) is too similar to IStableSwap3xUnderlying.remove_liquidity(uint256,uint256[3],bool).min_amounts
(contracts/contracts/interfaces/curve/IStableSwap.sol#142)
Variable IStableSwapUnderlying.remove_liquidity_one_coin(uint256,int128,uint256,bool)._min_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#66) is too similar to IStableSwap3xUnderlying.remove_liquidity(uint256,uint256[3],boo
l).min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#142)
Variable IStableSwap.remove_liquidity_one_coin(uint256,int128,uint256)._min_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#45) is too similar to IStableSwap4xUnderlying.remove_liquidity(uint256,uint256[4],bool).min_amounts
(contracts/contracts/interfaces/curve/IStableSwap.sol#162)
Variable IStableSwapUnderlying.remove_liquidity_one_coin(uint256,int128,uint256,bool)._min_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#66) is too similar to IStableSwap4xUnderlying.remove_liquidity(uint256,uint256[4],boo
l).min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#162)
Variable Strategy.migrate(address)._newStrategy (contracts/contracts/strategies/Strategy.sol#87) is too similar to CurvePoolBase._beforeMigration(address).newStrategy_ (contracts/contracts/strategies/curve/CurvePoolBase.sol#155)
Variable IStrategy.migrate(address)._newStrategy (vesper-commons/contracts/interfaces/vesper/IStrategy.sol#24) is too similar to CurvePoolBase._beforeMigration(address).newStrategy_ (contracts/contracts/strategies/curve/CurvePoolBase.sol
#155)
Variable Strategy._beforeMigration(address)._newStrategy (contracts/contracts/strategies/Strategy.sol#199) is too similar to CurvePoolBase._beforeMigration(address).newStrategy_ (contracts/contracts/strategies/curve/CurvePoolBase.sol#155
)
Variable CurvePoolBase._claimRewardsAndConvertTo(address)._rewardToken (contracts/contracts/strategies/curve/CurvePoolBase.sol#187) is too similar to CurvePoolBase.rewardTokens (contracts/contracts/strategies/curve/CurvePoolBase.sol#45)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-are-too-similar

Curve2LendingPool.slitherConstructorConstantVariables() (contracts/contracts/strategies/curve/2Pool/Curve2LendingPool.sol#9-39) uses literals with too many digits:
        - ADDRESS_PROVIDER = IAddressProvider(0x0000000022D53366457F9d5E68Ec105046FC4383) (contracts/contracts/strategies/curve/CurvePoolBase.sol#28)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits
```

Curve3LendingPoolAave.sol

```
CurvePoolBase._withdrawHere(uint256) (contracts/contracts/strategies/curve/CurvePoolBase.sol#306-322) uses a dangerous strict equality:
        - _lpToBurn == 0 (contracts/contracts/strategies/curve/CurvePoolBase.sol#314)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dangerous-strict-equalities

CurvePoolBase.setRewardTokens(address[]).i (contracts/contracts/strategies/curve/CurvePoolBase.sol#330) is a local variable never initialized
CurvePoolBase._claimRewardsAndConvertTo(address).i (contracts/contracts/strategies/curve/CurvePoolBase.sol#186) is a local variable never initialized
CurvePoolBase._approveToken(uint256).i (contracts/contracts/strategies/curve/CurvePoolBase.sol#148) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

Strategy._swapExactInput(address,address,uint256) (contracts/contracts/strategies/Strategy.sol#210-216) ignores return value by swapper.swapExactInput(_tokenIn,_tokenOut,_amountIn,1,address(this)) (contracts/contracts/strategies/Strategy
.sol#215)
Strategy._safeSwapExactInput(address,address,uint256) (contracts/contracts/strategies/Strategy.sol#218-224) ignores return value by swapper.swapExactInput(_tokenIn,_tokenOut,_amountIn,1,address(this)) (contracts/contracts/strategies/Stra
tegy.sol#223)
CurvePoolBase._claimRewardsAndConvertTo(address) (contracts/contracts/strategies/curve/CurvePoolBase.sol#183-196) ignores return value by swapper.swapExactInput(_rewardToken,tokenOut_,_amountIn,1,address(this)) (contracts/contracts/strat
egies/curve/CurvePoolBase.sol#190-193)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

Curve3LendingPoolAave._canStartCooldown(uint256,uint256) (contracts/contracts/strategies/curve/3Pool/Curve3LendingPoolAave.sol#53-55) uses timestamp for comparisons
        Dangerous comparisons:
        - STKAAVE.balanceOf(address(this)) > 0 && (cooldownStart_ == 0 || block.timestamp > unstakeEnd_) (contracts/contracts/strategies/curve/3Pool/Curve3LendingPoolAave.sol#54)
Curve3LendingPoolAave._canUnstake(uint256,uint256) (contracts/contracts/strategies/curve/3Pool/Curve3LendingPoolAave.sol#57-59) uses timestamp for comparisons
        Dangerous comparisons:
        - block.timestamp > cooldownEnd_ && block.timestamp <= unstakeEnd_ (contracts/contracts/strategies/curve/3Pool/Curve3LendingPoolAave.sol#58)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

Address.isContract(address) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#26-35) uses assembly
        - INLINE ASM (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#33)
Address._verifyCallResult(bool,bytes,string) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#171-188) uses assembly
        - INLINE ASM (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#180-183)
EnumerableSet.values(EnumerableSet.AddressSet) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#273-282) uses assembly
        - INLINE ASM (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#277-279)
EnumerableSet.values(EnumerableSet.UintSet) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#346-355) uses assembly
        - INLINE ASM (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#350-352)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Different versions of Solidity is used:
        - Version used: ['^0.8.9', '^0.8.0']
        - 0.8.9 (contracts/contracts/interfaces/aave/ILave.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/curve/IAddressProvider.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#4)
        - 0.8.9 (contracts/contracts/interfaces/curve/IRegistry.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/curve/IStableSwap.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/curve/ITokenMinter.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/one-oracle/IMasterOracle.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/swapper/IRoutedSwapper.sol#3)
        - 0.8.9 (contracts/contracts/strategies/Strategy.sol#3)
        - 0.8.9 (contracts/contracts/strategies/curve/3Pool/Curve3LendingPool.sol#3)
        - 0.8.9 (contracts/contracts/strategies/curve/3Pool/Curve3LendingPoolAave.sol#3)
        - 0.8.9 (contracts/contracts/strategies/curve/3Pool/Curve3PlainPool.sol#3)
        - 0.8.9 (contracts/contracts/strategies/curve/CurvePoolBase.sol#3)
        - 0.8.9 (vesper-commons/contracts/interfaces/vesper/IStrategy.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/IERC20.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Context.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/Math.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#3)
        - 0.8.9 (vesper-pools/contracts/interfaces/vesper/IGovernable.sol#3)
        - 0.8.9 (vesper-pools/contracts/interfaces/vesper/IPausable.sol#3)
        - 0.8.9 (vesper-pools/contracts/interfaces/vesper/IVesperPool.sol#3)
```

```
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

Address.functionCall(address,bytes) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#79-81) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#104-106) is never used and should be removed
Address.functionDelegateCall(address,bytes) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#153-155) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#163-169) is never used and should be removed
Address.functionStaticCall(address,bytes) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#129-131) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#139-145) is never used and should be removed
Context._msgData() (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Context.sol#20-23) is never used and should be removed
CurveSPlainPool._depositToCurve(uint256) (contracts/contracts/strategies/curve/SPool/CurveSPlainPool.sol#24-32) is never used and should be removed
CurvePoolBase._depositToCurve(uint256) (contracts/contracts/strategies/curve/CurvePoolBase.sol#203) is never used and should be removed
CurvePoolBase._withdrawFromCurve(uint256,uint256,int128) (contracts/contracts/strategies/curve/CurvePoolBase.sol#296-304) is never used and should be removed
EnumerableSet._at(EnumerableSet.Set,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#129-131) is never used and should be removed
EnumerableSet._length(EnumerableSet.Set) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#115-117) is never used and should be removed
EnumerableSet.add(EnumerableSet.Bytes32Set,bytes32) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#157-159) is never used and should be removed
EnumerableSet.add(EnumerableSet.UintSet,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#296-298) is never used and should be removed
EnumerableSet.at(EnumerableSet.AddressSet,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#261-263) is never used and should be removed
EnumerableSet.at(EnumerableSet.Bytes32Set,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#195-197) is never used and should be removed
EnumerableSet.at(EnumerableSet.UintSet,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#334-336) is never used and should be removed
EnumerableSet.contains(EnumerableSet.Bytes32Set,bytes32) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#174-176) is never used and should be removed
EnumerableSet.contains(EnumerableSet.UintSet,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#313-315) is never used and should be removed
EnumerableSet.length(EnumerableSet.AddressSet) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#247-249) is never used and should be removed
EnumerableSet.length(EnumerableSet.Bytes32Set) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#181-183) is never used and should be removed
EnumerableSet.length(EnumerableSet.UintSet) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#320-322) is never used and should be removed
EnumerableSet.remove(EnumerableSet.Bytes32Set,bytes32) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#167-169) is never used and should be removed
EnumerableSet.remove(EnumerableSet.UintSet,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#306-308) is never used and should be removed
EnumerableSet.values(EnumerableSet.Bytes32Set) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#207-209) is never used and should be removed
EnumerableSet.values(EnumerableSet.UintSet) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#346-355) is never used and should be removed
Math.average(uint256,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/Math.sol#27-30) is never used and should be removed
Math.max(uint256,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/Math.sol#12-14) is never used and should be removed
SafeCast.toInt16(int256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#175-176) is never used and should be removed
SafeCast.toInt256(uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#205-208) is never used and should be removed
SafeCast.toInt32(int256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#157-160) is never used and should be removed
SafeCast.toInt64(int256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#138-142) is never used and should be removed
SafeCast.toInt8(int256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#193-196) is never used and should be removed
SafeCast.toUint128(uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#31-34) is never used and should be removed
SafeCast.toUint16(uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#76-79) is never used and should be removed
SafeCast.toUint256(int256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#103-106) is never used and should be removed
SafeCast.toUint32(uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#61-64) is never used and should be removed
SafeCast.toUint64(uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#46-49) is never used and should be removed
SafeCast.toUint8(uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#91-94) is never used and should be removed
SafeERC20.safeDecreaseAllowance(IERC20,address,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#51-58) is never used and should be removed
SafeERC20.safeDecreaseAllowance(IERC20Permit,address,address,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#46-49) is never used and should be removed
SafeERC20.safeTransferFrom(IERC20,address,address,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#24-26) is never used and should be removed
Strategy._beforeMigration(address) (contracts/contracts/strategies/Strategy.sol#195) is never used and should be removed
Strategy._rebalance() (contracts/contracts/strategies/Strategy.sol#201-208) is never used and should be removed
Strategy._safeSwapExactInput(address,address,uint256) (contracts/contracts/strategies/Strategy.sol#218-224) is never used and should be removed
Strategy._swapExactInput(address,address,uint256) (contracts/contracts/strategies/Strategy.sol#210-216) is never used and should be removed
Strategy._withdrawHere(uint256) (contracts/contracts/strategies/Strategy.sol#227) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version0.8.9 (contracts/contracts/interfaces/save/IAave.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/curve/IAddressProvider.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#4) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/curve/IRegistry.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/curve/IStableSwap.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/curve/ITokenMinter.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/one-oracle/IBaseOracle.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/swapper/IRoutedSwapper.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/strategies/Strategy.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/strategies/curve/SPool/CurveLendingPool.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/strategies/curve/SPool/CurveSLendingPoolAave.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/strategies/curve/SPool/CurveSPlainPool.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/strategies/curve/CurvePoolBase.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (vesper-commons/contracts/interfaces/vesper/IStrategy.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/IERC20.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Context.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/Math.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#3) allows old versions
Pragma version0.8.9 (vesper-pools/contracts/interfaces/vesper/IGovernable.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (vesper-pools/contracts/interfaces/vesper/IPausable.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (vesper-pools/contracts/interfaces/vesper/IVesperPool.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
solc-0.8.9 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

