// HALBORN

DRAFT

Vesper Finance

Smart Contract Security Audit - OneOracle

Prepared by: **Halborn**

Date of Engagement: **May 31st, 2022 - June 7th, 2022**

Visit: **Halborn.com**

DOCUMENT REVISION HISTORY

VERSION	MODIFICATION	DATE	AUTHOR
0.1	Document Creation	05/31/2022	Roberto Reigada
0.2	Document Updates	06/07/2022	Roberto Reigada
0.3	Draft Review	06/07/2022	Gabi Urrutia
0.4	Document Updates	06/17/2022	Roberto Reigada

CONTACTS

CONTACT	COMPANY	EMAIL
Rob Behnke	Halborn	Rob.Behnke@halborn.com
Steven Walbroehl	Halborn	Steven.Walbroehl@halborn.com
Gabi Urrutia	Halborn	Gabi.Urrutia@halborn.com
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 May 31st, 2022 and ending on June 7th, 2022. The security assessment was scoped to the smart contracts provided in the GitHub repository bloqpriv/one-oracle.

1.2 AUDIT SUMMARY

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

The purpose of this audit is to:

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

In summary, Halborn identified some security risks that should be addressed by 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

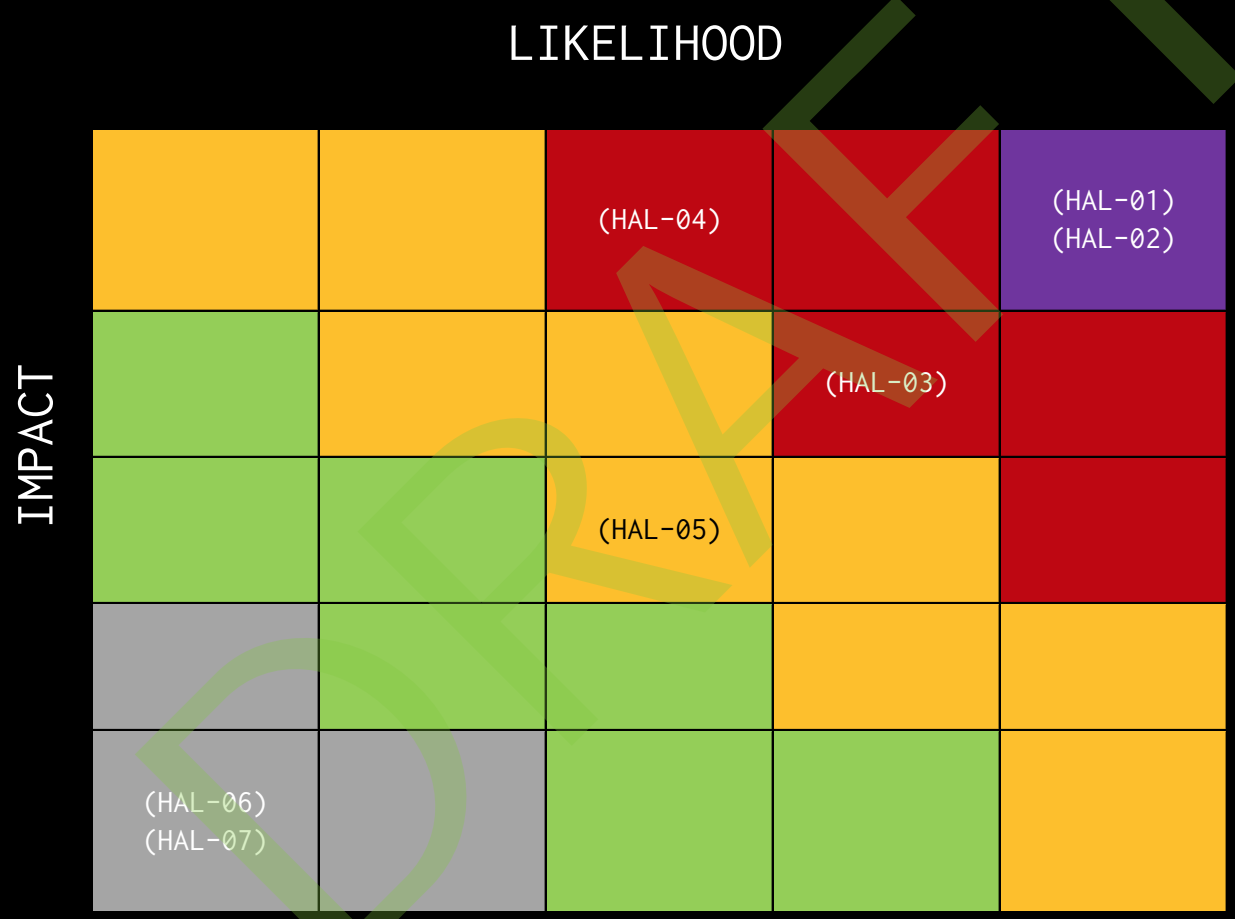
IN-SCOPE:
The security assessment was scoped to the following smart contracts:

- Governable.sol
- ChainlinkArbitrumPriceProvider.sol
- ChainlinkAvalanchePriceProvider.sol
- ChainlinkMainnetPriceProvider.sol
- ChainlinkPolygonPriceProvider.sol
- ChainlinkPriceProvider.sol
- PriceProvidersAggregator.sol
- UniswapV2LikePriceProvider.sol
- UniswapV3PriceProvider.sol
- Oracle.sol
- SwapperOracle.sol
- Swapper.sol
- UniswapV2LikeExchange.sol
- CTokenOracle.sol
- CurveLpTokenOracle.sol
- SynthDefaultOracle.sol
- UniswapV3CrossPoolOracle.sol

Initial Commit ID:
- 84c39f610c30283ea23358f27304ef7783921c44

2. ASSESSMENT SUMMARY & FINDINGS OVERVIEW

CRITICAL	HIGH	MEDIUM	LOW	INFORMATIONAL
2	2	1	0	2



SECURITY ANALYSIS	RISK LEVEL	REMEDIATION DATE
HAL01 - SAFEAPPROVE(0) IS NOT CALLED BEFORE SETTING A NEW APPROVAL	Critical	-
HAL02 - UNISWAPV2LIKEPRICEPROVIDER WILL REVERT WITH OVERFLOWS	Critical	-
HAL03 - SWAPPER.GETBESTAMOUNT FUNCTIONS MAY RETURN INCORRECT MAX AND MIN AMOUNTS	High	-
HAL04 - INCOMPATIBILITY WITH TRANSFER-ON-FEE OR DEFLATIONARY TOKENS	High	-
HAL05 - MAX DEVIATION TOO HIGH IN SWAPPERORACLE	Medium	-
HAL06 - POSSIBLE MISUSE OF PUBLIC FUNCTIONS	Informational	-
HAL07 - STATE VARIABLES MISSING IMMUTABLE MODIFIER	Informational	-

FINDINGS & TECH DETAILS

3.1 (HAL-01) SAFEAPPROVE(0) IS NOT CALLED BEFORE SETTING A NEW APPROVAL - CRITICAL

Description:

In the UniswapV2LikeExchange contract, the functions swapExactInput() and swapExactOutput() do a SafeApprove call before calling swapExactTokensForTokens() in the Router contract:

Listing 1: UniswapV2LikeExchange.sol (Lines 114,128)

```
108 function swapExactInput(
109     address[] calldata path_,
110     uint256 amountIn_,
111     uint256 amountOutMin_,
112     address outReceiver_
113 ) external returns (uint256 _amountOut) {
114     IERC20(path_[0]).safeApprove(address(router), amountIn_);
115     _amountOut = router.swapExactTokensForTokens(amountIn_,
 ↳ amountOutMin_, path_, outReceiver_, block.timestamp)[
116         path_.length - 1
117     ];
118 }
119
120 /// @inheritdoc IExchange
121 function swapExactOutput(
122     address[] calldata path_,
123     uint256 amountOut_,
124     uint256 amountInMax_,
125     address inSender_,
126     address outRecipient_
127 ) external returns (uint256 _amountIn) {
128     IERC20(path_[0]).safeApprove(address(router), amountInMax_);
129     _amountIn = router.swapTokensForExactTokens(amountOut_,
 ↳ amountInMax_, path_, outRecipient_, block.timestamp)[0];
130     // If swap end up costly less than _amountInMax then return
 ↳ remaining
131     uint256 _remainingAmountIn = IERC20(path_[0]).balanceOf(
 ↳ address(this));
132     if (_remainingAmountIn > 0) {
```

```
133         IERC20(path_[0]).safeTransfer(inSender_,
 ↳ _remainingAmountIn);
134     }
135 }
```

SafeERC20.safeApprove() function is used:

Listing 2: SafeERC20.safeApprove (Lines 53-56)

```
45 function safeApprove(
46     IERC20 token,
47     address spender,
48     uint256 value
49 ) internal {
50     // safeApprove should only be called when setting an initial
 ↳ allowance,
51     // or when resetting it to zero. To increase and decrease it,
 ↳ use
52     // 'safeIncreaseAllowance' and 'safeDecreaseAllowance'
53     require(
54         (value == 0) || (token.allowance(address(this), spender)
 ↳ == 0),
55         "SafeERC20: approve from non-zero to non-zero allowance"
56     );
57     _callOptionalReturn(token, abi.encodeWithSelector(token.
 ↳ approve.selector, spender, value));
58 }
```

As per the SafeERC20 contract's comment: safeApprove should only be called when setting an initial allowance, or when resetting it to zero. To increase and decrease it, use safeIncreaseAllowance and safeDecreaseAllowance.

In the line highlighted we can see that SafeERC20.safeApprove() checks that the allowance is zero before setting a new approval.

This means that every second call by a user to Swapper.swapExactOutput() with the same tokenIn_ token will fail with the error: SafeERC20: approve from non-zero to non-zero allowance as the allowance after the first swap will be higher than zero.

Proof of Concept:

In the image below, we can see how the second swap call for DAI as tokenIn_ fails with the error: SafeERC20: approve from non-zero to non-zero allowance:

```
contract_DAI.balanceOf(user1) -> 4000000000000000000000
contract_DAI.balanceOf(contract_Swapper) -> 0
contract_WETH.balanceOf(user1) -> 0
contract_WETH.balanceOf(contract_Swapper) -> 0
Calling -> contract_DAI.approve(contract_Swapper.address, 4000_000000000000000000, {'from': user1})
Transaction sent: 0xd614e5e3890f53fb64fe3c2ce69f32925180a258745db6e122c42e7c56a9fc91
  Gas price: 0.0 gwei   Gas limit: 600000000   Nonce: 0
  DAI.approve confirmed   Block: 14898684   Gas used: 44058 (0.01%)

Calling -> contract_Swapper.swapExactOutput(contract_DAI.address, contract_WETH.address, 1_000000000000000000, user1.address, {'from': user1})
Transaction sent: 0xc8a919c334f4fbb0441f1dce9e8c1487a3857497d1df30dc8dd012e7d46ac600
  Gas price: 0.0 gwei   Gas limit: 600000000   Nonce: 1
  Swapper.swapExactOutput confirmed   Block: 14898685   Gas used: 278067 (0.05%)

contract_DAI.balanceOf(user1) -> 2248157218382782526549
contract_DAI.balanceOf(contract_Swapper) -> 0
contract_WETH.balanceOf(user1) -> 1000000000000000000
contract_WETH.balanceOf(contract_Swapper) -> 0

Calling -> contract_Swapper.swapExactOutput(contract_DAI.address, contract_WETH.address, 1_000000000000000000, user1.address, {'from': user1})
Transaction sent: 0x4282ae95b68eb49d48fa94d6340a123268915e03007771954f0e1773f5a4977b
  Gas price: 0.0 gwei   Gas limit: 600000000   Nonce: 2
  Swapper.swapExactOutput confirmed (SafeERC20: approve from non-zero to non-zero allowance)   Block: 14898686   Gas used: 160976 (0.03%)

contract_DAI.balanceOf(user1) -> 2248157218382782526549
contract_DAI.balanceOf(contract_Swapper) -> 0
contract_WETH.balanceOf(user1) -> 1000000000000000000
contract_WETH.balanceOf(contract_Swapper) -> 0
```