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

Function AToken.UNDERLYING_ASSET_ADDRESS() (contracts/contracts/interfaces/save/IAave.sol#45) is not in mixedCase
Function StakedAave.COOLDOWN_SECONDS() (contracts/contracts/interfaces/save/IAave.sol#180) is not in mixedCase
Function StakedAave.UNSTAKE_WINDOW() (contracts/contracts/interfaces/save/IAave.sol#182) is not in mixedCase
Function IAddressProvider.get_registry() (contracts/contracts/interfaces/curve/IAddressProvider.sol#6) is not in mixedCase
Function IAddressProvider.get_address(uint256) (contracts/contracts/interfaces/curve/IAddressProvider.sol#8) is not in mixedCase
Function ILiquidityGaugeV2.claim_rewards(address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#15) is not in mixedCase
Function ILiquidityGaugeV2.claim_rewards() (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#17) is not in mixedCase
Function ILiquidityGaugeV2.claimable_tokens(address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#19) is not in mixedCase
Function ILiquidityGaugeV2.claimable_reward(address,address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#21) is not in mixedCase
Function ILiquidityGaugeV2.integrate_fraction(address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#23) is not in mixedCase
Function ILiquidityGaugeV2.user_checkpoint(address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#25) is not in mixedCase
Function ILiquidityGaugeV2.reward_integral(address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#27) is not in mixedCase
Function ILiquidityGaugeV2.reward_integral_for(address,address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#29) is not in mixedCase
Function ILiquidityGaugeV2.lp_token() (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#31) is not in mixedCase
Function ILiquidityGaugeV2.reward_count() (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#33) is not in mixedCase
Function ILiquidityGaugeV2.reward_tokens(uint256) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#35) is not in mixedCase
Function ILiquidityGaugeV3.claimable_reward(address,address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#39) is not in mixedCase
Function ILiquidityGaugeV3.claimable_reward_write(address,address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#41) is not in mixedCase
Function IMetapoolFactory.get_underlying_coins(address) (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#7) is not in mixedCase
Function IMetapoolFactory.get_underlying_decimals(address) (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#9) is not in mixedCase
Function IMetapoolFactory.get_coins(address) (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#11) is not in mixedCase
Function IMetapoolFactory.get_n_coins(address) (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#13) is not in mixedCase
Function IMetapoolFactory.get_meta_n_coins(address) (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#15) is not in mixedCase
Function IMetapoolFactory.get_decimals(address) (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#17) is not in mixedCase
Function IMetapoolFactory.get_gauge(address) (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#19) is not in mixedCase
Function IMetapoolFactory.is_meta(address) (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#21) is not in mixedCase
Function IRegistry.get_pool_from_lp_token() (contracts/contracts/interfaces/curve/IRegistry.sol#6) is not in mixedCase
Function IRegistry.get_lp_tokens(address) (contracts/contracts/interfaces/curve/IRegistry.sol#8) is not in mixedCase
Function IRegistry.get_n_coins(address) (contracts/contracts/interfaces/curve/IRegistry.sol#10) is not in mixedCase
Function IRegistry.get_underlying_coins(address) (contracts/contracts/interfaces/curve/IRegistry.sol#12) is not in mixedCase
Function IRegistry.get_underlying_decimals(address) (contracts/contracts/interfaces/curve/IRegistry.sol#14) is not in mixedCase
Function IRegistry.get_gauges(address) (contracts/contracts/interfaces/curve/IRegistry.sol#16) is not in mixedCase
Function IStableSwap.lp_token() (contracts/contracts/interfaces/curve/IStableSwap.sol#11) is not in mixedCase
Function IStableSwap.A() (contracts/contracts/interfaces/curve/IStableSwap.sol#13) is not in mixedCase
Function IStableSwap.A_precise() (contracts/contracts/interfaces/curve/IStableSwap.sol#15) is not in mixedCase
Function IStableSwap.get_virtual_price() (contracts/contracts/interfaces/curve/IStableSwap.sol#19) is not in mixedCase
Function IStableSwap.get_dy(int128,int128,uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#21-25) is not in mixedCase
Function IStableSwap.get_dy_underlying(int128,int128,uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#27-31) is not in mixedCase
Parameter IStableSwap.exchange(int128,int128,uint256,uint256).min_dy (contracts/contracts/interfaces/curve/IStableSwap.sol#37) is not in mixedCase
Function IStableSwap.calc_withdraw_one_coin(uint256,int128) (contracts/contracts/interfaces/curve/IStableSwap.sol#40) is not in mixedCase
Parameter IStableSwap.calc_withdraw_one_coin(uint256,int128)._token_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#40) is not in mixedCase
Function IStableSwap.remove_liquidity_one_coin(uint256,int128,uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#42-46) is not in mixedCase
Parameter IStableSwap.remove_liquidity_one_coin(uint256,int128,uint256)._token_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#43) is not in mixedCase
Parameter IStableSwap.remove_liquidity_one_coin(uint256,int128,uint256)._min_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#45) is not in mixedCase
Function IStableSwapUnderlying.underlying_coins(uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#54) is not in mixedCase
Function IStableSwapUnderlying.exchange_underlying(int128,int128,uint256,uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#56-61) is not in mixedCase
Parameter IStableSwapUnderlying.exchange_underlying(int128,int128,uint256,uint256).min_dy (contracts/contracts/interfaces/curve/IStableSwap.sol#60) is not in mixedCase
Function IStableSwapUnderlying.remove_liquidity_one_coin(uint256,int128,uint256,bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#63-68) is not in mixedCase
Parameter IStableSwapUnderlying.remove_liquidity_one_coin(uint256,int128,uint256,bool)._token_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#64) is not in mixedCase
Parameter IStableSwapUnderlying.remove_liquidity_one_coin(uint256,int128,uint256,bool)._min_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#66) is not in mixedCase
Parameter IStableSwapUnderlying.remove_liquidity_one_coin(uint256,int128,uint256,bool)._use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#67) is not in mixedCase
Function IStableSwap2x.calc_token_amount(uint256[2],bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#72) is not in mixedCase
Parameter IStableSwap2x.calc_token_amount(uint256[2],bool).is_deposit (contracts/contracts/interfaces/curve/IStableSwap.sol#72) is not in mixedCase
Function IStableSwap2x.add_liquidity(uint256[2],uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#74) is not in mixedCase
Parameter IStableSwap2x.add_liquidity(uint256[2],uint256)._min_mint_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#74) is not in mixedCase
Function IStableSwap2x.remove_liquidity(uint256,uint256[2]) (contracts/contracts/interfaces/curve/IStableSwap.sol#76) is not in mixedCase
Parameter IStableSwap2x.remove_liquidity(uint256,uint256[2])._min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#76) is not in mixedCase
Function IStableSwap2x.remove_liquidity_imbalance(uint256[2],uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#78-80) is not in mixedCase
Parameter IStableSwap2x.remove_liquidity_imbalance(uint256[2],uint256)._max_burn_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#78) is not in mixedCase
Function IStableSwap3x.calc_token_amount(uint256[3],bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#84) is not in mixedCase
Parameter IStableSwap3x.calc_token_amount(uint256[3],bool).is_deposit (contracts/contracts/interfaces/curve/IStableSwap.sol#84) is not in mixedCase
Function IStableSwap3x.add_liquidity(uint256[3],uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#86) is not in mixedCase
Parameter IStableSwap3x.add_liquidity(uint256[3],uint256)._min_mint_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#86) is not in mixedCase
Function IStableSwap3x.remove_liquidity(uint256,uint256[3]) (contracts/contracts/interfaces/curve/IStableSwap.sol#88) is not in mixedCase
Parameter IStableSwap3x.remove_liquidity(uint256,uint256[3])._min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#88) is not in mixedCase
Function IStableSwap3x.remove_liquidity_imbalance(uint256[3],uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#90-92) is not in mixedCase
Parameter IStableSwap3x.remove_liquidity_imbalance(uint256[3],uint256)._max_burn_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#90) is not in mixedCase
Function IStableSwap4x.calc_token_amount(uint256[4],bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#96) is not in mixedCase
Parameter IStableSwap4x.calc_token_amount(uint256[4],bool).is_deposit (contracts/contracts/interfaces/curve/IStableSwap.sol#96) is not in mixedCase
Function IStableSwap4x.add_liquidity(uint256[4],uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#98) is not in mixedCase
Parameter IStableSwap4x.add_liquidity(uint256[4],uint256)._min_mint_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#98) is not in mixedCase
Function IStableSwap4x.remove_liquidity(uint256,uint256[4]) (contracts/contracts/interfaces/curve/IStableSwap.sol#100) is not in mixedCase
Parameter IStableSwap4x.remove_liquidity(uint256,uint256[4])._min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#100) is not in mixedCase
Function IStableSwap4x.remove_liquidity_imbalance(uint256[4],uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#102-104) is not in mixedCase
Parameter IStableSwap4x.remove_liquidity_imbalance(uint256[4],uint256)._max_burn_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#102) is not in mixedCase
Function IStableSwap2xUnderlying.add_liquidity(uint256[2],uint256,bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#108-112) is not in mixedCase
Parameter IStableSwap2xUnderlying.add_liquidity(uint256[2],uint256,bool).min_mint_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#110) is not in mixedCase
Parameter IStableSwap2xUnderlying.add_liquidity(uint256[2],uint256,bool).use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#111) is not in mixedCase
Function IStableSwap2xUnderlying.remove_liquidity_imbalance(uint256[2],uint256,bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#114-118) is not in mixedCase
Parameter IStableSwap2xUnderlying.remove_liquidity_imbalance(uint256[2],uint256,bool).max_burn_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#116) is not in mixedCase
Parameter IStableSwap2xUnderlying.remove_liquidity_imbalance(uint256[2],uint256,bool).use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#117) is not in mixedCase
Function IStableSwap2xUnderlying.remove_liquidity(uint256,uint256[2],bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#120-124) is not in mixedCase
Parameter IStableSwap2xUnderlying.remove_liquidity(uint256,uint256[2],bool).min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#122) is not in mixedCase
Parameter IStableSwap2xUnderlying.remove_liquidity(uint256,uint256[2],bool).use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#123) is not in mixedCase
Function IStableSwap3xUnderlying.add_liquidity(uint256[3],uint256,bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#126-132) is not in mixedCase
```