Risk Level:

Likelihood - 5
Impact - 5

Recommendation:

It is recommended to perform 2 different SafeApprove() calls in the swap functions. One resetting the allowance to 0 by calling SafeApprove(0) and another one with the amount needed to be approved:

Listing 3: UniswapV2LikeExchange.sol (Lines 114,115,129,130)

```
108 function swapExactInput(
109     address[] calldata path_,
110     uint256 amountIn_,
111     uint256 amountOutMin_,
112     address outReceiver_
113 ) external returns (uint256 _amountOut) {
114     IERC20(path_[0]).safeApprove(address(router), 0);
115     IERC20(path_[0]).safeApprove(address(router), amountIn_);
```

```
116     _amountOut = router.swapExactTokensForTokens(amountIn_,
 ↳ amountOutMin_, path_, outReceiver_, block.timestamp)[
117         path_.length - 1
118     ];
119 }
120
121 /// @inheritdoc IExchange
122 function swapExactOutput(
123     address[] calldata path_,
124     uint256 amountOut_,
125     uint256 amountInMax_,
126     address inSender_,
127     address outRecipient_
128 ) external returns (uint256 _amountIn) {
129     IERC20(path_[0]).safeApprove(address(router), 0);
130     IERC20(path_[0]).safeApprove(address(router), amountInMax_);
131     _amountIn = router.swapTokensForExactTokens(amountOut_,
 ↳ amountInMax_, path_, outRecipient_, block.timestamp)[0];
132     // If swap end up costly less than _amountInMax then return
 ↳ remaining
133     uint256 _remainingAmountIn = IERC20(path_[0]).balanceOf(
 ↳ address(this));
134     if (_remainingAmountIn > 0) {
135         IERC20(path_[0]).safeTransfer(inSender_,
 ↳ _remainingAmountIn);
136     }
137 }
```

3.2 (HAL-02) UNISWAPV2LIKEPRICEPROVIDER WILL REVERT WITH OVERFLOWS - CRITICAL

Description:

The UniswapV2LikePriceProvider contract makes use of the library UniswapV2OracleLibrary.sol which only works with Solidity < 0.8.0.

According to Uniswap official documentation:
The UniswapV2Pair cumulative price variables are designed to eventually overflow, i.e. price0CumulativeLast and price1CumulativeLast and blockTimestampLast will overflow through 0.

This means that blockTimestamp can be lower than blockTimestampLast and the new price0Cumulative can be less than price0CumulativeLast etc.

In that case, the current implementation of _updateIfNeeded() will revert due to underflow as the UniswapV2LikePriceProvider uses 0.8.9 Solidity version and after the Solidity version 0.8.0 arithmetic operations revert to underflow and overflow by default.

As a result, in some situations price0Average and price1Average will not be updated for a long time and any other contracts or functions that rely on this price will be affected, as the oracle would be displaying unaccurate information.

Code Location:

Listing 4: UniswapV2LikePriceProvider.sol (Lines 199,205,206,207,208,209,210)

```
188 /**
189  * @notice Update an oracle
190  * @param pair_ The pair to update
191  * @param twapPeriod_ The TWAP period
192  * @return True if updated was performed
```

```
193  */
194 function _updateIfNeeded(IUniswapV2Pair pair_, uint256 twapPeriod_
 ↳ ) private returns (bool) {
195     Oracle storage _oracle = oracles[pair_][twapPeriod_];
196 
197     (uint256 price0Cumulative, uint256 price1Cumulative, uint32
 ↳ blockTimestamp) = UniswapV2OracleLibrary
198         .currentCumulativePrices(address(pair_));
199     uint32 timeElapsed = blockTimestamp - _oracle.
 ↳ blockTimestampLast; // overflow is desired
200     // ensure that at least one full period has passed since the
 ↳ last update
201     if (timeElapsed < twapPeriod_) return false;
202 
203     // overflow is desired, casting never truncates
204     // cumulative price is in (uq112x112 price * seconds) units so
 ↳  we simply wrap it after division by time elapsed
205     _oracle.price0Average = FixedPoint.uq112x112(
206         uint224((price0Cumulative - _oracle.price0CumulativeLast)
 ↳ / timeElapsed)
207     );
208     _oracle.price1Average = FixedPoint.uq112x112(
209         uint224((price1Cumulative - _oracle.price1CumulativeLast)
 ↳ / timeElapsed)
210     );
211     _oracle.price0CumulativeLast = price0Cumulative;
212     _oracle.price1CumulativeLast = price1Cumulative;
213     _oracle.blockTimestampLast = blockTimestamp;
214     return true;
215 }
```


Risk Level:

Likelihood - 5
Impact - 5

Recommendation:

It is recommended to update the _updateIfNeeded() function as shown below:

Listing 5: UniswapV2LikePriceProvider.sol (Lines 12,13,14,18,19,20,21)

```
1 /**
2  * @notice Update an oracle
3  * @param pair_ The pair to update
4  * @param twapPeriod_ The TWAP period
5  * @return True if updated was performed
6  */
7 function _updateIfNeeded(IUniswapV2Pair pair_, uint256 twapPeriod_
↳ ) private returns (bool) {
8     Oracle storage _oracle = oracles[pair_][twapPeriod_];
9
10     (uint256 price0Cumulative, uint256 price1Cumulative, uint32
↳ blockTimestamp) = UniswapV2OracleLibrary
11         .currentCumulativePrices(address(pair_));
12     unchecked {
13         uint32 timeElapsed = blockTimestamp - _oracle.
↳ blockTimestampLast; // overflow is desired
14     }
15     // ensure that at least one full period has passed since the
↳ last update
16     if (timeElapsed < twapPeriod_) return false;
17
18     unchecked {
19         uint256 price0new = price0Cumulative - _oracle.
↳ price0CumulativeLast;
20         uint256 price1new = price1Cumulative - _oracle.
↳ price1CumulativeLast;
21     }
22
23     // overflow is desired, casting never truncates
24     // cumulative price is in (uq112x112 price * seconds) units so
↳  we simply wrap it after division by time elapsed
25     _oracle.price0Average = FixedPoint.uq112x112(
26         uint224(price0new / timeElapsed)
27     );
28     _oracle.price1Average = FixedPoint.uq112x112(
29         uint224(price1new / timeElapsed)
30     );
31     _oracle.price0CumulativeLast = price0Cumulative;
32     _oracle.price1CumulativeLast = price1Cumulative;
33     _oracle.blockTimestampLast = blockTimestamp;
34     return true;
35 }
```

3.3 (HAL-03) SWAPPER.GETBESTAMOUNT FUNCTIONS MAY RETURN INCORRECT MAX AND MIN AMOUNTS - HIGH

Description:

In the Swapper contract, the function getBestAmountIn() given an input token, an output token and an output amount, calculates:

- The minimum amount of tokens that should be input by the user to get the amount of output tokens set => _amountIn
- The maximum amount of tokens that should be input by the user to get the amount of output tokens set => _amountInMax
- The exchange address from where the _amountIn data was pulled => _exchange
- The swap path that must be used in the _exchange to get that _amountIn => _path

Listing 6: Swapper.sol (Lines 71,75,89,90)

```
56 function getBestAmountIn(
57     address tokenIn_,
58     address tokenOut_,
59     uint256 amountOut_
60 )
61     public
62     view
63     returns (
64         uint256 _amountIn,
65         uint256 _amountInMax,
66         IExchange _exchange,
67         address[] memory _path
68     )
69 {
70     _exchange = IExchange(exchanges.at(0));
71     (_amountIn, _path) = _getBestAmountIn(_exchange, tokenIn_,
↳ tokenOut_, amountOut_);
72     uint256 _len = exchanges.length();
73     for (uint256 i = 1; i < _len; ++i) {
```

```
74          IExchange _iExchange = IExchange(exchanges.at(i));
75          (uint256 _iAmountIn, address[] memory _iPath) =
↳ _getBestAmountIn(
76              _iExchange,
77              tokenIn_,
78              tokenOut_,
79              amountOut_
80          );
81          if (_iAmountIn > 0 && _iAmountIn < _amountIn) {
82              _amountIn = _iAmountIn;
83              _exchange = _iExchange;
84              _path = _iPath;
85          }
86      }
87      require(_path.length > 0, "no-path-found");
88
89      uint256 _amountInFromOracle = oracle.quote(tokenOut_, tokenIn_
↳ , amountOut_);
90      _amountInMax = (_amountInFromOracle * (1e18 + maxSlippage)) /
↳ 1e18;
91  }
```

In order to retrieve all that data, this function calls the `getBestAmountIn()` function in the Exchange Router Contract:

Listing 7: UniswapV2LikeExchange.sol (Lines 51,64)

```
42  function getBestAmountIn(
43      address tokenIn_,
44      address tokenOut_,
45      uint256 amountOut_
46  ) external view returns (uint256 _amountIn, address[] memory _path
↳ ) {
47      // 1. Check IN-OUT pair
48      address[] memory _pathA = new address[](2);
49      _pathA[0] = tokenIn_;
50      _pathA[1] = tokenOut_;
51      uint256 _amountInA = _getAmountsIn(amountOut_, _pathA);
52
53      if (tokenIn_ == wethLike || tokenOut_ == wethLike) {
54          // Returns if one of the token is WETH-Like
55          require(_amountInA > 0, "invalid-swap");
56          return (_amountInA, _pathA);
```

```
57      }
58
59      // 2. Check IN-WETH-OUT path
60      address[] memory _pathB = new address[](3);
61      _pathB[0] = tokenIn_;
62      _pathB[1] = wethLike;
63      _pathB[2] = tokenOut_;
64      uint256 _amountInB = _getAmountsIn(amountOut_, _pathB);
65
66      // 3. Get best route between paths A and B
67      require(_amountInA > 0 || _amountInB > 0, "invalid-swap");
68
69      // Returns A if it's valid and better than B or if B isn't
↳ valid
70      if ((_amountInA > 0 && _amountInA < _amountInB) || _amountInB
↳ == 0) {
71          return (_amountInA, _pathA);
72      }
73      return (_amountInB, _pathB);
74  }
```

Although occasionally the direct path (tokenIn_ => tokenOut_) will not be available, as it occurs here when trying to swap WBTC into ApeCoins:

```
>>> contract_UNISWAPV2_ROUTER.getAmountsIn(5000_000000000000000000, [contract_WBTC, contract_APECOIN])
  File "<console>", line 1, in <module>
  File "/usr/local/lib/python3.8/dist-packages/brownie/network/contract.py", line 1665, in __call__
    return self.call(*args, block_identifier=block_identifier)
  File "/usr/local/lib/python3.8/dist-packages/brownie/network/contract.py", line 1461, in call
    raise VirtualMachineError(e) from None
VirtualMachineError: revert
>>> contract_SUSHISWAP_ROUTER.getAmountsIn(5000_000000000000000000, [contract_WBTC, contract_APECOIN])
  File "<console>", line 1, in <module>
  File "/usr/local/lib/python3.8/dist-packages/brownie/network/contract.py", line 1665, in __call__
    return self.call(*args, block_identifier=block_identifier)
  File "/usr/local/lib/python3.8/dist-packages/brownie/network/contract.py", line 1461, in call
    raise VirtualMachineError(e) from None
VirtualMachineError: revert
```

In this case, the returned values will be taken from the path (tokenIn_ => WETH => tokenOut_):

```
>>> contract_UNISWAPV2_ROUTER.getAmountsIn(5000_000000000000000000, [contract_WBTC, contract_WETH, contract_APECOIN])
(1290119022, 2120276910339257143, 5000000000000000000000)
>>> contract_SUSHISWAP_ROUTER.getAmountsIn(5000_000000000000000000, [contract_WBTC, contract_WETH, contract_APECOIN])
(310462612, 5095184732694640459, 5000000000000000000000)
```

The _amountIn retrieved when this last path is used is way higher, as there is an extra jump in the swap path. On the other hand, SwapperOracle is used to retrieve the _amountInMax. Chainlink price feeds are used here as primary oracle. This results in an _amountInMax way lower than the _amountIn value returned.