```
Parameter IStableSwap3xUnderlying.add_liquidity(uint256[3],uint256,bool).min_mint_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#130) is not in mixedCase
Parameter IStableSwap3xUnderlying.add_liquidity(uint256[3],uint256,bool).use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#131) is not in mixedCase
Function IStableSwap3xUnderlying.remove_liquidity_imbalance(uint256[3],uint256,bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#134-138) is not in mixedCase
Parameter IStableSwap3xUnderlying.remove_liquidity_imbalance(uint256[3],uint256,bool).max_burn_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#136) is not in mixedCase
Parameter IStableSwap3xUnderlying.remove_liquidity_imbalance(uint256[3],uint256,bool).use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#137) is not in mixedCase
Function IStableSwap3xUnderlying.remove_liquidity(uint256,uint256[3],bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#140-144) is not in mixedCase
Parameter IStableSwap3xUnderlying.remove_liquidity(uint256,uint256[3],bool).min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#142) is not in mixedCase
Parameter IStableSwap3xUnderlying.remove_liquidity(uint256,uint256[3],bool).use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#143) is not in mixedCase
Function IStableSwap4xUnderlying.add_liquidity(uint256[4],uint256,bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#146-152) is not in mixedCase
Parameter IStableSwap4xUnderlying.add_liquidity(uint256[4],uint256,bool).min_mint_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#150) is not in mixedCase
Parameter IStableSwap4xUnderlying.add_liquidity(uint256[4],uint256,bool).use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#151) is not in mixedCase
Function IStableSwap4xUnderlying.remove_liquidity_imbalance(uint256[4],uint256,bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#154-158) is not in mixedCase
Parameter IStableSwap4xUnderlying.remove_liquidity_imbalance(uint256[4],uint256,bool).max_burn_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#156) is not in mixedCase
Parameter IStableSwap4xUnderlying.remove_liquidity_imbalance(uint256[4],uint256,bool).use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#157) is not in mixedCase
Function IStableSwap4xUnderlying.remove_liquidity(uint256,uint256[4],bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#160-164) is not in mixedCase
Parameter IStableSwap4xUnderlying.remove_liquidity(uint256,uint256[4],bool).min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#162) is not in mixedCase
Parameter IStableSwap4xUnderlying.remove_liquidity(uint256,uint256[4],bool).use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#163) is not in mixedCase
Parameter ITokenMinter.mint(address).gauge_addr (contracts/contracts/interfaces/curve/ITokenMinter.sol#9) is not in mixedCase
Parameter Strategy.addKeeper(address)._keeperAddress (contracts/contracts/strategies/Strategy.sol#64) is not in mixedCase
Parameter Strategy.migrate(address)._newStrategy (contracts/contracts/strategies/Strategy.sol#87) is not in mixedCase
Parameter Strategy.removeKeeper(address)._keeperAddress (contracts/contracts/strategies/Strategy.sol#119) is not in mixedCase
Parameter Strategy.sweepERC20(address)._fromToken (contracts/contracts/strategies/Strategy.sol#127) is not in mixedCase
Parameter Strategy.updateFeeCollector(address)._feeCollector (contracts/contracts/strategies/Strategy.sol#156) is not in mixedCase
Parameter Strategy.updateSwapper(IRoutedSwapper)._swapper (contracts/contracts/strategies/Strategy.sol#167) is not in mixedCase
Parameter Strategy.withdraw(uint256)._amount (contracts/contracts/strategies/Strategy.sol#178) is not in mixedCase
Variable CurvePoolBase.CRV (contracts/contracts/strategies/curve/CurvePoolBase.sol#37) is not in mixedCase
Variable CurvePoolBase.NAME (contracts/contracts/strategies/curve/CurvePoolBase.sol#39) is not in mixedCase
Function IStrategy.VERSION() (vesper-commons/contracts/interfaces/vesper/IStrategy.sol#31) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

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

Variable IStableSwapUnderlying.remove_liquidity_one_coin(uint256,int128,uint256,bool)._min_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#66) is too similar to IStableSwap2xUnderlying.remove_liquidity(uint256,uint256[2],bool).min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#122)
Variable IStableSwap.remove_liquidity_one_coin(uint256,int128,uint256)._min_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#45) is too similar to IStableSwap2xUnderlying.remove_liquidity(uint256,uint256[2],bool).min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#122)
Variable IStableSwapUnderlying.remove_liquidity_one_coin(uint256,int128,uint256,bool)._min_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#66) is too similar to IStableSwap3xUnderlying.remove_liquidity(uint256,uint256[3],bool).min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#142)
Variable IStableSwap.remove_liquidity_one_coin(uint256,int128,uint256)._min_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#45) is too similar to IStableSwap3xUnderlying.remove_liquidity(uint256,uint256[3],bool).min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#142)
Variable IStableSwap.remove_liquidity_one_coin(uint256,int128,uint256)._min_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#45) is too similar to IStableSwap4xUnderlying.remove_liquidity(uint256,uint256[4],bool).min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#162)
Variable IStableSwapUnderlying.remove_liquidity_one_coin(uint256,int128,uint256,bool)._min_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#66) is too similar to IStableSwap4xUnderlying.remove_liquidity(uint256,uint256[4],bool).min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#162)
Variable IStrategy.migrate(address)._newStrategy (vesper-commons/contracts/interfaces/vesper/IStrategy.sol#24) is too similar to CurvePoolBase._beforeMigration(address).newStrategy_ (contracts/contracts/strategies/curve/CurvePoolBase.sol#155)
Variable Strategy._beforeMigration(address)._newStrategy (contracts/contracts/strategies/Strategy.sol#199) is too similar to CurvePoolBase._beforeMigration(address).newStrategy_ (contracts/contracts/strategies/curve/CurvePoolBase.sol#155)
Variable Strategy.migrate(address)._newStrategy (contracts/contracts/strategies/Strategy.sol#87) is too similar to CurvePoolBase._beforeMigration(address).newStrategy_ (contracts/contracts/strategies/curve/CurvePoolBase.sol#155)
Variable CurvePoolBase._claimRewardsAndConvertTo(address)._rewardToken (contracts/contracts/strategies/curve/CurvePoolBase.sol#187) is too similar to CurvePoolBase.rewardTokens (contracts/contracts/strategies/curve/CurvePoolBase.sol#45)
Variable Curve3LendingPoolAave._claimAave()._cooldownEnd (contracts/contracts/strategies/curve/3Pool/Curve3LendingPoolAave.sol#62) is too similar to Curve3LendingPoolAave._canUnstake(uint256,uint256).cooldownEnd_ (contracts/contracts/strategies/curve/3Pool/Curve3LendingPoolAave.sol#57)
Variable Curve3LendingPoolAave.cooldownData()._cooldownEnd (contracts/contracts/strategies/curve/3Pool/Curve3LendingPoolAave.sol#44) is too similar to Curve3LendingPoolAave._canUnstake(uint256,uint256).cooldownEnd_ (contracts/contracts/strategies/curve/3Pool/Curve3LendingPoolAave.sol#57)
Variable Curve3LendingPoolAave.canStartCooldown()._cooldownStart (contracts/contracts/strategies/curve/3Pool/Curve3LendingPoolAave.sol#30) is too similar to Curve3LendingPoolAave._canStartCooldown(uint256,uint256).cooldownStart_ (contracts/contracts/strategies/curve/3Pool/Curve3LendingPoolAave.sol#53)
Variable Curve3LendingPoolAave._claimAave()._cooldownStart (contracts/contracts/strategies/curve/3Pool/Curve3LendingPoolAave.sol#62) is too similar to Curve3LendingPoolAave._canStartCooldown(uint256,uint256).cooldownStart_ (contracts/contracts/strategies/curve/3Pool/Curve3LendingPoolAave.sol#53)
Variable Curve3LendingPoolAave.cooldownData()._cooldownStart (contracts/contracts/strategies/curve/3Pool/Curve3LendingPoolAave.sol#43) is too similar to Curve3LendingPoolAave._canStartCooldown(uint256,uint256).cooldownStart_ (contracts/contracts/strategies/curve/3Pool/Curve3LendingPoolAave.sol#53)
Variable Curve3LendingPoolAave.canStartCooldown()._unstakeEnd (contracts/contracts/strategies/curve/3Pool/Curve3LendingPoolAave.sol#30) is too similar to Curve3LendingPoolAave._canUnstake(uint256,uint256).unstakeEnd_ (contracts/contracts/strategies/curve/3Pool/Curve3LendingPoolAave.sol#57)
Variable Curve3LendingPoolAave.canStartCooldown()._unstakeEnd (contracts/contracts/strategies/curve/3Pool/Curve3LendingPoolAave.sol#30) is too similar to Curve3LendingPoolAave._canStartCooldown(uint256,uint256).unstakeEnd_ (contracts/contracts/strategies/curve/3Pool/Curve3LendingPoolAave.sol#53)
Variable Curve3LendingPoolAave.canUnstake()._cooldownEnd (contracts/contracts/strategies/curve/3Pool/Curve3LendingPoolAave.sol#35) is too similar to Curve3LendingPoolAave._canUnstake(uint256,uint256).cooldownEnd_ (contracts/contracts/strategies/curve/3Pool/Curve3LendingPoolAave.sol#57)
Variable Curve3LendingPoolAave.canUnstake()._unstakeEnd (contracts/contracts/strategies/curve/3Pool/Curve3LendingPoolAave.sol#35) is too similar to Curve3LendingPoolAave._canStartCooldown(uint256,uint256).unstakeEnd_ (contracts/contracts/strategies/curve/3Pool/Curve3LendingPoolAave.sol#53)
Variable Curve3LendingPoolAave.canUnstake()._unstakeEnd (contracts/contracts/strategies/curve/3Pool/Curve3LendingPoolAave.sol#35) is too similar to Curve3LendingPoolAave._canUnstake(uint256,uint256).unstakeEnd_ (contracts/contracts/strategies/curve/3Pool/Curve3LendingPoolAave.sol#57)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-are-too-similar

Curve3LendingPoolAave.slitherConstructorConstantVariables() (contracts/contracts/strategies/curve/3Pool/Curve3LendingPoolAave.sol#14-75) uses literals with too many digits:
        - ADDRESS_PROVIDER = IAddressProvider(0x0000000022D53366457F9d5E68Ec105046FC4383) (contracts/contracts/strategies/curve/CurvePoolBase.sol#28)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits

Curve3LendingPoolAave.AAVE (contracts/contracts/strategies/curve/3Pool/Curve3LendingPoolAave.sol#18) is never used in Curve3LendingPoolAave (contracts/contracts/strategies/curve/3Pool/Curve3LendingPoolAave.sol#14-75)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable
```

Curve4FactoryMetaPool.sol

```
CurvePoolBase._withdrawHere(uint256) (contracts/contracts/strategies/curve/CurvePoolBase.sol#306-322) uses a dangerous strict equality:
        - _lpToburn == 0 (contracts/contracts/strategies/curve/CurvePoolBase.sol#314)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dangerous-strict-equalities

CurvePoolBase.setRewardTokens(address[]).i (contracts/contracts/strategies/curve/CurvePoolBase.sol#330) is a local variable never initialized
CurvePoolBase._approveToken(uint256).i (contracts/contracts/strategies/curve/CurvePoolBase.sol#148) is a local variable never initialized
CurvePoolBase._claimRewardsAndConvertTo(address).i (contracts/contracts/strategies/curve/CurvePoolBase.sol#186) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

Strategy._swapExactInput(address,address,uint256) (contracts/contracts/strategies/Strategy.sol#210-216) ignores return value by swapper.swapExactInput(_tokenIn,_tokenOut,_amountIn,1,address(this)) (contracts/contracts/strategies/Strategy.sol#215)
Strategy._safeSwapExactInput(address,address,uint256) (contracts/contracts/strategies/Strategy.sol#218-224) ignores return value by swapper.swapExactInput(_tokenIn,_tokenOut,_amountIn,1,address(this)) (contracts/contracts/strategies/Strategy.sol#223)
CurvePoolBase._claimRewardsAndConvertTo(address) (contracts/contracts/strategies/curve/CurvePoolBase.sol#183-196) ignores return value by swapper.swapExactInput(_rewardToken,tokenOut_,_amountIn,1,address(this)) (contracts/contracts/strategies/curve/CurvePoolBase.sol#190-193)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

Address.isContract(address) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#26-35) uses assembly
        - INLINE ASM (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#33)
Address._verifyCallResult(bool,bytes,string) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#171-188) uses assembly
        - INLINE ASM (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#180-183)
EnumerableSet.values(EnumerableSet.AddressSet) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#273-282) uses assembly
        - INLINE ASM (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#277-279)
EnumerableSet.values(EnumerableSet.UintSet) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#346-355) uses assembly
        - INLINE ASM (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#350-352)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Different versions of Solidity is used:
        - Version used: ['^0.8.9', '^0.8.0']
        - 0.8.9 (contracts/contracts/interfaces/curve/IAddressProvider.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/curve/IDepositZap.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#4)
        - 0.8.9 (contracts/contracts/interfaces/curve/IRegistry.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/curve/IStableSwap.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/curve/ITokenMinter.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/one-oracle/IMasterOracle.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/swapper/IRoutedSwapper.sol#3)
        - 0.8.9 (contracts/contracts/strategies/Strategy.sol#3)
        - 0.8.9 (contracts/contracts/strategies/curve/4Pool/Curve4FactoryMetaPool.sol#3)
        - 0.8.9 (contracts/contracts/strategies/curve/CurvePoolBase.sol#3)
        - 0.8.9 (vesper-commons/contracts/interfaces/vesper/IStrategy.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/IERC20.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Context.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/Math.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#3)
        - 0.8.9 (vesper-pools/contracts/interfaces/vesper/IGovernable.sol#3)
        - 0.8.9 (vesper-pools/contracts/interfaces/vesper/IPausable.sol#3)
        - 0.8.9 (vesper-pools/contracts/interfaces/vesper/IVesperPool.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

Address.functionCall(address,bytes) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#79-81) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#104-106) is never used and should be removed
Address.functionDelegateCall(address,bytes) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#153-155) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#163-169) is never used and should be removed
Address.functionStaticCall(address,bytes) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#129-131) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#139-145) is never used and should be removed
Context._msgData() (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Context.sol#20-23) is never used and should be removed
CurvePoolBase._depositToCurve(uint256) (contracts/contracts/strategies/curve/CurvePoolBase.sol#203) is never used and should be removed
CurvePoolBase._quoteLpToCoin(uint256,int128) (contracts/contracts/strategies/curve/CurvePoolBase.sol#259-263) is never used and should be removed
CurvePoolBase._withdrawFromCurve(uint256,uint256,int128) (contracts/contracts/strategies/curve/CurvePoolBase.sol#295-304) is never used and should be removed
EnumerableSet._at(EnumerableSet.Set,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#129-131) is never used and should be removed
EnumerableSet._length(EnumerableSet.Set) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#115-117) is never used and should be removed
EnumerableSet.add(EnumerableSet.Bytes32Set,bytes32) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#157-159) is never used and should be removed
EnumerableSet.add(EnumerableSet.UintSet,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#296-298) is never used and should be removed
EnumerableSet.at(EnumerableSet.AddressSet,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#261-263) is never used and should be removed
EnumerableSet.at(EnumerableSet.Bytes32Set,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#195-197) is never used and should be removed
EnumerableSet.at(EnumerableSet.UintSet,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#334-336) is never used and should be removed
EnumerableSet.contains(EnumerableSet.Bytes32Set,bytes32) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#174-176) is never used and should be removed
EnumerableSet.contains(EnumerableSet.UintSet,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#313-315) is never used and should be removed
EnumerableSet.length(EnumerableSet.AddressSet) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#247-249) is never used and should be removed
EnumerableSet.length(EnumerableSet.Bytes32Set) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#181-183) is never used and should be removed
EnumerableSet.length(EnumerableSet.UintSet) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#320-322) is never used and should be removed
EnumerableSet.remove(EnumerableSet.Bytes32Set,bytes32) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#167-169) is never used and should be removed
EnumerableSet.remove(EnumerableSet.UintSet,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#306-308) is never used and should be removed
EnumerableSet.values(EnumerableSet.Bytes32Set) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#207-209) is never used and should be removed
```

```
EnumerableSet.values(EnumerableSet.UintSet) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#346-355) is never used and should be removed
Math.average(uint256,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/Math.sol#27-30) is never used and should be removed
Math.max(uint256,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/Math.sol#12-14) is never used and should be removed
SafeCast.toInt16(int256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#175-178) is never used and should be removed
SafeCast.toInt256(uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#205-208) is never used and should be removed
SafeCast.toInt32(int256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#157-160) is never used and should be removed
SafeCast.toInt64(int256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#139-142) is never used and should be removed
SafeCast.toInt8(int256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#193-196) is never used and should be removed
SafeCast.toUint128(uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#31-34) is never used and should be removed
SafeCast.toUint16(uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#76-79) is never used and should be removed
SafeCast.toUint256(int256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#103-106) is never used and should be removed
SafeCast.toUint32(uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#61-64) is never used and should be removed
SafeCast.toUint64(uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#46-49) is never used and should be removed
SafeCast.toUint8(uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#91-94) is never used and should be removed
SafeERC20.safeDecreaseAllowance(IERC20,address,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#51-58) is never used and should be removed
SafeERC20.safeIncreaseAllowance(IERC20,address,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#46-49) is never used and should be removed
SafeERC20.safeTransferFrom(IERC20,address,address,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#24-26) is never used and should be removed
Strategy._beforeMigration(address) (contracts/contracts/strategies/Strategy.sol#159) is never used and should be removed
Strategy._rebalance() (contracts/contracts/strategies/Strategy.sol#201-208) is never used and should be removed
Strategy._safeSwapExactInput(address,address,uint256) (contracts/contracts/strategies/Strategy.sol#218-224) is never used and should be removed
Strategy._swapExactInput(address,address,uint256) (contracts/contracts/strategies/Strategy.sol#210-216) is never used and should be removed
Strategy._withdrawHere(uint256) (contracts/contracts/strategies/Strategy.sol#227) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version0.8.9 (contracts/contracts/interfaces/curve/IAddressProvider.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/curve/IDepositZap.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#4) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/curve/IRegistry.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/curve/IStableSwap.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/curve/ITokenMinter.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/one-oracle/IMasterOracle.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/swapper/IRoutedSwapper.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/strategies/Strategy.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/strategies/curve/4Pool/Curve4FactoryMetaPool.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/strategies/curve/CurvePoolBase.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (vesper-commons/contracts/interfaces/vesper/IStrategy.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/IERC20.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Context.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/Math.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#3) allows old versions
Pragma version0.8.9 (vesper-pools/contracts/interfaces/vesper/IGovernable.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (vesper-pools/contracts/interfaces/vesper/IPausable.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (vesper-pools/contracts/interfaces/vesper/IVesperPool.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
solc-0.8.9 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

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

Function IAddressProvider.get_registry() (contracts/contracts/interfaces/curve/IAddressProvider.sol#6) is not in mixedCase
Function IAddressProvider.get_address(uint256) (contracts/contracts/interfaces/curve/IAddressProvider.sol#8) is not in mixedCase

Function IDepositZap4x.calc_token_amount(address,uint256[4],bool) (contracts/contracts/interfaces/curve/IDepositZap.sol#6-10) is not in mixedCase
Parameter IDepositZap4x.calc_token_amount(address,uint256[4],bool)._is_deposit (contracts/contracts/interfaces/curve/IDepositZap.sol#9) is not in mixedCase
Function IDepositZap4x.add_liquidity(address,uint256[4],uint256) (contracts/contracts/interfaces/curve/IDepositZap.sol#12-16) is not in mixedCase
Parameter IDepositZap4x.add_liquidity(address,uint256[4],uint256)._min_mint_amount (contracts/contracts/interfaces/curve/IDepositZap.sol#15) is not in mixedCase
Function IDepositZap4x.remove_liquidity(address,uint256,uint256[4]) (contracts/contracts/interfaces/curve/IDepositZap.sol#18-22) is not in mixedCase
Parameter IDepositZap4x.remove_liquidity(address,uint256,uint256[4])._min_amounts (contracts/contracts/interfaces/curve/IDepositZap.sol#21) is not in mixedCase
Function IDepositZap4x.remove_liquidity_one_coin(address,uint256,int128,uint256) (contracts/contracts/interfaces/curve/IDepositZap.sol#24-29) is not in mixedCase
Parameter IDepositZap4x.remove_liquidity_one_coin(address,uint256,int128,uint256)._token_amount (contracts/contracts/interfaces/curve/IDepositZap.sol#26) is not in mixedCase
Parameter IDepositZap4x.remove_liquidity_one_coin(address,uint256,int128,uint256)._min_amount (contracts/contracts/interfaces/curve/IDepositZap.sol#28) is not in mixedCase
Function IDepositZap4x.calc_withdraw_one_coin(address,uint256,int128) (contracts/contracts/interfaces/curve/IDepositZap.sol#31-35) is not in mixedCase
Parameter IDepositZap4x.calc_withdraw_one_coin(address,uint256,int128)._token_amount (contracts/contracts/interfaces/curve/IDepositZap.sol#33) is not in mixedCase
Function ILiquidityGaugeV2.claim_rewards(address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#15) is not in mixedCase
Function ILiquidityGaugeV2.claim_rewards() (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#17) is not in mixedCase
Function ILiquidityGaugeV2.claimable_tokens(address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#19) is not in mixedCase
Function ILiquidityGaugeV2.claimable_reward(address,address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#21) is not in mixedCase
Function ILiquidityGaugeV2.integrate_fraction(address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#23) is not in mixedCase
Function ILiquidityGaugeV2.user_checkpoint(address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#25) is not in mixedCase
Function ILiquidityGaugeV2.reward_integral(address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#27) is not in mixedCase
Function ILiquidityGaugeV2.reward_integral_for(address,address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#29) is not in mixedCase
Function ILiquidityGaugeV2.lp_token() (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#31) is not in mixedCase
Function ILiquidityGaugeV2.reward_count() (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#33) is not in mixedCase
Function ILiquidityGaugeV2.reward_tokens(uint256) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#35) is not in mixedCase
Function ILiquidityGaugeV3.claimable_reward(address,address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#39) is not in mixedCase
Function ILiquidityGaugeV3.claimable_reward_write(address,address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#41) is not in mixedCase
Function IMetapoolFactory.get_underlying_coins(address) (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#7) is not in mixedCase
Function IMetapoolFactory.get_underlying_decimals(address) (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#9) is not in mixedCase
Function IMetapoolFactory.get_coins(address) (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#11) is not in mixedCase
Function IMetapoolFactory.get_n_coins(address) (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#13) is not in mixedCase
Function IMetapoolFactory.get_meta_n_coins(address) (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#15) is not in mixedCase
Function IMetapoolFactory.get_decimals(address) (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#17) is not in mixedCase
Function IMetapoolFactory.get_gauge(address) (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#19) is not in mixedCase
Function IMetapoolFactory.is_meta(address) (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#21) is not in mixedCase
Function IRegistry.get_pool_from_lp_token() (contracts/contracts/interfaces/curve/IRegistry.sol#6) is not in mixedCase
Function IRegistry.get_lp_token(address) (contracts/contracts/interfaces/curve/IRegistry.sol#8) is not in mixedCase
Function IRegistry.get_n_coins(address) (contracts/contracts/interfaces/curve/IRegistry.sol#10) is not in mixedCase
Function IRegistry.get_underlying_coins(address) (contracts/contracts/interfaces/curve/IRegistry.sol#12) is not in mixedCase
Function IRegistry.get_underlying_decimals(address) (contracts/contracts/interfaces/curve/IRegistry.sol#14) is not in mixedCase
Function IRegistry.get_gauges(address) (contracts/contracts/interfaces/curve/IRegistry.sol#16) is not in mixedCase
Function IStableSwap.lp_token() (contracts/contracts/interfaces/curve/IStableSwap.sol#11) is not in mixedCase
Function IStableSwap.A() (contracts/contracts/interfaces/curve/IStableSwap.sol#13) is not in mixedCase
Function IStableSwap.A_precise() (contracts/contracts/interfaces/curve/IStableSwap.sol#15) is not in mixedCase
Function IStableSwap.get_virtual_price() (contracts/contracts/interfaces/curve/IStableSwap.sol#19) is not in mixedCase
Function IStableSwap.get_dy(int128,int128,uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#21-25) is not in mixedCase
Function IStableSwap.get_dy_underlying(int128,int128,uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#27-31) is not in mixedCase
Parameter IStableSwap.exchange(int128,int128,uint256,uint256).min_dy (contracts/contracts/interfaces/curve/IStableSwap.sol#37) is not in mixedCase
Function IStableSwap.calc_withdraw_one_coin(uint256,int128) (contracts/contracts/interfaces/curve/IStableSwap.sol#40) is not in mixedCase
Parameter IStableSwap.calc_withdraw_one_coin(uint256,int128)._token_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#40) is not in mixedCase
Function IStableSwap.remove_liquidity_one_coin(uint256,int128,uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#42-46) is not in mixedCase
Parameter IStableSwap.remove_liquidity_one_coin(uint256,int128,uint256)._token_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#43) is not in mixedCase
Parameter IStableSwap.remove_liquidity_one_coin(uint256,int128,uint256)._min_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#45) is not in mixedCase