The same issue is also present in the Swapper.getBestAmountOut() function. But in this function, the _amountOutMin is higher than the _amountOut returned.

Proof of Concept:

1. getBestAmountIn(): USDC => WETH, 1_000000000000000000
2. getBestAmountIn(): USDC => DAI, 1_000000000000000000
3. getBestAmountIn(): WBTC => APECOIN, 5000_000000000000000000: Issue occurs here.

```
contract_Swapper.getBestAmountIn(contract_USDC.address, contract_WETH.address, 1_000000000000000000) ->
(1815526845, 1832147425, '0x15997189c2176B904d0Ae1012204d91Cd83AE2Cd', ('0xA0b86991c6218b36c1d19D4a2e9Eb0cE3606eB48', '0xC02aaA39b223FE8D0A0e5C4F27eAD9083C756Cc2'))

contract_Swapper.getBestAmountIn(contract_USDC.address, contract_DAI.address, 1_000000000000000000) ->
(1001087, 1009597, '0x0cd62D69A792d91CB10B1a2934086766 3DD753AC', ('0xA0b86991c6218b36c1d19D4a2e9Eb0cE3606eB48', '0x6B175474E89094C44Da98b954EedeAC495271d0F'))

contract_Swapper.getBestAmountIn(contract_WBTC.address, contract_APECOIN.address, 5000_000000000000000000) ->
(129019022, 104629050, '0x7744b4F4565ebcB2Fd833641C75D7F9c90159a02', ('0x2260FAC5E5542a773Aa44fBCfeDf7C193bc2C599', '0xC02aaA39b223FE8D0A0e5C4F27eAD9083C756Cc2', '0x4d224452801ACEd8B2F0aebE155379bb5D594381'))
```

WETH

We can see how in the last call _amountInMax is lower than _amountIn:
- _amountIn => 1_29019022
- _amountInMax => 1_04629050

And also below, we can appreciate the same issue in the getBestAmountOut() function:

```
contract_Swapper.getBestAmountOut(contract_APECOIN.address, contract_WBTC.address, 5000_000000000000000000) ->
(83586929, 102557107, '0x7744b4F4565ebcB2Fd833641C75D7F9c90159a02', ('0x4d224452801ACEd8B2F0aebE155379bb5D594381', '0xC02aaA39b223FE8D0A0e5C4F27eAD9083C756Cc2', '0x2260FAC5E5542a773Aa44fBCfeDf7C193bc2C599'))
```

In this case, we are getting an _amountOutMin higher than _amountOut.

Risk Level:

Likelihood - 4
Impact - 4

Recommendation:

An _amountIn higher than _amountInMax in the getBestAmountIn() return values will cause that the swapExactOutput() call fails, as the user would not have approved enough tokens. Something similar occurs with getBestAmountOut() and swapExactInput() with an _amountOutMin higher than _amountOut. Users in this case may receive an amount of tokens way lower

than what they expected from the swap.

It is recommended to revisit the logic of these functions, so these cases cannot happen.

3.4 (HAL-04) INCOMPATIBILITY WITH TRANSFER-ON-FEE OR DEFLATIONARY TOKENS - HIGH

Description:

In the Swapper contract, the functions swapExactInput() and swapExactOutput() are used to swap tokens in an exchange:

Listing 8: Swapper.sol (Lines 148,149,165,166)

```
137 function swapExactInput(
138     address tokenIn_,
139     address tokenOut_,
140     uint256 amountIn_,
141     address receiver_
142 ) external returns (uint256 _amountOut) {
143     (, uint256 _amountOutMin, IExchange _exchange, address[]
 ↳ memory _path) = getBestAmountOut(
144         tokenIn_,
145         tokenOut_,
146         amountIn_
147     );
148     IERC20(tokenIn_).safeTransferFrom(_msgSender(), address(
 ↳ _exchange), amountIn_);
149     return _exchange.swapExactInput(_path, amountIn_,
 ↳ _amountOutMin, receiver_);
150 }
151
152 /// @inheritdoc ISwapper
153 function swapExactOutput(
154     address tokenIn_,
155     address tokenOut_,
156     uint256 amountOut_,
157     address receiver_
158 ) external returns (uint256 _amountIn) {
159     (, uint256 _amountInMax, IExchange _exchange, address[] memory
 ↳  _path) = getBestAmountIn(
160         tokenIn_,
161         tokenOut_,
162         amountOut_
```

```
163     );
164     address _caller = _msgSender();
165     IERC20(tokenIn_).safeTransferFrom(_caller, address(_exchange),
 ↳  _amountInMax);
166     return _exchange.swapExactOutput(_path, amountOut_,
 ↳ _amountInMax, _caller, receiver_);
167 }
```

The functions assume that the IERC20(tokenIn_).safeTransferFrom() call will transfer the full amountIn_ and _amountInMax to the Swapper contract.

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

If this occurs the swap call will revert as we can see in the proof of concept below.

Proof of Concept:

```
Calling -> contract_USDT.setParams(10, 20, {'from': contract_USDT.owner()}) SETTING A 1% FEE
Transaction sent: 0x5d2935a26f699f1796cf9d785ef1bd486bfbb50a3f0e8f7ebf7604367ab7ff71
  Gas price: 0.0 gwei   Gas limit: 600000000   Nonce: 1
  USDT.setParams confirmed   Block: 14891371   Gas used: 66957 (0.01%)

contract_USDT.balanceOf(user1) -> 5000000000
Calling -> contract_USDT.approve(contract_Swapper.address, 5000_000000, {'from': user1})
Transaction sent: 0x5f902b03d2e1b4d9f9b5999be2aceec1298bca22a4e5e92bf973a3bece9177d3
  Gas price: 0.0 gwei   Gas limit: 600000000   Nonce: 2
  USDT.approve confirmed   Block: 14891372   Gas used: 45961 (0.01%)

Calling -> contract_Swapper.swapExactInput(contract_USDT.address, contract_WETH.address, 5000_000000, user1.address, {'from': user1})
Transaction sent: 0x39b3d83670e11de125ab434b06a0cd49af8b67aca745b1fe51c6b40969cd83ed
  Gas price: 0.0 gwei   Gas limit: 600000000   Nonce: 3
  Swapper.swapExactInput confirmed (TransferHelper: TRANSFER_FROM_FAILED)   Block: 14891373   Gas used: 572322191 (95.39%)
```

Risk Level:

Likelihood - 3
Impact - 5

Recommendation:

It is recommended to get the exact received amount of tokenIn_ by calculating the difference of tokenIn_ balance before and after the transfer and use it in the swapExactInput() and swapExactOutput() calls.

3.5 (HAL-05) MAX DEVIATION TOO HIGH IN SWAPPERORACLE - MEDIUM

Description:

In the contract SwapperOracle, the constant MAX_DEVIATION sets the max deviation accepted between 2 different price providers:

Listing 9: SwapperOracle.sol

```
16 uint256 internal constant MAX_DEVIATION = 0.1e18;
```

Listing 10: SwapperOracle.sol (Line 102)

```
62 function quote(
63     address tokenIn_,
64     address tokenOut_,
65     uint256 amountIn_
66 ) external view returns (uint256 _amountOut) {
67     // 1. Get price from chainlink
68     uint256 _lastUpdatedAt;
69     (_amountOut, _lastUpdatedAt) = _quote(DataTypes.Provider.
↳ CHAINLINK, tokenIn_, tokenOut_, amountIn_);
70
71     // 2. If price from chainlink is OK return it
72     if (_amountOut > 0 && !_priceIsStale(_lastUpdatedAt)) {
73         return _amountOut;
74     }
75
76     // 3. Get price from fallback A
77     (uint256 _amountOutA, uint256 _lastUpdatedAtA) = _quote(
↳ fallbackProviderA, tokenIn_, tokenOut_, amountIn_);
78
79     // 4. If price from fallback A is OK and there isn't a
↳ fallback B, return price from fallback A
80     bool _aPriceOK = _amountOutA > 0 && !_priceIsStale(
↳ _lastUpdatedAtA);
81     if (fallbackProviderB == DataTypes.Provider.NONE) {
82         require(_aPriceOK, "fallback-a-failed");
83         return _amountOutA;
84     }
```

```
85
86      // 5. Get price from fallback B
87      (uint256 _amountOutB, uint256 _lastUpdatedAtB) = _quote(
↳ fallbackProviderB, tokenIn_, tokenOut_, amountIn_);
88
89      // 6. If only one price from fallbacks is valid, return it
90      bool _bPriceOK = _amountOutB > 0 && !_priceIsStale(
↳ _lastUpdatedAtB);
91      if (!_bPriceOK && _aPriceOK) {
92          return _amountOutA;
93      } else if (_bPriceOK && !_aPriceOK) {
94          return _amountOutB;
95      }
96
97      // 7. Check fallback prices deviation
98      require(_aPriceOK && _bPriceOK, "fallbacks-failed");
99      uint256 _deviation = _amountOutA > _amountOutB
100         ? ((_amountOutA - _amountOutB) * 1e18) / _amountOutA
101         : ((_amountOutB - _amountOutA) * 1e18) / _amountOutB;
102     require(_deviation <= MAX_DEVIATION, "prices-deviation-too-
↳ high");
103
104     // 8. If deviation is OK, return price from fallback A
105     return _amountOutA;
106 }
```

If the initial price provider does not return a correct price and there are 2 different fallback price providers set and both return correct prices, the price returned by these two will be compared with each other and if the difference is less than the 10% the price will be considered valid. Then the price of the fallback price provider A will be returned.

A 10% price deviation is way too high and may mean that the prices received are not correct.

Risk Level:

Likelihood - 3
Impact - 3

Recommendation:

It is recommended to lower the MAX_DEVIATION constant to a max. of 1-2%, instead of 10:

Listing 11: SwapperOracle.sol

```
16 uint256 internal constant MAX_DEVIATION = 0.02e18;
```

3.6 (HAL-06) POSSIBLE MISUSE OF PUBLIC FUNCTIONS - INFORMATIONAL

Description:

In the contracts below contracts, there are some functions marked as public that are never called directly within the contract itself or in any of their descendants:

ChainlinkPriceProvider.sol
- getPriceInUsd() (ChainlinkPriceProvider.sol#31-33)

UniswapV2LikePriceProvider.sol
- updateDefaultTwapPeriod() (UniswapV2LikePriceProvider.sol#218-221)

SwapperOracle.sol
- updateFallbackProviders() (SwapperOracle.sol#131-139)
- updateProvidersAggregator() (SwapperOracle.sol#144-148)
- updateStalePeriod() (SwapperOracle.sol#153-156)

UniswapV3CrossPoolOracle.sol
- assetToAssetThruRoute() (UniswapV3CrossPoolOracle.sol#57-83)

Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

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

3.7 (HAL-07) STATE VARIABLES MISSING IMMUTABLE MODIFIER - INFORMATIONAL

Description:

Some contracts have state variables that can be declared as immutable to save some gas.