Function IStableSwapUnderlying.underlying_coins(uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#54) is not in mixedCase
Function IStableSwapUnderlying.exchange_underlying(int128,int128,uint256,uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#56-61) is not in mixedCase
Parameter IStableSwapUnderlying.exchange_underlying(int128,int128,uint256,uint256).min_dy (contracts/contracts/interfaces/curve/IStableSwap.sol#60) is not in mixedCase
Function IStableSwapUnderlying.remove_liquidity_one_coin(uint256,int128,uint256,bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#63-68) is not in mixedCase
Parameter IStableSwapUnderlying.remove_liquidity_one_coin(uint256,int128,uint256,bool)._token_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#64) is not in mixedCase
Parameter IStableSwapUnderlying.remove_liquidity_one_coin(uint256,int128,uint256,bool)._min_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#66) is not in mixedCase
Parameter IStableSwapUnderlying.remove_liquidity_one_coin(uint256,int128,uint256,bool)._use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#67) is not in mixedCase
Function IStableSwap2x.calc_token_amount(uint256[2],bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#72) is not in mixedCase
Parameter IStableSwap2x.calc_token_amount(uint256[2],bool)._is_deposit (contracts/contracts/interfaces/curve/IStableSwap.sol#72) is not in mixedCase
Function IStableSwap2x.add_liquidity(uint256[2],uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#74) is not in mixedCase
Parameter IStableSwap2x.add_liquidity(uint256[2],uint256)._min_mint_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#74) is not in mixedCase
Function IStableSwap2x.remove_liquidity(uint256,uint256[2]) (contracts/contracts/interfaces/curve/IStableSwap.sol#76) is not in mixedCase
Parameter IStableSwap2x.remove_liquidity(uint256,uint256[2])._min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#76) is not in mixedCase
Function IStableSwap2x.remove_liquidity_imbalance(uint256[2],uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#78-80) is not in mixedCase
Parameter IStableSwap2x.remove_liquidity_imbalance(uint256[2],uint256)._max_burn_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#78) is not in mixedCase
Function IStableSwap3x.calc_token_amount(uint256[3],bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#84) is not in mixedCase
Parameter IStableSwap3x.calc_token_amount(uint256[3],bool)._is_deposit (contracts/contracts/interfaces/curve/IStableSwap.sol#84) is not in mixedCase
Function IStableSwap3x.add_liquidity(uint256[3],uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#86) is not in mixedCase
Parameter IStableSwap3x.add_liquidity(uint256[3],uint256)._min_mint_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#86) is not in mixedCase
Function IStableSwap3x.remove_liquidity(uint256,uint256[3]) (contracts/contracts/interfaces/curve/IStableSwap.sol#88) is not in mixedCase
Parameter IStableSwap3x.remove_liquidity(uint256,uint256[3])._min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#88) is not in mixedCase
Function IStableSwap3x.remove_liquidity_imbalance(uint256[3],uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#90-92) is not in mixedCase
Parameter IStableSwap3x.remove_liquidity_imbalance(uint256[3],uint256)._max_burn_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#90) is not in mixedCase
Function IStableSwap4x.calc_token_amount(uint256[4],bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#96) is not in mixedCase
Parameter IStableSwap4x.calc_token_amount(uint256[4],bool)._is_deposit (contracts/contracts/interfaces/curve/IStableSwap.sol#96) is not in mixedCase
Function IStableSwap4x.add_liquidity(uint256[4],uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#98) is not in mixedCase
Parameter IStableSwap4x.add_liquidity(uint256[4],uint256)._min_mint_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#98) is not in mixedCase
Function IStableSwap4x.remove_liquidity(uint256,uint256[4]) (contracts/contracts/interfaces/curve/IStableSwap.sol#100) is not in mixedCase
Parameter IStableSwap4x.remove_liquidity(uint256,uint256[4])._min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#100) is not in mixedCase
Function IStableSwap4x.remove_liquidity_imbalance(uint256[4],uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#102-104) is not in mixedCase
Parameter IStableSwap4x.remove_liquidity_imbalance(uint256[4],uint256)._max_burn_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#102) is not in mixedCase
Function IStableSwap2xUnderlying.add_liquidity(uint256[2],uint256,bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#108-112) is not in mixedCase
Function IStableSwap2xUnderlying.add_liquidity(uint256[2],uint256,bool).min_mint_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#110) is not in mixedCase
Parameter IStableSwap2xUnderlying.add_liquidity(uint256[2],uint256,bool)._use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#111) is not in mixedCase
Function IStableSwap2xUnderlying.remove_liquidity_imbalance(uint256[2],uint256,bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#114-118) is not in mixedCase
Parameter IStableSwap2xUnderlying.remove_liquidity_imbalance(uint256[2],uint256,bool).max_burn_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#116) is not in mixedCase
Parameter IStableSwap2xUnderlying.remove_liquidity_imbalance(uint256[2],uint256,bool)._use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#117) is not in mixedCase
Function IStableSwap2xUnderlying.remove_liquidity(uint256,uint256[2],bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#120-124) is not in mixedCase
Parameter IStableSwap2xUnderlying.remove_liquidity(uint256,uint256[2],bool).min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#122) is not in mixedCase
Parameter IStableSwap2xUnderlying.remove_liquidity(uint256,uint256[2],bool)._use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#123) is not in mixedCase
Function IStableSwap3xUnderlying.add_liquidity(uint256[3],uint256,bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#128-132) is not in mixedCase
Parameter IStableSwap3xUnderlying.add_liquidity(uint256[3],uint256,bool).min_mint_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#130) is not in mixedCase
Parameter IStableSwap3xUnderlying.add_liquidity(uint256[3],uint256,bool)._use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#131) is not in mixedCase
Function IStableSwap3xUnderlying.remove_liquidity_imbalance(uint256[3],uint256,bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#134-138) is not in mixedCase
Parameter IStableSwap3xUnderlying.remove_liquidity_imbalance(uint256[3],uint256,bool).max_burn_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#136) is not in mixedCase
Parameter IStableSwap3xUnderlying.remove_liquidity_imbalance(uint256[3],uint256,bool)._use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#137) is not in mixedCase
Function IStableSwap3xUnderlying.remove_liquidity(uint256,uint256[3],bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#140-144) is not in mixedCase
Parameter IStableSwap3xUnderlying.remove_liquidity(uint256,uint256[3],bool).min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#142) is not in mixedCase
Parameter IStableSwap3xUnderlying.remove_liquidity(uint256,uint256[3],bool)._use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#143) is not in mixedCase
Function IStableSwap4xUnderlying.add_liquidity(uint256[4],uint256,bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#148-152) is not in mixedCase
Parameter IStableSwap4xUnderlying.add_liquidity(uint256[4],uint256,bool).min_mint_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#150) is not in mixedCase
Parameter IStableSwap4xUnderlying.add_liquidity(uint256[4],uint256,bool)._use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#151) is not in mixedCase
Function IStableSwap4xUnderlying.remove_liquidity_imbalance(uint256[4],uint256,bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#154-158) is not in mixedCase
Parameter IStableSwap4xUnderlying.remove_liquidity_imbalance(uint256[4],uint256,bool).max_burn_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#156) is not in mixedCase
Parameter IStableSwap4xUnderlying.remove_liquidity_imbalance(uint256[4],uint256,bool)._use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#157) is not in mixedCase
Function IStableSwap4xUnderlying.remove_liquidity(uint256,uint256[4],bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#160-164) is not in mixedCase
Parameter IStableSwap4xUnderlying.remove_liquidity(uint256,uint256[4],bool).min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#162) is not in mixedCase
Parameter IStableSwap4xUnderlying.remove_liquidity(uint256,uint256[4],bool)._use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#163) is not in mixedCase
Parameter ITokenMinter.mint(address)._gauge_addr (contracts/contracts/interfaces/curve/ITokenMinter.sol#9) is not in mixedCase
Parameter Strategy.addKeeper(address)._keeperAddress (contracts/contracts/strategies/Strategy.sol#64) is not in mixedCase
Parameter Strategy.migrate(address)._newStrategy (contracts/contracts/strategies/Strategy.sol#87) is not in mixedCase
Parameter Strategy.removeKeeper(address)._keeperAddress (contracts/contracts/strategies/Strategy.sol#119) is not in mixedCase
Parameter Strategy.sweepERC20(address)._fromToken (contracts/contracts/strategies/Strategy.sol#127) is not in mixedCase
```

```
Parameter Strategy.updateFeeCollector(address)._feeCollector (contracts/contracts/strategies/Strategy.sol#156) is not in mixedCase
Parameter Strategy.updateSwapper(IRoutedSwapper)._swapper (contracts/contracts/strategies/Strategy.sol#167) is not in mixedCase
Parameter Strategy.withdraw(uint256)._amount (contracts/contracts/strategies/Strategy.sol#178) is not in mixedCase
Variable CurvePoolBase.CRV (contracts/contracts/strategies/curve/CurvePoolBase.sol#37) is not in mixedCase
Variable CurvePoolBase.WAHE (contracts/contracts/strategies/curve/CurvePoolBase.sol#39) is not in mixedCase
Function IStrategy.VERSION() (vesper-commons/contracts/interfaces/vesper/IStrategy.sol#31) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