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:

CTokenOracle.sol
- Line 18: ISynthOracle public underlyingOracle;
- Line 23: address public wethLike;

CurveLpTokenOracle.sol
- Line 13: ICurveAddressProvider public addressProvider;

SynthDefaultOracle.sol
- Line 31: IChainlinkPriceProvider public chainlinkProvider;

Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

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

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:

Governable.sol

```
Address.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#201-221) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/utils/Address.sol#213-216)
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', '^0.8.1', '^0.8.2']
        - ^0.8.2 (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#4)
        - ^0.8.1 (node_modules/@openzeppelin/contracts/utils/Address.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#4)
        - 0.8.9 (contracts/access/Governable.sol#3)
        - 0.8.9 (contracts/interfaces/IGovernable.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

Address.functionCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#85-87) is never used and should be removed
Address.functionCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#95-101) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#114-120) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#128-139) is never used and should be removed
Address.functionDelegateCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#174-176) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#184-193) is never used and should be removed
Address.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#147-149) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#157-166) is never used and should be removed
Address.isContract(address) (node_modules/@openzeppelin/contracts/utils/Address.sol#36-42) is never used and should be removed
Address.sendValue(address,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#60-65) is never used and should be removed
Address.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#201-221) is never used and should be removed
Context._msgData() (node_modules/@openzeppelin/contracts/utils/Context.sol#21-23) is never used and should be removed
Governable.__Governable_init() (contracts/access/Governable.sol#38-42) is never used and should be removed
Initializable._disableInitializers() (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#129-131) is never used and should be removed
Initializable._setInitializedVersion(uint8) (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#133-148) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version^0.8.2 (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#4) allows old versions
Pragma version^0.8.1 (node_modules/@openzeppelin/contracts/utils/Address.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#4) allows old versions
Pragma version0.8.9 (contracts/access/Governable.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/interfaces/IGovernable.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) (node_modules/@openzeppelin/contracts/utils/Address.sol#60-65):
        - (success) = recipient.call{value: amount}() (node_modules/@openzeppelin/contracts/utils/Address.sol#63)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#128-139):
        - (success,returndata) = target.call{value: value}(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#137)
Low level call in Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#157-166):
        - (success,returndata) = target.staticcall(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#164)
Low level call in Address.functionDelegateCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#184-193):
        - (success,returndata) = target.delegatecall(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#191)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function Governable.__Governable_init() (contracts/access/Governable.sol#38-42) is not in mixedCase
Parameter Governable.transferGovernorship(address)._proposedGovernor (contracts/access/Governable.sol#56) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
```

ChainlinkArbitrumPriceProvider.sol

```
Address.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#201-221) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/utils/Address.sol#213-216)
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', '^0.8.1', '^0.8.2']
        - ^0.8.0 (node_modules/@chainlink/contracts/src/v0.8/interfaces/AggregatorV3Interface.sol#2)
        - ^0.8.2 (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol#4)
        - ^0.8.1 (node_modules/@openzeppelin/contracts/utils/Address.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/utils/math/Math.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#4)
        - 0.8.9 (contracts/access/Governable.sol#3)
        - 0.8.9 (contracts/core/ChainlinkArbitrumPriceProvider.sol#3)
        - 0.8.9 (contracts/core/ChainlinkPriceProvider.sol#3)
        - 0.8.9 (contracts/interfaces/IGovernable.sol#3)
        - 0.8.9 (contracts/interfaces/core/IChainlinkPriceProvider.sol#3)
        - 0.8.9 (contracts/interfaces/core/IPriceProvider.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

Address.functionCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#85-87) is never used and should be removed
Address.functionCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#95-101) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#114-120) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#128-139) is never used and should be removed
Address.functionDelegateCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#174-176) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#184-193) is never used and should be removed
Address.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#147-149) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#157-166) is never used and should be removed
Address.isContract(address) (node_modules/@openzeppelin/contracts/utils/Address.sol#36-42) is never used and should be removed
Address.sendValue(address,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#60-65) is never used and should be removed
Address.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#201-221) is never used and should be removed
Context._msgData() (node_modules/@openzeppelin/contracts/utils/Context.sol#21-23) is never used and should be removed
Governable.__Governable_init() (contracts/access/Governable.sol#38-42) is never used and should be removed
```

```
Initializable._disableInitializers() (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#129-131) is never used and should be removed
Initializable._setInitializedVersion(uint8) (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#133-145) is never used and should be removed
Math.average(uint256,uint256) (node_modules/@openzeppelin/contracts/utils/math/Math.sol#28-31) is never used and should be removed
Math.ceilDiv(uint256,uint256) (node_modules/@openzeppelin/contracts/utils/math/Math.sol#39-42) is never used and should be removed
Math.max(uint256,uint256) (node_modules/@openzeppelin/contracts/utils/math/Math.sol#13-15) is never used and should be removed
SafeCast.toInt128(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#152-155) is never used and should be removed
SafeCast.toInt16(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#206-209) is never used and should be removed
SafeCast.toInt256(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#236-240) is never used and should be removed
SafeCast.toInt32(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#185-191) is never used and should be removed
SafeCast.toInt64(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#170-173) is never used and should be removed
SafeCast.toInt8(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#224-227) is never used and should be removed
SafeCast.toUint128(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#47-50) is never used and should be removed
SafeCast.toUint16(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#107-110) is never used and should be removed
SafeCast.toUint224(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#32-35) is never used and should be removed
SafeCast.toUint32(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#92-95) is never used and should be removed
SafeCast.toUint64(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#77-80) is never used and should be removed
SafeCast.toUint8(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#122-125) is never used and should be removed
SafeCast.toUint96(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#62-65) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version^0.8.0 (node_modules/@chainlink/contracts/src/v0.8/interfaces/AggregatorV3Interface.sol#2) allows old versions
Pragma version^0.8.2 (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol#4) allows old versions
Pragma version^0.8.1 (node_modules/@openzeppelin/contracts/utils/Address.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/math/Math.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#4) allows old versions
Pragma version0.8.9 (contracts/access/Governable.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/core/ChainlinkArbitrumPriceProvider.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/core/ChainlinkPriceProvider.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/interfaces/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 (contracts/interfaces/core/IChainlinkPriceProvider.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/interfaces/core/IPriceProvider.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) (node_modules/@openzeppelin/contracts/utils/Address.sol#60-65):
        - (success) = recipient.call{value: amount}() (node_modules/@openzeppelin/contracts/utils/Address.sol#63)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#128-139):
        - (success,returndata) = target.call{value: value}(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#137)
Low level call in Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#157-166):
        - (success,returndata) = target.staticcall(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#164)
Low level call in Address.functionDelegateCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#184-193):
        - (success,returndata) = target.delegatecall(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#191)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function Governable.__Governable_init() (contracts/access/Governable.sol#38-42) is not in mixedCase
Parameter Governable.transferGovernorship(address)._proposedGovernor (contracts/access/Governable.sol#56) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Variable ChainlinkPriceProvider._aggregatorOf(address)._aggregator (contracts/core/ChainlinkPriceProvider.sol#75) is too similar to ChainlinkPriceProvider.updateAggregator(address,AggregatorV3Interface).aggregator_ (contracts/core/ChainlinkPriceProvider.sol#93)
Variable ChainlinkPriceProvider._aggregatorOf(address)._aggregator (contracts/core/ChainlinkPriceProvider.sol#75) is too similar to ChainlinkPriceProvider.aggregators (contracts/core/ChainlinkPriceProvider.sol#25)
Variable ChainlinkPriceProvider.updateAggregator(address,AggregatorV3Interface).aggregator_ (contracts/core/ChainlinkPriceProvider.sol#93) is too similar to ChainlinkPriceProvider.aggregators (contracts/core/ChainlinkPriceProvider.sol#25
)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-are-too-similar

getPriceInUsd(address) should be declared external:
        - ChainlinkPriceProvider.getPriceInUsd(address) (contracts/core/ChainlinkPriceProvider.sol#31-33)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

ChainlinkAvalanchePriceProvider.sol

```
Address.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#201-221) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/utils/Address.sol#213-216)
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', '^0.8.1', '^0.8.2']
        - ^0.8.0 (node_modules/@chainlink/contracts/src/v0.8/interfaces/AggregatorV3Interface.sol#2)
        - ^0.8.2 (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol#4)
        - ^0.8.1 (node_modules/@openzeppelin/contracts/utils/Address.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/utils/math/Math.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#4)
        - 0.8.9 (contracts/access/Governable.sol#3)
        - 0.8.9 (contracts/core/ChainlinkAvalanchePriceProvider.sol#3)
        - 0.8.9 (contracts/core/ChainlinkPriceProvider.sol#3)
        - 0.8.9 (contracts/interfaces/IGovernable.sol#3)
        - 0.8.9 (contracts/interfaces/core/IChainlinkPriceProvider.sol#3)
        - 0.8.9 (contracts/interfaces/core/IPriceProvider.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

Address.functionCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#85-87) is never used and should be removed
Address.functionCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#95-101) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#118-120) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#128-139) is never used and should be removed
Address.functionDelegateCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#174-176) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#184-193) is never used and should be removed
Address.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#147-149) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#157-166) is never used and should be removed
Address.isContract(address) (node_modules/@openzeppelin/contracts/utils/Address.sol#36-42) is never used and should be removed
Address.sendValue(address,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#60-65) is never used and should be removed
Address.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#201-221) is never used and should be removed
Context._msgData() (node_modules/@openzeppelin/contracts/utils/Context.sol#21-23) is never used and should be removed
Governable.__Governable_init() (contracts/access/Governable.sol#38-42) is never used and should be removed
Initializable._disableInitializers() (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#129-131) is never used and should be removed
Initializable._setInitializedVersion(uint8) (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#133-145) is never used and should be removed
Math.average(uint256,uint256) (node_modules/@openzeppelin/contracts/utils/math/Math.sol#28-31) is never used and should be removed
Math.ceilDiv(uint256,uint256) (node_modules/@openzeppelin/contracts/utils/math/Math.sol#39-42) is never used and should be removed
Math.max(uint256,uint256) (node_modules/@openzeppelin/contracts/utils/math/Math.sol#13-15) is never used and should be removed
SafeCast.toInt128(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#152-155) is never used and should be removed
SafeCast.toInt16(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#206-209) is never used and should be removed
SafeCast.toInt256(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#236-240) is never used and should be removed
SafeCast.toInt32(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#185-191) is never used and should be removed
SafeCast.toInt64(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#170-173) is never used and should be removed
SafeCast.toInt8(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#224-227) is never used and should be removed
SafeCast.toUint128(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#47-50) is never used and should be removed
SafeCast.toUint16(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#107-110) is never used and should be removed
SafeCast.toUint224(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#32-35) is never used and should be removed
SafeCast.toUint32(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#92-95) is never used and should be removed
SafeCast.toUint64(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#77-80) is never used and should be removed
SafeCast.toUint8(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#122-125) is never used and should be removed
SafeCast.toUint96(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#62-65) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version^0.8.0 (node_modules/@chainlink/contracts/src/v0.8/interfaces/AggregatorV3Interface.sol#2) allows old versions
Pragma version^0.8.2 (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol#4) allows old versions
Pragma version^0.8.1 (node_modules/@openzeppelin/contracts/utils/Address.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/math/Math.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#4) allows old versions
Pragma version0.8.9 (contracts/access/Governable.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/core/ChainlinkAvalanchePriceProvider.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/core/ChainlinkPriceProvider.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/interfaces/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 (contracts/interfaces/core/IChainlinkPriceProvider.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/interfaces/core/IPriceProvider.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) (node_modules/@openzeppelin/contracts/utils/Address.sol#60-65):
        - (success) = recipient.call{value: amount}() (node_modules/@openzeppelin/contracts/utils/Address.sol#63)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#128-139):
        - (success,returndata) = target.call{value: value}(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#137)
Low level call in Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#157-166):
        - (success,returndata) = target.staticcall(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#164)
Low level call in Address.functionDelegateCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#184-193):
        - (success,returndata) = target.delegatecall(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#191)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function Governable.__Governable_init() (contracts/access/Governable.sol#38-42) is not in mixedCase
Parameter Governable.transferGovernorship(address)._proposedGovernor (contracts/access/Governable.sol#56) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Variable ChainlinkPriceProvider._aggregatorOf(address)._aggregator (contracts/core/ChainlinkPriceProvider.sol#75) is too similar to ChainlinkPriceProvider.updateAggregator(address,AggregatorV3Interface).aggregator_ (contracts/core/ChainlinkPriceProvider.sol#93)
Variable ChainlinkPriceProvider._aggregatorOf(address)._aggregator (contracts/core/ChainlinkPriceProvider.sol#75) is too similar to ChainlinkPriceProvider.aggregators (contracts/core/ChainlinkPriceProvider.sol#25)
Variable ChainlinkPriceProvider.updateAggregator(address,AggregatorV3Interface).aggregator_ (contracts/core/ChainlinkPriceProvider.sol#93) is too similar to ChainlinkPriceProvider.aggregators (contracts/core/ChainlinkPriceProvider.sol#25
)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-are-too-similar

getPriceInUsd(address) should be declared external:
        - ChainlinkPriceProvider.getPriceInUsd(address) (contracts/core/ChainlinkPriceProvider.sol#31-33)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