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

Variable IStableSwap.remove_liquidity_one_coin(uint256,int128,uint256)._min_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#45) is too similar to IStableSwap2xUnderlying.remove_liquidity(uint256,uint256[2],bool).min_amounts
(contracts/contracts/interfaces/curve/IStableSwap.sol#122)
Variable IStableSwapUnderlying.remove_liquidity_one_coin(uint256,int128,uint256,bool)._min_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#66) is too similar to IStableSwap2xUnderlying.remove_liquidity(uint256,uint256[2],boo
l).min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#122)
Variable IStableSwap.remove_liquidity_one_coin(uint256,int128,uint256)._min_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#45) is too similar to IStableSwap3xUnderlying.remove_liquidity(uint256,uint256[3],bool).min_amounts
(contracts/contracts/interfaces/curve/IStableSwap.sol#142)
Variable IStableSwapUnderlying.remove_liquidity_one_coin(uint256,int128,uint256,bool)._min_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#66) is too similar to IStableSwap3xUnderlying.remove_liquidity(uint256,uint256[3],boo
l).min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#142)
Variable IStableSwapUnderlying.remove_liquidity_one_coin(uint256,int128,uint256,bool)._min_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#66) is too similar to IStableSwap4xUnderlying.remove_liquidity(uint256,uint256[4],boo
l).min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#162)
Variable IStableSwap.remove_liquidity_one_coin(uint256,int128,uint256)._min_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#45) is too similar to IStableSwap4xUnderlying.remove_liquidity(uint256,uint256[4],bool).min_amounts
(contracts/contracts/interfaces/curve/IStableSwap.sol#162)
Variable CurvePoolBase._claimRewardsAndConvertTo(address)._rewardToken (contracts/contracts/strategies/curve/CurvePoolBase.sol#187) is too similar to CurvePoolBase.rewardTokens (contracts/contracts/strategies/curve/CurvePoolBase.sol#45)
Variable Strategy._beforeMigration(address)._newStrategy (contracts/contracts/strategies/Strategy.sol#199) is too similar to CurvePoolBase._beforeMigration(address).newStrategy_ (contracts/contracts/strategies/curve/CurvePoolBase.sol#155
)
Variable Strategy.migrate(address)._newStrategy (contracts/contracts/strategies/Strategy.sol#87) is too similar to CurvePoolBase._beforeMigration(address).newStrategy_ (contracts/contracts/strategies/curve/CurvePoolBase.sol#155)
Variable IStrategy.migrate(address)._newStrategy (vesper-commons/contracts/interfaces/vesper/IStrategy.sol#24) is too similar to CurvePoolBase._beforeMigration(address).newStrategy_ (contracts/contracts/strategies/curve/CurvePoolBase.sol
#155)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-are-too-similar

Curve4FactoryMetaPool.slitherConstructorConstantVariables() (contracts/contracts/strategies/curve/4Pool/Curve4FactoryMetaPool.sol#10-59) uses literals with too many digits:
        - ADDRESS_PROVIDER = IAddressProvider(0x0000000022D53366457F9d5E68Ec105046FC4383) (contracts/contracts/strategies/curve/CurvePoolBase.sol#28)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits
```

Curve4PlainOr4MetaPool.sol

```
CurvePoolBase._withdrawHere(uint256) (contracts/contracts/strategies/curve/CurvePoolBase.sol#306-322) uses a dangerous strict equality:
        - _lpToBurn == 0 (contracts/contracts/strategies/curve/CurvePoolBase.sol#314)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dangerous-strict-equalities

CurvePoolBase.setRewardTokens(address[]).i (contracts/contracts/strategies/curve/CurvePoolBase.sol#330) is a local variable never initialized
CurvePoolBase._approveToken(uint256).i (contracts/contracts/strategies/curve/CurvePoolBase.sol#148) is a local variable never initialized
CurvePoolBase._claimRewardsAndConvertTo(address).i (contracts/contracts/strategies/curve/CurvePoolBase.sol#186) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

Strategy._swapExactInput(address,address,uint256) (contracts/contracts/strategies/Strategy.sol#210-216) ignores return value by swapper.swapExactInput(_tokenIn,_tokenOut,_amountIn,1,address(this)) (contracts/contracts/strategies/Strategy
.sol#215)
Strategy._safeSwapExactInput(address,address,uint256) (contracts/contracts/strategies/Strategy.sol#218-224) ignores return value by swapper.swapExactInput(_tokenIn,_tokenOut,_amountIn,1,address(this)) (contracts/contracts/strategies/Stra
tegy.sol#223)
CurvePoolBase._claimRewardsAndConvertTo(address) (contracts/contracts/strategies/curve/CurvePoolBase.sol#183-196) ignores return value by swapper.swapExactInput(_rewardToken,tokenOut_,_amountIn,1,address(this)) (contracts/contracts/strat
egies/curve/CurvePoolBase.sol#190-193)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

Address.isContract(address) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#26-35) uses assembly
        - INLINE ASM (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#33)
Address._verifyCallResult(bool,bytes,string) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#171-188) uses assembly
        - INLINE ASM (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#180-183)
EnumerableSet.values(EnumerableSet.AddressSet) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#273-282) uses assembly
        - INLINE ASM (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#277-279)
EnumerableSet.values(EnumerableSet.UintSet) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#346-355) uses assembly
        - INLINE ASM (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#350-352)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Different versions of Solidity is used:
        - Version used: ['0.8.9', '^0.8.0']
        - 0.8.9 (contracts/contracts/interfaces/curve/IAddressProvider.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/curve/IDeposit.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#4)
        - 0.8.9 (contracts/contracts/interfaces/curve/IRegistry.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/curve/IStableSwap.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/curve/ITokenMinter.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/one-oracle/IMasterOracle.sol#3)
        - 0.8.9 (contracts/contracts/interfaces/swapper/IRoutedSwapper.sol#3)
        - 0.8.9 (contracts/contracts/strategies/Strategy.sol#3)
        - 0.8.9 (contracts/contracts/strategies/curve/4Pool/Curve4PlainOr4MetaPool.sol#3)
        - 0.8.9 (contracts/contracts/strategies/curve/CurvePoolBase.sol#3)
        - 0.8.9 (vesper-commons/contracts/interfaces/vesper/IStrategy.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/IERC20.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Context.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/Math.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#3)
        - ^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#3)
        - 0.8.9 (vesper-pools/contracts/interfaces/vesper/IGovernable.sol#3)
        - 0.8.9 (vesper-pools/contracts/interfaces/vesper/IPausable.sol#3)
        - 0.8.9 (vesper-pools/contracts/interfaces/vesper/IVesperPool.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

Address.functionCall(address,bytes) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#79-81) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#104-106) is never used and should be removed
Address.functionDelegateCall(address,bytes) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#153-155) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#163-169) is never used and should be removed
Address.functionStaticCall(address,bytes) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#129-131) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#139-145) is never used and should be removed
Context._msgData() (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Context.sol#20-23) is never used and should be removed
CurvePoolBase._depositToCurve(uint256) (contracts/contracts/strategies/curve/CurvePoolBase.sol#203) is never used and should be removed
CurvePoolBase._quoteLpToCoin(uint256,int128) (contracts/contracts/strategies/curve/CurvePoolBase.sol#259-263) is never used and should be removed
CurvePoolBase._withdrawFromCurve(uint256,uint256,int128) (contracts/contracts/strategies/curve/CurvePoolBase.sol#296-304) is never used and should be removed
EnumerableSet._at(EnumerableSet.Set,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#129-131) is never used and should be removed
EnumerableSet._length(EnumerableSet.Set) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#115-117) is never used and should be removed
EnumerableSet.add(EnumerableSet.Bytes32Set,bytes32) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#157-159) is never used and should be removed
EnumerableSet.add(EnumerableSet.UintSet,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#296-298) is never used and should be removed
EnumerableSet.at(EnumerableSet.AddressSet,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#261-263) is never used and should be removed
EnumerableSet.at(EnumerableSet.Bytes32Set,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#195-197) is never used and should be removed
EnumerableSet.at(EnumerableSet.UintSet,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#334-336) is never used and should be removed
EnumerableSet.contains(EnumerableSet.Bytes32Set,bytes32) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#174-176) is never used and should be removed
EnumerableSet.contains(EnumerableSet.UintSet,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#313-315) is never used and should be removed
EnumerableSet.length(EnumerableSet.AddressSet) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#247-249) is never used and should be removed
EnumerableSet.length(EnumerableSet.Bytes32Set) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#181-183) is never used and should be removed
EnumerableSet.length(EnumerableSet.UintSet) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#320-322) is never used and should be removed
EnumerableSet.remove(EnumerableSet.Bytes32Set,bytes32) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#167-169) is never used and should be removed
EnumerableSet.remove(EnumerableSet.UintSet,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#306-308) is never used and should be removed
EnumerableSet.values(EnumerableSet.Bytes32Set) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#207-209) is never used and should be removed
EnumerableSet.values(EnumerableSet.UintSet) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#346-355) is never used and should be removed
Math.average(uint256,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/Math.sol#27-30) is never used and should be removed
Math.max(uint256,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/Math.sol#12-14) is never used and should be removed
SafeCast.toInt16(int256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#175-178) is never used and should be removed
SafeCast.toInt256(int256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#205-208) is never used and should be removed
SafeCast.toInt32(int256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#157-160) is never used and should be removed
SafeCast.toInt64(int256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#139-142) is never used and should be removed
SafeCast.toInt8(int256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#193-196) is never used and should be removed
SafeCast.toUint128(uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#31-34) is never used and should be removed
SafeCast.toUint16(uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#76-79) is never used and should be removed
SafeCast.toUint256(int256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#103-106) is never used and should be removed
SafeCast.toUint32(uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#61-64) is never used and should be removed
SafeCast.toUint64(uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#46-49) is never used and should be removed
SafeCast.toUint8(uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#91-94) is never used and should be removed
SafeERC20.safeDecreaseAllowance(IERC20,address,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#51-58) is never used and should be removed
SafeERC20.safeIncreaseAllowance(IERC20,address,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#46-49) is never used and should be removed
SafeERC20.safeTransferFrom(IERC20,address,address,uint256) (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#24-26) is never used and should be removed
Strategy._beforeMigration(address) (contracts/contracts/strategies/Strategy.sol#199) is never used and should be removed
Strategy._rebalance() (contracts/contracts/strategies/Strategy.sol#201-208) is never used and should be removed
Strategy._safeSwapExactInput(address,address,uint256) (contracts/contracts/strategies/Strategy.sol#218-224) is never used and should be removed
Strategy._swapExactInput(address,address,uint256) (contracts/contracts/strategies/Strategy.sol#210-216) is never used and should be removed
Strategy._withdrawHere(uint256) (contracts/contracts/strategies/Strategy.sol#227) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version0.8.9 (contracts/contracts/interfaces/curve/IAddressProvider.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/curve/IDeposit.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#4) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/curve/IRegistry.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/curve/IStableSwap.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/curve/ITokenMinter.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/one-oracle/IMasterOracle.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/interfaces/swapper/IRoutedSwapper.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/strategies/Strategy.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/strategies/curve/4Pool/Curve4PlainOr4MetaPool.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (contracts/contracts/strategies/curve/CurvePoolBase.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (vesper-commons/contracts/interfaces/vesper/IStrategy.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/IERC20.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Address.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/Context.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/Math.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/math/SafeCast.sol#3) allows old versions
Pragma version^0.8.0 (vesper-pools/contracts/dependencies/openzeppelin/contracts/utils/structs/EnumerableSet.sol#3) allows old versions
Pragma version0.8.9 (vesper-pools/contracts/interfaces/vesper/IGovernable.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
```

AUTOMATED TESTING

```
Pragma version0.8.9 (vesper-pools/contracts/interfaces/vesper/IPausable.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.9 (vesper-pools/contracts/interfaces/vesper/IVesperPool.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
solc-0.8.9 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