ChainlinkMainnetPriceProvider.sol

```
Address.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#201-221) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/utils/Address.sol#213-216)
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', '^0.8.1', '^0.8.2']
        - ^0.8.0 (node_modules/@chainlink/contracts/src/v0.8/interfaces/AggregatorInterface.sol#2)
        - ^0.8.0 (node_modules/@chainlink/contracts/src/v0.8/interfaces/AggregatorV2V3Interface.sol#2)
        - ^0.8.0 (node_modules/@chainlink/contracts/src/v0.8/interfaces/AggregatorV3Interface.sol#2)
        - ^0.8.0 (node_modules/@chainlink/contracts/src/v0.8/interfaces/FeedRegistryInterface.sol#2)
        - v2 (node_modules/@chainlink/contracts/src/v0.8/interfaces/FeedRegistryInterface.sol#3)
        - ^0.8.2 (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol#4)
        - ^0.8.1 (node_modules/@openzeppelin/contracts/utils/Address.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/utils/math/Math.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#4)
        - 0.8.9 (contracts/access/Governable.sol#3)
        - 0.8.9 (contracts/core/ChainlinkMainnetPriceProvider.sol#3)
        - 0.8.9 (contracts/core/ChainlinkPriceProvider.sol#3)
        - 0.8.9 (contracts/interfaces/IGovernable.sol#3)
        - 0.8.9 (contracts/interfaces/core/IChainlinkPriceProvider.sol#3)
        - 0.8.9 (contracts/interfaces/core/IPriceProvider.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

Address.functionCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#85-87) is never used and should be removed
Address.functionCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#95-101) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#114-120) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#128-139) is never used and should be removed
Address.functionDelegateCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#174-176) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#184-193) is never used and should be removed
Address.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#147-149) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#157-166) is never used and should be removed
Address.isContract(address) (node_modules/@openzeppelin/contracts/utils/Address.sol#36-42) is never used and should be removed
Address.sendValue(address,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#60-65) is never used and should be removed
Address.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#201-221) is never used and should be removed
ChainlinkPriceProvider._aggregatorOf(address) (contracts/core/ChainlinkPriceProvider.sol#75-78) is never used and should be removed
ChainlinkPriceProvider._getUsdPriceOfAsset(address) (contracts/core/ChainlinkPriceProvider.sol#85-88) is never used and should be removed
Context._msgData() (node_modules/@openzeppelin/contracts/utils/Context.sol#21-23) is never used and should be removed
Governable.__Governable_init() (contracts/access/Governable.sol#38-42) is never used and should be removed
Initializable._disableInitializers() (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#129-131) is never used and should be removed
Initializable._setInitializedVersion(uint8) (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#133-148) is never used and should be removed
Math.average(uint256,uint256) (node_modules/@openzeppelin/contracts/utils/math/Math.sol#28-31) is never used and should be removed
Math.ceilDiv(uint256,uint256) (node_modules/@openzeppelin/contracts/utils/math/Math.sol#39-42) is never used and should be removed
Math.max(uint256,uint256) (node_modules/@openzeppelin/contracts/utils/math/Math.sol#13-15) is never used and should be removed
SafeCast.toInt128(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#152-155) is never used and should be removed
SafeCast.toInt16(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#206-209) is never used and should be removed
SafeCast.toInt256(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#236-240) is never used and should be removed
SafeCast.toInt32(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#188-191) is never used and should be removed
SafeCast.toInt64(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#170-173) is never used and should be removed
SafeCast.toInt8(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#224-227) is never used and should be removed
SafeCast.toUint128(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#47-50) is never used and should be removed
SafeCast.toUint16(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#107-110) is never used and should be removed
SafeCast.toUint224(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#32-35) is never used and should be removed
SafeCast.toUint32(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#92-95) is never used and should be removed
SafeCast.toUint64(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#77-80) is never used and should be removed
SafeCast.toUint8(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#122-125) is never used and should be removed
SafeCast.toUint96(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#62-65) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version^0.8.0 (node_modules/@chainlink/contracts/src/v0.8/interfaces/AggregatorInterface.sol#2) allows old versions
Pragma version^0.8.0 (node_modules/@chainlink/contracts/src/v0.8/interfaces/AggregatorV2V3Interface.sol#2) allows old versions
Pragma version^0.8.0 (node_modules/@chainlink/contracts/src/v0.8/interfaces/AggregatorV3Interface.sol#2) allows old versions
Pragma version^0.8.0 (node_modules/@chainlink/contracts/src/v0.8/interfaces/FeedRegistryInterface.sol#2) allows old versions
Pragma version^0.8.2 (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol#4) allows old versions
Pragma version^0.8.1 (node_modules/@openzeppelin/contracts/utils/Address.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/math/Math.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#4) allows old versions
Pragma version0.8.9 (contracts/access/Governable.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/core/ChainlinkMainnetPriceProvider.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/core/ChainlinkPriceProvider.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/interfaces/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 (contracts/interfaces/core/IChainlinkPriceProvider.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/interfaces/core/IPriceProvider.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) (node_modules/@openzeppelin/contracts/utils/Address.sol#60-65):
        - (success) = recipient.call{value: amount}() (node_modules/@openzeppelin/contracts/utils/Address.sol#63)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#128-139):
        - (success,returndata) = target.call{value: value}(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#137)
Low level call in Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#157-166):
        - (success,returndata) = target.staticcall(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#164)
Low level call in Address.functionDelegateCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#184-193):
        - (success,returndata) = target.delegatecall(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#191)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function Governable.__Governable_init() (contracts/access/Governable.sol#38-42) is not in mixedCase
Parameter Governable.transferGovernorship(address)._proposedGovernor (contracts/access/Governable.sol#56) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Variable ChainlinkPriceProvider._aggregatorOf(address)._aggregator (contracts/core/ChainlinkPriceProvider.sol#75) is too similar to ChainlinkPriceProvider.updateAggregator(address,AggregatorV3Interface).aggregator_ (contracts/core/Chainl
inkPriceProvider.sol#93)
Variable ChainlinkPriceProvider._aggregatorOf(address)._aggregator (contracts/core/ChainlinkPriceProvider.sol#75) is too similar to ChainlinkPriceProvider.aggregators (contracts/core/ChainlinkPriceProvider.sol#25)
Variable ChainlinkPriceProvider.updateAggregator(address,AggregatorV3Interface).aggregator_ (contracts/core/ChainlinkPriceProvider.sol#93) is too similar to ChainlinkPriceProvider.aggregators (contracts/core/ChainlinkPriceProvider.sol#25
)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-are-too-similar

getPriceInUsd(address) should be declared external:
        - ChainlinkPriceProvider.getPriceInUsd(address) (contracts/core/ChainlinkPriceProvider.sol#31-33)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

ChainlinkPolygonPriceProvider.sol

```
Address.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#201-221) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/utils/Address.sol#213-216)
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', '^0.8.1', '^0.8.2']
        - ^0.8.0 (node_modules/@chainlink/contracts/src/v0.8/interfaces/AggregatorV3Interface.sol#2)
        - ^0.8.2 (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol#4)
        - ^0.8.1 (node_modules/@openzeppelin/contracts/utils/Address.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/utils/math/Math.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#4)
        - 0.8.9 (contracts/access/Governable.sol#3)
        - 0.8.9 (contracts/core/ChainlinkPolygonPriceProvider.sol#3)
        - 0.8.9 (contracts/core/ChainlinkPriceProvider.sol#3)
        - 0.8.9 (contracts/interfaces/IGovernable.sol#3)
        - 0.8.9 (contracts/interfaces/core/IChainlinkPriceProvider.sol#3)
        - 0.8.9 (contracts/interfaces/core/IPriceProvider.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

Address.functionCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#85-87) is never used and should be removed
Address.functionCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#95-101) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#114-120) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#128-139) is never used and should be removed
Address.functionDelegateCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#174-176) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#184-193) is never used and should be removed
Address.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#147-149) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#157-166) is never used and should be removed
Address.isContract(address) (node_modules/@openzeppelin/contracts/utils/Address.sol#36-42) is never used and should be removed
Address.sendValue(address,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#60-65) is never used and should be removed
Address.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#201-221) is never used and should be removed
Context._msgData() (node_modules/@openzeppelin/contracts/utils/Context.sol#21-23) is never used and should be removed
Governable.__Governable_init() (contracts/access/Governable.sol#38-42) is never used and should be removed
Initializable._disableInitializers() (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#129-131) is never used and should be removed
Initializable._setInitializedVersion(uint8) (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#133-148) is never used and should be removed
Math.average(uint256,uint256) (node_modules/@openzeppelin/contracts/utils/math/Math.sol#28-31) is never used and should be removed
Math.ceilDiv(uint256,uint256) (node_modules/@openzeppelin/contracts/utils/math/Math.sol#39-42) is never used and should be removed
Math.max(uint256,uint256) (node_modules/@openzeppelin/contracts/utils/math/Math.sol#13-15) is never used and should be removed
SafeCast.toInt128(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#152-155) is never used and should be removed
SafeCast.toInt16(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#206-209) is never used and should be removed
SafeCast.toInt256(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#236-240) is never used and should be removed
SafeCast.toInt32(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#188-191) is never used and should be removed
SafeCast.toInt64(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#170-173) is never used and should be removed
SafeCast.toInt8(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#224-227) is never used and should be removed
SafeCast.toUint128(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#47-50) is never used and should be removed
SafeCast.toUint16(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#107-110) is never used and should be removed
SafeCast.toUint224(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#32-35) is never used and should be removed
SafeCast.toUint32(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#92-95) is never used and should be removed
SafeCast.toUint64(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#77-80) is never used and should be removed
SafeCast.toUint8(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#122-125) is never used and should be removed
SafeCast.toUint96(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#62-65) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
```

```
Pragma version^0.8.0 (node_modules/@chainlink/contracts/src/v0.8/interfaces/AggregatorV3Interface.sol#2) allows old versions
Pragma version^0.8.2 (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol#4) allows old versions
Pragma version^0.8.1 (node_modules/@openzeppelin/contracts/utils/Address.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/math/Math.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#4) allows old versions
Pragma version0.8.9 (contracts/access/Governable.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/core/ChainlinkPolygonPriceProvider.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/core/ChainlinkPriceProvider.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/interfaces/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 (contracts/interfaces/core/IChainlinkPriceProvider.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/interfaces/core/IPriceProvider.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) (node_modules/@openzeppelin/contracts/utils/Address.sol#60-65):
        - (success) = recipient.call{value: amount}() (node_modules/@openzeppelin/contracts/utils/Address.sol#63)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#128-139):
        - (success,returndata) = target.call{value: value}(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#137)
Low level call in Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#157-166):
        - (success,returndata) = target.staticcall(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#164)
Low level call in Address.functionDelegateCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#184-193):
        - (success,returndata) = target.delegatecall(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#191)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function Governable.__Governable_init() (contracts/access/Governable.sol#38-42) is not in mixedCase
Parameter Governable.transferGovernorship(address)._proposedGovernor (contracts/access/Governable.sol#56) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Variable ChainlinkPriceProvider._aggregatorOf(address)._aggregator (contracts/core/ChainlinkPriceProvider.sol#75) is too similar to ChainlinkPriceProvider.updateAggregator(address,AggregatorV3Interface).aggregator_ (contracts/core/Chainl
inkPriceProvider.sol#93)
Variable ChainlinkPriceProvider._aggregatorOf(address)._aggregator (contracts/core/ChainlinkPriceProvider.sol#75) is too similar to ChainlinkPriceProvider.aggregators (contracts/core/ChainlinkPriceProvider.sol#25)
Variable ChainlinkPriceProvider.updateAggregator(address,AggregatorV3Interface).aggregator_ (contracts/core/ChainlinkPriceProvider.sol#93) is too similar to ChainlinkPriceProvider.aggregators (contracts/core/ChainlinkPriceProvider.sol#25
)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-are-too-similar

getPriceInUsd(address) should be declared external:
        - ChainlinkPriceProvider.getPriceInUsd(address) (contracts/core/ChainlinkPriceProvider.sol#31-33)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