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

Function IAddressProvider.get_registry() (contracts/contracts/interfaces/curve/IAddressProvider.sol#6) is not in mixedCase
Function IAddressProvider.get_address(uint256) (contracts/contracts/interfaces/curve/IAddressProvider.sol#8) is not in mixedCase
Function IDeposit4x.add_liquidity(uint256[4],uint256) (contracts/contracts/interfaces/curve/IDeposit.sol#6) is not in mixedCase
Parameter IDeposit4x.add_liquidity(uint256[4],uint256)._min_mint_amount (contracts/contracts/interfaces/curve/IDeposit.sol#6) is not in mixedCase
Function IDeposit4x.remove_liquidity(uint256,uint256[4]) (contracts/contracts/interfaces/curve/IDeposit.sol#8) is not in mixedCase
Function IDeposit4x.remove_liquidity(uint256,uint256[4])._min_amounts (contracts/contracts/interfaces/curve/IDeposit.sol#8) is not in mixedCase
Function IDeposit4x.remove_liquidity_one_coin(uint256,int128,uint256) (contracts/contracts/interfaces/curve/IDeposit.sol#10-14) is not in mixedCase
Parameter IDeposit4x.remove_liquidity_one_coin(uint256,int128,uint256)._token_amount (contracts/contracts/interfaces/curve/IDeposit.sol#11) is not in mixedCase
Parameter IDeposit4x.remove_liquidity_one_coin(uint256,int128,uint256)._min_amount (contracts/contracts/interfaces/curve/IDeposit.sol#13) is not in mixedCase
Function IDeposit4x.calc_withdraw_one_coin(uint256,int128) (contracts/contracts/interfaces/curve/IDeposit.sol#16) is not in mixedCase
Parameter IDeposit4x.calc_withdraw_one_coin(uint256,int128)._token_amount (contracts/contracts/interfaces/curve/IDeposit.sol#16) is not in mixedCase
Function IDeposit4x.calc_token_amount(uint256[4],bool) (contracts/contracts/interfaces/curve/IDeposit.sol#18) is not in mixedCase
Parameter IDeposit4x.calc_token_amount(uint256[4],bool)._is_deposit (contracts/contracts/interfaces/curve/IDeposit.sol#18) is not in mixedCase
Function ILiquidityGaugeV2.claim_rewards() (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#15) is not in mixedCase
Function ILiquidityGaugeV2.claim_rewards() (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#17) is not in mixedCase
Function ILiquidityGaugeV2.claimable_tokens(address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#19) is not in mixedCase
Function ILiquidityGaugeV2.claimable_reward(address,address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#21) is not in mixedCase
Function ILiquidityGaugeV2.integrate_fraction(address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#23) is not in mixedCase
Function ILiquidityGaugeV2.user_checkpoint(address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#25) is not in mixedCase
Function ILiquidityGaugeV2.reward_integral(address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#27) is not in mixedCase
Function ILiquidityGaugeV2.reward_integral_for(address,address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#29) is not in mixedCase
Function ILiquidityGaugeV2.lp_token() (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#31) is not in mixedCase
Function ILiquidityGaugeV2.reward_count() (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#33) is not in mixedCase
Function ILiquidityGaugeV2.reward_tokens(uint256) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#35) is not in mixedCase
Function ILiquidityGaugeV3.claimable_reward(address,address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#39) is not in mixedCase
Function ILiquidityGaugeV3.claimable_reward_write(address,address) (contracts/contracts/interfaces/curve/ILiquidityGauge.sol#41) is not in mixedCase
Function IMetapoolFactory.get_underlying_coins(address) (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#7) is not in mixedCase
Function IMetapoolFactory.get_underlying_decimals(address) (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#9) is not in mixedCase
Function IMetapoolFactory.get_coins(address) (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#11) is not in mixedCase
Function IMetapoolFactory.get_n_coins(address) (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#13) is not in mixedCase
Function IMetapoolFactory.get_meta_n_coins(address) (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#15) is not in mixedCase
Function IMetapoolFactory.get_decimals(address) (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#17) is not in mixedCase
Function IMetapoolFactory.get_gauge(address) (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#19) is not in mixedCase
Function IMetapoolFactory.is_meta(address) (contracts/contracts/interfaces/curve/IMetapoolFactory.sol#21) is not in mixedCase
Function IRegistry.get_pool_from_lp_token() (contracts/contracts/interfaces/curve/IRegistry.sol#6) is not in mixedCase
Function IRegistry.get_lp_token(address) (contracts/contracts/interfaces/curve/IRegistry.sol#8) is not in mixedCase
Function IRegistry.get_n_coins(address) (contracts/contracts/interfaces/curve/IRegistry.sol#10) is not in mixedCase
Function IRegistry.get_underlying_coins(address) (contracts/contracts/interfaces/curve/IRegistry.sol#12) is not in mixedCase
Function IRegistry.get_underlying_decimals(address) (contracts/contracts/interfaces/curve/IRegistry.sol#14) is not in mixedCase
Function IRegistry.get_gauges(address) (contracts/contracts/interfaces/curve/IRegistry.sol#16) is not in mixedCase
Function IStableSwap.lp_token() (contracts/contracts/interfaces/curve/IStableSwap.sol#11) is not in mixedCase
Function IStableSwap.A() (contracts/contracts/interfaces/curve/IStableSwap.sol#13) is not in mixedCase
Function IStableSwap.A_precise() (contracts/contracts/interfaces/curve/IStableSwap.sol#15) is not in mixedCase
Function IStableSwap.get_virtual_price() (contracts/contracts/interfaces/curve/IStableSwap.sol#19) is not in mixedCase
Function IStableSwap.get_dy(int128,int128,uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#21-25) is not in mixedCase
Function IStableSwap.get_dy_underlying(int128,int128,uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#27-31) is not in mixedCase
Parameter IStableSwap.exchange(int128,int128,uint256,uint256)._min_dy (contracts/contracts/interfaces/curve/IStableSwap.sol#37) is not in mixedCase
Function IStableSwap.calc_withdraw_one_coin(uint256,int128) (contracts/contracts/interfaces/curve/IStableSwap.sol#40) is not in mixedCase
Parameter IStableSwap.calc_withdraw_one_coin(uint256,int128)._token_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#40) is not in mixedCase
Function IStableSwap.remove_liquidity_one_coin(uint256,int128,uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#42-46) is not in mixedCase
Parameter IStableSwap.remove_liquidity_one_coin(uint256,int128,uint256)._token_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#43) is not in mixedCase
Parameter IStableSwap.remove_liquidity_one_coin(uint256,int128,uint256)._min_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#45) is not in mixedCase
Function IStableSwapUnderlying.underlying_coins(uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#54) is not in mixedCase
Function IStableSwapUnderlying.exchange_underlying(int128,int128,uint256,uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#56-61) is not in mixedCase
Parameter IStableSwapUnderlying.exchange_underlying(int128,int128,uint256,uint256)._min_dy (contracts/contracts/interfaces/curve/IStableSwap.sol#60) is not in mixedCase
Function IStableSwapUnderlying.remove_liquidity_one_coin(uint256,int128,uint256,bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#63-68) is not in mixedCase
Parameter IStableSwapUnderlying.remove_liquidity_one_coin(uint256,int128,uint256,bool)._token_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#64) is not in mixedCase
Parameter IStableSwapUnderlying.remove_liquidity_one_coin(uint256,int128,uint256,bool)._min_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#66) is not in mixedCase
Parameter IStableSwapUnderlying.remove_liquidity_one_coin(uint256,int128,uint256,bool)._use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#67) is not in mixedCase
Function IStableSwap2x.calc_token_amount(uint256[2],bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#72) is not in mixedCase
Parameter IStableSwap2x.calc_token_amount(uint256[2],bool)._is_deposit (contracts/contracts/interfaces/curve/IStableSwap.sol#72) is not in mixedCase
Function IStableSwap2x.add_liquidity(uint256[2],uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#74) is not in mixedCase
Parameter IStableSwap2x.add_liquidity(uint256[2],uint256)._min_mint_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#74) is not in mixedCase
Function IStableSwap2x.remove_liquidity(uint256,uint256[2]) (contracts/contracts/interfaces/curve/IStableSwap.sol#76) is not in mixedCase
Parameter IStableSwap2x.remove_liquidity(uint256,uint256[2])._min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#76) is not in mixedCase
Function IStableSwap2x.remove_liquidity_imbalance(uint256[2],uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#78-80) is not in mixedCase
Parameter IStableSwap2x.remove_liquidity_imbalance(uint256[2],uint256)._max_burn_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#78) is not in mixedCase
Function IStableSwap3x.calc_token_amount(uint256[3],bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#84) is not in mixedCase
Parameter IStableSwap3x.calc_token_amount(uint256[3],bool)._is_deposit (contracts/contracts/interfaces/curve/IStableSwap.sol#84) is not in mixedCase
Function IStableSwap3x.add_liquidity(uint256[3],uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#86) is not in mixedCase
Parameter IStableSwap3x.add_liquidity(uint256[3],uint256)._min_mint_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#86) is not in mixedCase
Function IStableSwap3x.remove_liquidity(uint256,uint256[3]) (contracts/contracts/interfaces/curve/IStableSwap.sol#88) is not in mixedCase
Parameter IStableSwap3x.remove_liquidity(uint256,uint256[3])._min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#88) is not in mixedCase
Function IStableSwap3x.remove_liquidity_imbalance(uint256[3],uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#90-92) is not in mixedCase
Parameter IStableSwap3x.remove_liquidity_imbalance(uint256[3],uint256)._max_burn_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#90) is not in mixedCase
Function IStableSwap4x.calc_token_amount(uint256[4],bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#96) is not in mixedCase
Parameter IStableSwap4x.calc_token_amount(uint256[4],bool)._is_deposit (contracts/contracts/interfaces/curve/IStableSwap.sol#96) is not in mixedCase
Function IStableSwap4x.add_liquidity(uint256[4],uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#98) is not in mixedCase
Parameter IStableSwap4x.add_liquidity(uint256[4],uint256)._min_mint_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#98) is not in mixedCase
Function IStableSwap4x.remove_liquidity(uint256,uint256[4]) (contracts/contracts/interfaces/curve/IStableSwap.sol#100) is not in mixedCase
Parameter IStableSwap4x.remove_liquidity(uint256,uint256[4])._min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#100) is not in mixedCase
Function IStableSwap4x.remove_liquidity_imbalance(uint256[4],uint256) (contracts/contracts/interfaces/curve/IStableSwap.sol#102-104) is not in mixedCase
Parameter IStableSwap4x.remove_liquidity_imbalance(uint256[4],uint256)._max_burn_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#102) is not in mixedCase
Function IStableSwap2xUnderlying.add_liquidity(uint256[2],uint256,bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#108-112) is not in mixedCase
Parameter IStableSwap2xUnderlying.add_liquidity(uint256[2],uint256,bool)._min_mint_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#110) is not in mixedCase
Parameter IStableSwap2xUnderlying.add_liquidity(uint256[2],uint256,bool)._use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#111) is not in mixedCase
Function IStableSwap2xUnderlying.remove_liquidity_imbalance(uint256[2],uint256,bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#114-118) is not in mixedCase
Parameter IStableSwap2xUnderlying.remove_liquidity_imbalance(uint256[2],uint256,bool).max_burn_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#116) is not in mixedCase
Parameter IStableSwap2xUnderlying.remove_liquidity_imbalance(uint256[2],uint256,bool)._use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#117) is not in mixedCase
Function IStableSwap2xUnderlying.remove_liquidity(uint256,uint256[2],bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#120-124) is not in mixedCase
Parameter IStableSwap2xUnderlying.remove_liquidity(uint256,uint256[2],bool).min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#122) is not in mixedCase
Parameter IStableSwap2xUnderlying.remove_liquidity(uint256,uint256[2],bool)._use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#123) is not in mixedCase
Function IStableSwap3xUnderlying.add_liquidity(uint256[3],uint256,bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#126-132) is not in mixedCase
Parameter IStableSwap3xUnderlying.add_liquidity(uint256[3],uint256,bool)._min_mint_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#130) is not in mixedCase
Parameter IStableSwap3xUnderlying.add_liquidity(uint256[3],uint256,bool)._use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#131) is not in mixedCase
Function IStableSwap3xUnderlying.remove_liquidity_imbalance(uint256[3],uint256,bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#134-138) is not in mixedCase
Parameter IStableSwap3xUnderlying.remove_liquidity_imbalance(uint256[3],uint256,bool).max_burn_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#136) is not in mixedCase
Parameter IStableSwap3xUnderlying.remove_liquidity_imbalance(uint256[3],uint256,bool)._use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#137) is not in mixedCase
Function IStableSwap3xUnderlying.remove_liquidity(uint256,uint256[3],bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#140-144) is not in mixedCase
Parameter IStableSwap3xUnderlying.remove_liquidity(uint256,uint256[3],bool).min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#142) is not in mixedCase
Parameter IStableSwap3xUnderlying.remove_liquidity(uint256,uint256[3],bool)._use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#143) is not in mixedCase
Function IStableSwap4xUnderlying.add_liquidity(uint256[4],uint256,bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#146-152) is not in mixedCase
Parameter IStableSwap4xUnderlying.add_liquidity(uint256[4],uint256,bool)._min_mint_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#150) is not in mixedCase
Parameter IStableSwap4xUnderlying.add_liquidity(uint256[4],uint256,bool)._use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#151) is not in mixedCase
Function IStableSwap4xUnderlying.remove_liquidity_imbalance(uint256[4],uint256,bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#154-158) is not in mixedCase
Parameter IStableSwap4xUnderlying.remove_liquidity_imbalance(uint256[4],uint256,bool).max_burn_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#156) is not in mixedCase
Parameter IStableSwap4xUnderlying.remove_liquidity_imbalance(uint256[4],uint256,bool)._use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#157) is not in mixedCase
Function IStableSwap4xUnderlying.remove_liquidity(uint256,uint256[4],bool) (contracts/contracts/interfaces/curve/IStableSwap.sol#160-164) is not in mixedCase
Parameter IStableSwap4xUnderlying.remove_liquidity(uint256,uint256[4],bool).min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#162) is not in mixedCase
Parameter IStableSwap4xUnderlying.remove_liquidity(uint256,uint256[4],bool)._use_underlying (contracts/contracts/interfaces/curve/IStableSwap.sol#163) is not in mixedCase
Parameter ITokenMinter.mint(address)._gauge_addr (contracts/contracts/interfaces/curve/ITokenMinter.sol#9) is not in mixedCase
Parameter Strategy.addKeeper(address)._keeperAddress (contracts/contracts/strategies/Strategy.sol#66) is not in mixedCase
Parameter Strategy.migrate(address)._newStrategy (contracts/contracts/strategies/Strategy.sol#87) is not in mixedCase
Parameter Strategy.removeKeeper(address)._keeperAddress (contracts/contracts/strategies/Strategy.sol#119) is not in mixedCase
Parameter Strategy.sweepERC20(address)._fromToken (contracts/contracts/strategies/Strategy.sol#127) is not in mixedCase
Parameter Strategy.updateFeeCollector(address)._feeCollector (contracts/contracts/strategies/Strategy.sol#156) is not in mixedCase
Parameter Strategy.updateSwapper(IRoutedSwapper)._swapper (contracts/contracts/strategies/Strategy.sol#167) is not in mixedCase
Parameter Strategy.withdraw(uint256)._amount (contracts/contracts/strategies/Strategy.sol#178) is not in mixedCase
Variable CurvePoolBase.CRV (contracts/contracts/strategies/curve/CurvePoolBase.sol#37) is not in mixedCase
Variable CurvePoolBase.NAME (contracts/contracts/strategies/curve/CurvePoolBase.sol#39) is not in mixedCase
Function IStrategy.VERSION() (vesper-commons/contracts/interfaces/vesper/IStrategy.sol#31) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