ChainlinkPriceProvider.sol

```
Address.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#201-221) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/utils/Address.sol#213-216)
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', '^0.8.1', '^0.8.2']
        - ^0.8.0 (node_modules/@chainlink/contracts/src/v0.8/interfaces/AggregatorV3Interface.sol#2)
        - ^0.8.2 (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol#4)
        - ^0.8.1 (node_modules/@openzeppelin/contracts/utils/Address.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/utils/math/Math.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#4)
        - 0.8.9 (contracts/access/Governable.sol#3)
        - 0.8.9 (contracts/core/ChainlinkPriceProvider.sol#3)
        - 0.8.9 (contracts/interfaces/IGovernable.sol#3)
        - 0.8.9 (contracts/interfaces/core/IChainlinkPriceProvider.sol#3)
        - 0.8.9 (contracts/interfaces/core/IPriceProvider.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

Address.functionCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#85-87) is never used and should be removed
Address.functionCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#95-101) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#116-120) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#128-139) is never used and should be removed
Address.functionDelegateCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#174-176) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#184-193) is never used and should be removed
Address.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#147-149) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#157-166) is never used and should be removed
Address.isContract(address) (node_modules/@openzeppelin/contracts/utils/Address.sol#36-42) is never used and should be removed
Address.sendValue(address,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#60-65) is never used and should be removed
Address.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#201-221) is never used and should be removed
Context._msgData() (node_modules/@openzeppelin/contracts/utils/Context.sol#21-23) is never used and should be removed
Governable.__Governable_init() (contracts/access/Governable.sol#38-42) is never used and should be removed
Initializable._disableInitializers() (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#129-131) is never used and should be removed
Initializable._setInitializedVersion(uint8) (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#133-148) is never used and should be removed
Math.average(uint256,uint256) (node_modules/@openzeppelin/contracts/utils/math/Math.sol#25-31) is never used and should be removed
Math.ceilDiv(uint256,uint256) (node_modules/@openzeppelin/contracts/utils/math/Math.sol#39-42) is never used and should be removed
Math.max(uint256,uint256) (node_modules/@openzeppelin/contracts/utils/math/Math.sol#13-15) is never used and should be removed
SafeCast.toInt128(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#152-155) is never used and should be removed
SafeCast.toInt16(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#206-209) is never used and should be removed
SafeCast.toInt256(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#236-240) is never used and should be removed
SafeCast.toInt32(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#188-191) is never used and should be removed
SafeCast.toInt64(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#170-173) is never used and should be removed
SafeCast.toInt8(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#224-227) is never used and should be removed
SafeCast.toUint128(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#47-50) is never used and should be removed
SafeCast.toUint16(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#107-110) is never used and should be removed
SafeCast.toUint224(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#32-35) is never used and should be removed
SafeCast.toUint32(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#92-95) is never used and should be removed
SafeCast.toUint64(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#77-80) is never used and should be removed
SafeCast.toUint8(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#122-125) is never used and should be removed
SafeCast.toUint96(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#62-65) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version^0.8.0 (node_modules/@chainlink/contracts/src/v0.8/interfaces/AggregatorV3Interface.sol#2) allows old versions
Pragma version^0.8.2 (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol#4) allows old versions
Pragma version^0.8.1 (node_modules/@openzeppelin/contracts/utils/Address.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/math/Math.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#4) allows old versions
Pragma version0.8.9 (contracts/access/Governable.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/core/ChainlinkPriceProvider.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/interfaces/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 (contracts/interfaces/core/IChainlinkPriceProvider.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/interfaces/core/IPriceProvider.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) (node_modules/@openzeppelin/contracts/utils/Address.sol#60-65):
        - (success) = recipient.call{value: amount}() (node_modules/@openzeppelin/contracts/utils/Address.sol#63)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#128-139):
        - (success,returndata) = target.call{value: value}(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#137)
Low level call in Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#157-166):
        - (success,returndata) = target.staticcall(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#164)
Low level call in Address.functionDelegateCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#184-193):
        - (success,returndata) = target.delegatecall(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#191)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function Governable.__Governable_init() (contracts/access/Governable.sol#38-42) is not in mixedCase
Parameter Governable.transferGovernorship(address)._proposedGovernor (contracts/access/Governable.sol#56) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Variable ChainlinkPriceProvider._aggregatorOf(address)._aggregator (contracts/core/ChainlinkPriceProvider.sol#75) is too similar to ChainlinkPriceProvider.updateAggregator(address,AggregatorV3Interface).aggregator_ (contracts/core/Chainl
inkPriceProvider.sol#93)
Variable ChainlinkPriceProvider._aggregatorOf(address)._aggregator (contracts/core/ChainlinkPriceProvider.sol#75) is too similar to ChainlinkPriceProvider.aggregators (contracts/core/ChainlinkPriceProvider.sol#25)
Variable ChainlinkPriceProvider.updateAggregator(address,AggregatorV3Interface).aggregator_ (contracts/core/ChainlinkPriceProvider.sol#93) is too similar to ChainlinkPriceProvider.aggregators (contracts/core/ChainlinkPriceProvider.sol#25
)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-are-too-similar

getPriceInUsd(address) should be declared external:
        - ChainlinkPriceProvider.getPriceInUsd(address) (contracts/core/ChainlinkPriceProvider.sol#31-33)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

PriceProvidersAggregator.sol

```
Address.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#201-221) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/utils/Address.sol#213-216)
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', '^0.8.1', '^0.8.2']
        - ^0.8.2 (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#4)
        - ^0.8.1 (node_modules/@openzeppelin/contracts/utils/Address.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/utils/math/Math.sol#4)
        - 0.8.9 (contracts/access/Governable.sol#3)
        - 0.8.9 (contracts/core/PriceProvidersAggregator.sol#3)
        - 0.8.9 (contracts/interfaces/IGovernable.sol#3)
        - 0.8.9 (contracts/interfaces/core/IPriceProvider.sol#3)
        - 0.8.9 (contracts/interfaces/core/IPriceProvidersAggregator.sol#3)
        - 0.8.9 (contracts/libraries/DataTypes.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used
```

```
Address.functionCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#85-87) is never used and should be removed
Address.functionCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#95-101) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#114-120) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#123-139) is never used and should be removed
Address.functionDelegateCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#174-176) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#184-193) is never used and should be removed
Address.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#147-149) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#157-166) is never used and should be removed
Address.isContract(address) (node_modules/@openzeppelin/contracts/utils/Address.sol#36-42) is never used and should be removed
Address.sendValue(address,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#60-65) is never used and should be removed
Address.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#201-221) is never used and should be removed
Context._msgData() (node_modules/@openzeppelin/contracts/utils/Context.sol#21-23) is never used and should be removed
Governable._Governable_init() (contracts/access/Governable.sol#38-42) is never used and should be removed
Initializable._disableInitializers() (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#129-131) is never used and should be removed
Initializable._setInitializedVersion(uint8) (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#133-148) is never used and should be removed
Math.average(uint256,uint256) (node_modules/@openzeppelin/contracts/utils/math/Math.sol#28-31) is never used and should be removed
Math.ceilDiv(uint256,uint256) (node_modules/@openzeppelin/contracts/utils/math/Math.sol#39-42) is never used and should be removed
Math.max(uint256,uint256) (node_modules/@openzeppelin/contracts/utils/math/Math.sol#13-15) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version^0.8.2 (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#4) allows old versions
Pragma version^0.8.1 (node_modules/@openzeppelin/contracts/utils/Address.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/math/Math.sol#4) allows old versions
Pragma version0.8.9 (contracts/access/Governable.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/core/PriceProvidersAggregator.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/interfaces/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 (contracts/interfaces/core/IPriceProvider.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/interfaces/core/IPriceProvidersAggregator.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/libraries/DataTypes.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) (node_modules/@openzeppelin/contracts/utils/Address.sol#60-65):
        - (success) = recipient.call{value: amount}() (node_modules/@openzeppelin/contracts/utils/Address.sol#63)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#128-139):
        - (success,returndata) = target.call{value: value}(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#137)
Low level call in Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#157-166):
        - (success,returndata) = target.staticcall(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#164)
Low level call in Address.functionDelegateCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#184-193):
        - (success,returndata) = target.delegatecall(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#191)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function Governable.__Governable_init() (contracts/access/Governable.sol#38-42) is not in mixedCase
Parameter Governable.transferGovernorship(address)._proposedGovernor (contracts/access/Governable.sol#56) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Variable IPriceProvidersAggregator.quote(DataTypes.Provider,address,DataTypes.Provider,address,uint256)._providerIn (contracts/core/IPriceProvidersAggregator.sol#56) is too similar to IPriceProvidersAggregator.quote(DataTypes.Provider,address,DataTypes.Provider,address,uint256)._providerOut (contracts/core/IPriceProvidersAggregator.sol#60)
Variable IPriceProvidersAggregator.quote(DataTypes.Provider,address,DataTypes.Provider,address,uint256)._providerOut (contracts/core/IPriceProvidersAggregator.sol#63) is too similar to IPriceProvidersAggregator.quote(DataTypes.Provider,address,DataTypes.Provider,address,uint256).providerOut_ (contracts/core/PriceProvidersAggregator.sol#52)
Variable IPriceProvidersAggregator.priceProviders(DataTypes.Provider)._priceProvider (contracts/interfaces/core/IPriceProvidersAggregator.sol#12) is too similar to PriceProvidersAggregator.setPriceProvider(DataTypes.Provider,IPriceProvider).priceProvider_ (contracts/core/PriceProvidersAggregator.sol#73)
Variable IPriceProvidersAggregator.priceProviders(DataTypes.Provider)._priceProvider (contracts/interfaces/core/IPriceProvidersAggregator.sol#12) is too similar to IPriceProvidersAggregator.setPriceProvider(DataTypes.Provider,IPriceProvider).priceProvider_ (contracts/interfaces/core/IPriceProvidersAggregator.sol#55)
Variable PriceProvidersAggregator.setPriceProvider(DataTypes.Provider,IPriceProvider).priceProvider_ (contracts/core/PriceProvidersAggregator.sol#73) is too similar to PriceProvidersAggregator.priceProviders (contracts/core/PriceProvidersAggregator.sol#22)
Variable IPriceProvidersAggregator.priceProviders(DataTypes.Provider)._priceProvider (contracts/interfaces/core/IPriceProvidersAggregator.sol#12) is too similar to PriceProvidersAggregator.priceProviders (contracts/core/PriceProvidersAggregator.sol#22)
Variable IPriceProvidersAggregator.setPriceProvider(DataTypes.Provider,IPriceProvider).priceProvider_ (contracts/interfaces/core/IPriceProvidersAggregator.sol#55) is too similar to PriceProvidersAggregator.priceProviders (contracts/core/PriceProvidersAggregator.sol#22)
Variable PriceProvidersAggregator.quote(DataTypes.Provider,address,DataTypes.Provider,address,uint256)._providerIn (contracts/core/PriceProvidersAggregator.sol#56) is too similar to IPriceProvidersAggregator.quote(DataTypes.Provider,address,DataTypes.Provider,address,uint256).providerIn_ (contracts/interfaces/core/IPriceProvidersAggregator.sol#42)
Variable PriceProvidersAggregator.quote(DataTypes.Provider,address,DataTypes.Provider,address,uint256)._providerOut (contracts/core/PriceProvidersAggregator.sol#63) is too similar to IPriceProvidersAggregator.quote(DataTypes.Provider,address,DataTypes.Provider,address,uint256).providerOut_ (contracts/interfaces/core/IPriceProvidersAggregator.sol#44)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-are-too-similar

quote(DataTypes.Provider,address,DataTypes.Provider,address,uint256) should be declared external:
        - PriceProvidersAggregator.quote(DataTypes.Provider,address,DataTypes.Provider,address,uint256) (contracts/core/PriceProvidersAggregator.sol#49-70)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

UniswapV2LikePriceProvider.sol

```
UniswapV2OracleLibrary.currentBlockTimestamp() (node_modules/@uniswap/v2-periphery/contracts/libraries/UniswapV2OracleLibrary.sol#11-13) uses a weak PRNG: "uint32(block.timestamp % 2 ** 32) (node_modules/@uniswap/v2-periphery/contracts/libraries/UniswapV2OracleLibrary.sol#12)"
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#weak-PRNG

FixedPoint.mul(FixedPoint.uq112x112,uint256).z (node_modules/@uniswap/lib/contracts/libraries/FixedPoint.sol#44) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

UniswapV2LikePriceProvider.constructor(address,uint256,address).nativeToken_ (contracts/core/UniswapV2LikePriceProvider.sol#54) lacks a zero-check on :
        - nativeToken = nativeToken_ (contracts/core/UniswapV2LikePriceProvider.sol#59)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation

UniswapV2OracleLibrary.currentCumulativePrices(address) (node_modules/@uniswap/v2-periphery/contracts/libraries/UniswapV2OracleLibrary.sol#16-34) uses timestamp for comparisons
        Dangerous comparisons:
        - blockTimestampLast != blockTimestamp (node_modules/@uniswap/v2-periphery/contracts/libraries/UniswapV2OracleLibrary.sol#25)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

Address.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#201-221) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/utils/Address.sol#213-216)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Different versions of Solidity is used:
        - Version used: ['0.8.9', '>=0.4.0', '>=0.5.0', '^0.8.0', '^0.8.1', '^0.8.2']
        - ^0.8.2 (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#4)
        - ^0.8.1 (node_modules/@openzeppelin/contracts/utils/Address.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/utils/math/Math.sol#4)
        - >=0.4.0 (node_modules/@uniswap/lib/contracts/libraries/Babylonian.sol#3)
        - >=0.4.0 (node_modules/@uniswap/lib/contracts/libraries/FixedPoint.sol#3)
        - >=0.5.0 (node_modules/@uniswap/v2-core/contracts/interfaces/IUniswapV2Factory.sol#1)
        - >=0.5.0 (node_modules/@uniswap/v2-core/contracts/interfaces/IUniswapV2Pair.sol#1)
        - >=0.5.0 (node_modules/@uniswap/v2-periphery/contracts/libraries/UniswapV2OracleLibrary.sol#1)
        - 0.8.9 (contracts/access/Governable.sol#3)
        - 0.8.9 (contracts/core/UniswapV2LikePriceProvider.sol#3)
        - 0.8.9 (contracts/interfaces/IGovernable.sol#3)
        - 0.8.9 (contracts/interfaces/core/IPriceProvider.sol#3)
        - 0.8.9 (contracts/interfaces/core/IUniswapV2LikePriceProvider.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

Address.functionCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#85-87) is never used and should be removed
Address.functionCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#95-101) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#114-120) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#123-139) is never used and should be removed
Address.functionDelegateCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#174-176) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#184-193) is never used and should be removed
Address.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#147-149) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#157-166) is never used and should be removed
Address.isContract(address) (node_modules/@openzeppelin/contracts/utils/Address.sol#36-42) is never used and should be removed
Address.sendValue(address,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#60-65) is never used and should be removed
Address.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#201-221) is never used and should be removed
Babylonian.sqrt(uint256) (node_modules/@uniswap/lib/contracts/libraries/Babylonian.sol#4-20) is never used and should be removed
Context._msgData() (node_modules/@openzeppelin/contracts/utils/Context.sol#21-23) is never used and should be removed
FixedPoint.decode(FixedPoint.uq112x112) (node_modules/@uniswap/lib/contracts/libraries/FixedPoint.sol#57-59) is never used and should be removed
FixedPoint.div(FixedPoint.uq112x112,uint112) (node_modules/@uniswap/lib/contracts/libraries/FixedPoint.sol#36-39) is never used and should be removed
FixedPoint.encode144(uint144) (node_modules/@uniswap/lib/contracts/libraries/FixedPoint.sol#31-33) is never used and should be removed
FixedPoint.reciprocal(FixedPoint.uq112x112) (node_modules/@uniswap/lib/contracts/libraries/FixedPoint.sol#67-70) is never used and should be removed
FixedPoint.sqrt(FixedPoint.uq112x112) (node_modules/@uniswap/lib/contracts/libraries/FixedPoint.sol#73-75) is never used and should be removed
Governable._Governable_init() (contracts/access/Governable.sol#38-42) is never used and should be removed
Initializable._disableInitializers() (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#129-131) is never used and should be removed
Initializable._setInitializedVersion(uint8) (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#133-148) is never used and should be removed
Math.average(uint256,uint256) (node_modules/@openzeppelin/contracts/utils/math/Math.sol#28-31) is never used and should be removed
Math.ceilDiv(uint256,uint256) (node_modules/@openzeppelin/contracts/utils/math/Math.sol#39-42) is never used and should be removed
Math.max(uint256,uint256) (node_modules/@openzeppelin/contracts/utils/math/Math.sol#13-15) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version^0.8.2 (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#4) allows old versions
Pragma version^0.8.1 (node_modules/@openzeppelin/contracts/utils/Address.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/math/Math.sol#4) allows old versions
Pragma version>=0.4.0 (node_modules/@uniswap/lib/contracts/libraries/Babylonian.sol#3) allows old versions
Pragma version>=0.4.0 (node_modules/@uniswap/lib/contracts/libraries/FixedPoint.sol#3) allows old versions
Pragma version>=0.5.0 (node_modules/@uniswap/v2-core/contracts/interfaces/IUniswapV2Factory.sol#1) allows old versions
Pragma version>=0.5.0 (node_modules/@uniswap/v2-core/contracts/interfaces/IUniswapV2Pair.sol#1) allows old versions
Pragma version>=0.5.0 (node_modules/@uniswap/v2-periphery/contracts/libraries/UniswapV2OracleLibrary.sol#1) allows old versions
Pragma version0.8.9 (contracts/access/Governable.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/core/UniswapV2LikePriceProvider.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/interfaces/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 (contracts/interfaces/core/IPriceProvider.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/interfaces/core/IUniswapV2LikePriceProvider.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) (node_modules/@openzeppelin/contracts/utils/Address.sol#60-65):
        - (success) = recipient.call{value: amount}() (node_modules/@openzeppelin/contracts/utils/Address.sol#63)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#128-139):
        - (success,returndata) = target.call{value: value}(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#137)
Low level call in Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#157-166):
        - (success,returndata) = target.staticcall(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#164)
Low level call in Address.functionDelegateCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#184-193):
        - (success,returndata) = target.delegatecall(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#191)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
```

```
Struct FixedPoint.uq112x112 (node_modules/@uniswap/lib/contracts/libraries/FixedPoint.sol#11-13) is not in CapWords
Struct FixedPoint.uq144x112 (node_modules/@uniswap/lib/contracts/libraries/FixedPoint.sol#17-19) is not in CapWords
Function IUniswapV2Pair.DOMAIN_SEPARATOR() (node_modules/@uniswap/v2-core/contracts/interfaces/IUniswapV2Pair.sol#18) is not in mixedCase
Function IUniswapV2Pair.PERMIT_TYPEHASH() (node_modules/@uniswap/v2-core/contracts/interfaces/IUniswapV2Pair.sol#19) is not in mixedCase
Function IUniswapV2Pair.MINIMUM_LIQUIDITY() (node_modules/@uniswap/v2-core/contracts/interfaces/IUniswapV2Pair.sol#36) is not in mixedCase
Function Governable.__Governable_init() (contracts/access/Governable.sol#38-42) is not in mixedCase
Parameter Governable.transferGovernorship(address)._proposedGovernor (contracts/access/Governable.sol#56) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Variable UniswapV2OracleLibrary.currentCumulativePrices(address).price0Cumulative (node_modules/@uniswap/v2-periphery/contracts/libraries/UniswapV2OracleLibrary.sol#18) is too similar to UniswapV2OracleLibrary.currentCumulativePrices(add
ress).price1Cumulative (node_modules/@uniswap/v2-periphery/contracts/libraries/UniswapV2OracleLibrary.sol#18)
Variable UniswapV2LikePriceProvider._updateIfNeeded(IUniswapV2Pair,uint256).price0Cumulative (contracts/core/UniswapV2LikePriceProvider.sol#197) is too similar to UniswapV2LikePriceProvider._updateIfNeeded(IUniswapV2Pair,uint256).price1C
umulative (contracts/core/UniswapV2LikePriceProvider.sol#197)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-are-too-similar

updateDefaultTwapPeriod(uint256) should be declared external:
        - UniswapV2LikePriceProvider.updateDefaultTwapPeriod(uint256) (contracts/core/UniswapV2LikePriceProvider.sol#218-221)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

UniswapV3PriceProvider.sol

```
Address.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#201-221) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/utils/Address.sol#213-216)
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', '^0.8.1', '^0.8.2']
        - ^0.8.2 (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#4)
        - ^0.8.1 (node_modules/@openzeppelin/contracts/utils/Address.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#4)
        - 0.8.9 (contracts/access/Governable.sol#3)
        - 0.8.9 (contracts/core/UniswapV3PriceProvider.sol#3)
        - 0.8.9 (contracts/interfaces/IGovernable.sol#3)
        - 0.8.9 (contracts/interfaces/core/IPriceProvider.sol#3)
        - 0.8.9 (contracts/interfaces/core/IUniswapV3PriceProvider.sol#3)
        - 0.8.9 (contracts/interfaces/utils/IUniswapV3CrossPoolOracle.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