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

Variable IStableSwapUnderlying.remove_liquidity_one_coin(uint256,int128,uint256,bool)._min_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#66) is too similar to IStableSwap2xUnderlying.remove_liquidity(uint256,uint256[2],boo
l).min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#122)
Variable IStableSwap.remove_liquidity_one_coin(uint256,int128,uint256)._min_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#45) is too similar to IStableSwap2xUnderlying.remove_liquidity(uint256,uint256[2],bool).min_amounts
(contracts/contracts/interfaces/curve/IStableSwap.sol#122)
Variable IStableSwap.remove_liquidity_one_coin(uint256,int128,uint256)._min_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#45) is too similar to IStableSwap3xUnderlying.remove_liquidity(uint256,uint256[3],bool).min_amounts
(contracts/contracts/interfaces/curve/IStableSwap.sol#142)
Variable IStableSwapUnderlying.remove_liquidity_one_coin(uint256,int128,uint256,bool)._min_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#66) is too similar to IStableSwap3xUnderlying.remove_liquidity(uint256,uint256[3],boo
l).min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#142)
Variable IStableSwap.remove_liquidity_one_coin(uint256,int128,uint256)._min_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#45) is too similar to IStableSwap4xUnderlying.remove_liquidity(uint256,uint256[4],bool).min_amounts
(contracts/contracts/interfaces/curve/IStableSwap.sol#162)
Variable IStableSwapUnderlying.remove_liquidity_one_coin(uint256,int128,uint256,bool)._min_amount (contracts/contracts/interfaces/curve/IStableSwap.sol#66) is too similar to IStableSwap4xUnderlying.remove_liquidity(uint256,uint256[4],boo
l).min_amounts (contracts/contracts/interfaces/curve/IStableSwap.sol#162)
Variable Strategy._beforeMigration(address)._newStrategy (contracts/contracts/strategies/Strategy.sol#199) is too similar to CurvePoolBase._beforeMigration(address).newStrategy_ (contracts/contracts/strategies/curve/CurvePoolBase.sol#155
)
Variable Strategy.migrate(address)._newStrategy (contracts/contracts/strategies/Strategy.sol#87) is too similar to CurvePoolBase._beforeMigration(address).newStrategy_ (contracts/contracts/strategies/curve/CurvePoolBase.sol#155)
Variable IStrategy.migrate(address)._newStrategy (vesper-commons/contracts/interfaces/vesper/IStrategy.sol#24) is too similar to CurvePoolBase._beforeMigration(address).newStrategy_ (contracts/contracts/strategies/curve/CurvePoolBase.sol
#155)
Variable CurvePoolBase._claimRewardsAndConvertTo(address)._rewardToken (contracts/contracts/strategies/curve/CurvePoolBase.sol#187) is too similar to CurvePoolBase.rewardTokens (contracts/contracts/strategies/curve/CurvePoolBase.sol#45)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-are-too-similar

Curve4PlainOr4MetaPool.slitherConstructorConstantVariables() (contracts/contracts/strategies/curve/4Pool/Curve4PlainOr4MetaPool.sol#13-61) uses literals with too many digits:
        - ADDRESS_PROVIDER = IAddressProvider(0x0000000022D53366457F9d5E68Ec105046FC4383) (contracts/contracts/strategies/curve/CurvePoolBase.sol#28)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits
```

- Some unused returns were correctly flagged by Slither.
- All the reentrancies flagged were checked individually and they are

all false positives.

- No major issues found by Slither.

4.2 AUTOMATED SECURITY SCAN

Description:

Halborn used automated security scanners to assist with detection of well-known security issues and to identify low-hanging fruits on the targets for this engagement. Among the tools used was MythX, a security analysis service for Ethereum smart contracts. MythX performed a scan on the smart contracts and sent the compiled results to the analyzers to locate any vulnerabilities.

MythX results:

VesperMakerStrategy.sol

Report for contracts/contracts/strategies/maker/VesperMakerStrategy.sol
https://dashboard.mythx.io/#/console/analyses/a0470f76-b9d7-452e-8b71-c3e59c69a837

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

AaveV3.sol

Report for contracts/contracts/strategies/aave/v3/AaveV3.sol
https://dashboard.mythx.io/#/console/analyses/2866e90d-3538-45b5-9458-878c60d50d2c

Line	SWC Title	Severity	Short Description
44	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
53	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
54	(SWC-110) Assert Violation	Unknown	Out of bounds array access
67	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
68	(SWC-110) Assert Violation	Unknown	Out of bounds array access
69	(SWC-110) Assert Violation	Unknown	Out of bounds array access
91	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
94	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
96	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
98	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
100	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
106	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered

AaveV3VesperXy.sol

Report for contracts/contracts/strategies/aave/v3/AaveV3VesperXy.sol
https://dashboard.mythx.io/#/console/analyses/4461e6cd-1846-44e5-9a6f-2429f0f026d4

Line	SWC Title	Severity	Short Description
65	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
66	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
66	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
84	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
84	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
85	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
85	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
85	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered

Convex2PlainPool.sol

No issues found by MythX.

Convex3PlainPool.sol

No issues found by MythX.

Convex4FactoryMetaPool.sol

Report for contracts/contracts/strategies/curve/4Pool/Curve4FactoryMetaPool.sol
https://dashboard.mythx.io/#/console/analyses/b16e61de-2e91-4073-8841-0c5f119d05de

Line	SWC Title	Severity	Short Description
37	(SWC-110) Assert Violation	Unknown	Out of bounds array access

Convex4MetaPool.sol

No issues found by MythX.

Curve2LendingPool.sol

Report for contracts/contracts/strategies/curve/2Pool/Curve2LendingPool.sol
https://dashboard.mythx.io/#/console/analyses/c1d0c795-0ed2-4862-8b69-c67484216913

Line	SWC Title	Severity	Short Description
23	(SWC-110) Assert Violation	Unknown	Out of bounds array access

Curve3LendingPoolAave.sol

Report for contracts/contracts/strategies/curve/3Pool/Curve3LendingPoolAave.sol
https://dashboard.mythx.io/#/console/analyses/c14f25e2-09bf-4541-b955-95dba07a92d1

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

Curve4FactoryMetaPool.sol

Report for contracts/contracts/strategies/curve/4Pool/Curve4FactoryMetaPool.sol
https://dashboard.mythx.io/#/console/analyses/3cadbee0-bbbe-4cb6-8659-726e42287191

Line	SWC Title	Severity	Short Description
37	(SWC-110) Assert Violation	Unknown	Out of bounds array access

Curve4PlainOr4MetaPool.sol

Report for contracts/contracts/strategies/curve/4Pool/Curve4PlainOr4MetaPool.sol
https://dashboard.mythx.io/#/console/analyses/2970a34e-7904-4cc8-a3f8-10c729189f9d

Line	SWC Title	Severity	Short Description
41	(SWC-110) Assert Violation	Unknown	Out of bounds array access

- The requirement violations and assert violations are all false positives.
- Integer Overflows and Underflows flagged by MythX are false positives, as those contracts are using Solidity ^0.8.0 version. After the Solidity version 0.8.0 Arithmetic operations revert to underflow and overflow by default.
- No major issues found by MythX.

THANK YOU FOR CHOOSING

// HALBORN