Address.functionCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#85-87) is never used and should be removed
Address.functionCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#95-101) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#114-120) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#128-139) is never used and should be removed
Address.functionDelegateCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#174-176) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#184-193) is never used and should be removed
Address.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#147-149) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#157-166) is never used and should be removed
Address.isContract(address) (node_modules/@openzeppelin/contracts/utils/Address.sol#36-42) is never used and should be removed
Address.sendValue(address,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#60-65) is never used and should be removed
Address.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#201-221) is never used and should be removed
Context._msgData() (node_modules/@openzeppelin/contracts/utils/Context.sol#21-23) is never used and should be removed
Governable._Governable_init() (contracts/access/Governable.sol#38-42) is never used and should be removed
Initializable._disableInitializers() (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#129-131) is never used and should be removed
Initializable._setInitializedVersion(uint8) (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#133-148) is never used and should be removed
SafeCast.toInt128(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#152-155) is never used and should be removed
SafeCast.toInt16(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#206-209) is never used and should be removed
SafeCast.toInt256(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#236-240) is never used and should be removed
SafeCast.toInt32(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#188-191) is never used and should be removed
SafeCast.toInt64(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#170-173) is never used and should be removed
SafeCast.toInt8(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#224-227) is never used and should be removed
SafeCast.toUint128(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#47-50) is never used and should be removed
SafeCast.toUint16(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#107-110) is never used and should be removed
SafeCast.toUint224(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#32-35) is never used and should be removed
SafeCast.toUint256(int256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#134-137) is never used and should be removed
SafeCast.toUint32(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#92-95) is never used and should be removed
SafeCast.toUint64(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#77-80) is never used and should be removed
SafeCast.toUint8(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#122-125) is never used and should be removed
SafeCast.toUint96(uint256) (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#62-65) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version^0.8.2 (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#4) allows old versions
Pragma version^0.8.1 (node_modules/@openzeppelin/contracts/utils/Address.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/math/SafeCast.sol#4) allows old versions
Pragma version0.8.9 (contracts/access/Governable.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/core/UniswapV3PriceProvider.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/interfaces/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 (contracts/interfaces/core/IPriceProvider.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/interfaces/core/IUniswapV3PriceProvider.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/interfaces/utils/IUniswapV3CrossPoolOracle.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) (node_modules/@openzeppelin/contracts/utils/Address.sol#60-65):
        - (success) = recipient.call{value: amount}() (node_modules/@openzeppelin/contracts/utils/Address.sol#63)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#128-139):
        - (success,returndata) = target.call{value: value}(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#137)
Low level call in Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#157-166):
        - (success,returndata) = target.staticcall(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#164)
Low level call in Address.functionDelegateCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#184-193):
        - (success,returndata) = target.delegatecall(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#191)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function Governable.__Governable_init() (contracts/access/Governable.sol#38-42) is not in mixedCase
Parameter Governable.transferGovernorship(address)._proposedGovernor (contracts/access/Governable.sol#56) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

updateDefaultTwapPeriod(uint32) should be declared external:
        - UniswapV3PriceProvider.updateDefaultTwapPeriod(uint32) (contracts/core/UniswapV3PriceProvider.sol#98-101)
updateDefaultPoolFee(uint24) should be declared external:
        - UniswapV3PriceProvider.updateDefaultPoolFee(uint24) (contracts/core/UniswapV3PriceProvider.sol#104-107)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

Oracle.sol

```
Oracle.setUSDEquivalentToken(address).usdEquivalentToken_ (contracts/periphery/Oracle.sol#148) lacks a zero-check on :
        - usdEquivalentToken = usdEquivalentToken_ (contracts/periphery/Oracle.sol#149)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation

Address.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#201-221) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/utils/Address.sol#213-216)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Different versions of Solidity is used:
        - Version used: ['0.8.9', '>=0.6.2', '^0.8.0', '^0.8.1', '^0.8.2']
        - ^0.8.2 (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol#4)
        - ^0.8.1 (node_modules/@openzeppelin/contracts/utils/Address.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#4)
        - 0.8.9 (contracts/access/Governable.sol#3)
        - 0.8.9 (contracts/interfaces/IGovernable.sol#3)
        - 0.8.9 (contracts/interfaces/core/IChainlinkPriceProvider.sol#3)
        - 0.8.9 (contracts/interfaces/core/IPriceProvider.sol#3)
        - 0.8.9 (contracts/interfaces/core/IPriceProvidersAggregator.sol#3)
        - 0.8.9 (contracts/interfaces/periphery/IOracle.sol#3)
        - 0.8.9 (contracts/libraries/DataTypes.sol#3)
        - >=0.6.2 (contracts/libraries/OpenZeppelinHelpers.sol#3)
        - 0.8.9 (contracts/periphery/Oracle.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

Address.functionCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#85-87) is never used and should be removed
Address.functionCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#95-101) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#114-120) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#128-139) is never used and should be removed
Address.functionDelegateCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#174-176) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#184-193) is never used and should be removed
Address.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#147-149) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#157-166) is never used and should be removed
Address.isContract(address) (node_modules/@openzeppelin/contracts/utils/Address.sol#36-42) is never used and should be removed
Address.sendValue(address,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#60-65) is never used and should be removed
Address.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#201-221) is never used and should be removed
Context._msgData() (node_modules/@openzeppelin/contracts/utils/Context.sol#21-23) is never used and should be removed
Governable._Governable_init() (contracts/access/Governable.sol#38-42) is never used and should be removed
Initializable._disableInitializers() (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#129-131) is never used and should be removed
Initializable._setInitializedVersion(uint8) (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#133-148) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version^0.8.2 (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol#4) allows old versions
Pragma version^0.8.1 (node_modules/@openzeppelin/contracts/utils/Address.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#4) allows old versions
```

```
Pragma version0.8.9 (contracts/access/Governable.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/interfaces/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 (contracts/interfaces/core/IChainlinkPriceProvider.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/interfaces/core/IPriceProvider.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/interfaces/core/IPriceProvidersAggregator.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/interfaces/periphery/IOracle.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/libraries/DataTypes.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version>=0.6.2 (contracts/libraries/OracleHelpers.sol#3) allows old versions
Pragma version0.8.9 (contracts/periphery/Oracle.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) (node_modules/@openzeppelin/contracts/utils/Address.sol#60-65):
        - (success) = recipient.call{value: amount}() (node_modules/@openzeppelin/contracts/utils/Address.sol#63)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#128-139):
        - (success,returndata) = target.call{value: value}(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#137)
Low level call in Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#157-166):
        - (success,returndata) = target.staticcall(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#164)
Low level call in Address.functionDelegateCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#184-193):
        - (success,returndata) = target.delegatecall(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#191)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function Governable.__Governable_init() (contracts/access/Governable.sol#38-42) is not in mixedCase
Parameter Governable._transferGovernorship(address)._proposedGovernor (contracts/access/Governable.sol#56) is not in mixedCase
Parameter OracleHelpers.scaleDecimal(uint256,uint256,uint256)._fromDecimal (contracts/libraries/OracleHelpers.sol#9) is not in mixedCase
Parameter OracleHelpers.scaleDecimal(uint256,uint256,uint256)._toDecimal (contracts/libraries/OracleHelpers.sol#10) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Variable IPriceProvidersAggregator.priceProviders(DataTypes.Provider)._priceProvider (contracts/interfaces/core/IPriceProvidersAggregator.sol#12) is too similar to IPriceProvidersAggregator.setPriceProvider(DataTypes.Provider,IPriceProvider).priceProvider_ (contracts/interfaces/core/IPriceProvidersAggregator.sol#55)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-are-too-similar
```

SwapperOracle.sol

```
SwapperOracle._quote(DataTypes.Provider,address,address,uint256).__amountOut (contracts/periphery/SwapperOracle.sol#119) is a local variable never initialized
SwapperOracle._quote(DataTypes.Provider,address,address,uint256).__lastUpdatedAt (contracts/periphery/SwapperOracle.sol#120) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

SwapperOracle._quote(DataTypes.Provider,address,address,uint256) (contracts/periphery/SwapperOracle.sol#112-125) ignores return value by providersAggregator.quote(provider_,tokenIn_,tokenOut_,amountIn_) (contracts/periphery/SwapperOracle.sol#118-124)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

Variable 'SwapperOracle._quote(DataTypes.Provider,address,address,uint256).__amountOut (contracts/periphery/SwapperOracle.sol#119)' in SwapperOracle._quote(DataTypes.Provider,address,address,uint256) (contracts/periphery/SwapperOracle.sol#112-125) potentially used before declaration: __amountOut = __amountOut (contracts/periphery/SwapperOracle.sol#122)
Variable 'SwapperOracle._quote(DataTypes.Provider,address,address,uint256).__lastUpdatedAt (contracts/periphery/SwapperOracle.sol#120)' in SwapperOracle._quote(DataTypes.Provider,address,address,uint256) (contracts/periphery/SwapperOracle.sol#112-125) potentially used before declaration: _lastUpdatedAt = __lastUpdatedAt (contracts/periphery/SwapperOracle.sol#123)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables

SwapperOracle.quote(address,address,uint256) (contracts/periphery/SwapperOracle.sol#62-106) uses timestamp for comparisons
        Dangerous comparisons:
        - _amountOut > 0 && ! _priceIsStale(_lastUpdatedAt) (contracts/periphery/SwapperOracle.sol#72)
        - _aPriceOK = _amountOutA > 0 && ! _priceIsStale(_lastUpdatedAtA) (contracts/periphery/SwapperOracle.sol#80)
        - require(bool,string)(_aPriceOK,fallback-a-failed) (contracts/periphery/SwapperOracle.sol#82)
        - _bPriceOK = _amountOutB > 0 && ! _priceIsStale(_lastUpdatedAtB) (contracts/periphery/SwapperOracle.sol#90)
        - ! _bPriceOK && _aPriceOK (contracts/periphery/SwapperOracle.sol#91)
        - _bPriceOK && ! _aPriceOK (contracts/periphery/SwapperOracle.sol#93)
        - require(bool,string)(_aPriceOK && _bPriceOK,fallbacks-failed) (contracts/periphery/SwapperOracle.sol#98)
SwapperOracle._priceIsStale(uint256) (contracts/periphery/SwapperOracle.sol#163-165) uses timestamp for comparisons
        Dangerous comparisons:
        - block.timestamp - _timeOfLastUpdate > stalePeriod (contracts/periphery/SwapperOracle.sol#164)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

Address.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#201-221) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/utils/Address.sol#213-216)
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', '^0.8.1', '^0.8.2']
        - ^0.8.2 (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol#4)
        - ^0.8.1 (node_modules/@openzeppelin/contracts/utils/Address.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#4)
        - 0.8.9 (contracts/access/Governable.sol#3)
        - 0.8.9 (contracts/interfaces/IGovernable.sol#3)
        - 0.8.9 (contracts/interfaces/core/IChainlinkPriceProvider.sol#3)
        - 0.8.9 (contracts/interfaces/core/IPriceProvider.sol#3)
        - 0.8.9 (contracts/interfaces/core/IPriceProvidersAggregator.sol#3)
        - 0.8.9 (contracts/interfaces/periphery/ISwapperOracle.sol#3)
        - 0.8.9 (contracts/libraries/DataTypes.sol#3)
        - 0.8.9 (contracts/periphery/SwapperOracle.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

Address.functionCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#85-87) is never used and should be removed
Address.functionCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#95-101) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#114-120) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#128-139) is never used and should be removed
Address.functionDelegateCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#174-176) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#184-193) is never used and should be removed
Address.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#147-149) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#157-166) is never used and should be removed
Address.isContract(address) (node_modules/@openzeppelin/contracts/utils/Address.sol#36-42) is never used and should be removed
Address.sendValue(address,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#60-65) is never used and should be removed
Address.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#201-221) is never used and should be removed
Context._msgData() (node_modules/@openzeppelin/contracts/utils/Context.sol#21-23) is never used and should be removed
Governable.__Governable_init() (contracts/access/Governable.sol#38-42) is never used and should be removed
Initializable._disableInitializers() (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#129-131) is never used and should be removed
Initializable._setInitializedVersion(uint8) (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#133-148) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version^0.8.2 (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol#4) allows old versions
Pragma version^0.8.1 (node_modules/@openzeppelin/contracts/utils/Address.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#4) allows old versions
Pragma version0.8.9 (contracts/access/Governable.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/interfaces/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 (contracts/interfaces/core/IChainlinkPriceProvider.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/interfaces/core/IPriceProvider.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/interfaces/core/IPriceProvidersAggregator.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/interfaces/periphery/ISwapperOracle.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/libraries/DataTypes.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/periphery/SwapperOracle.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) (node_modules/@openzeppelin/contracts/utils/Address.sol#60-65):
        - (success) = recipient.call{value: amount}() (node_modules/@openzeppelin/contracts/utils/Address.sol#63)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#128-139):
        - (success,returndata) = target.call{value: value}(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#137)
Low level call in Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#157-166):
        - (success,returndata) = target.staticcall(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#164)
Low level call in Address.functionDelegateCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#184-193):
        - (success,returndata) = target.delegatecall(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#191)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function Governable.__Governable_init() (contracts/access/Governable.sol#38-42) is not in mixedCase
Parameter Governable.transferGovernorship(address)._proposedGovernor (contracts/access/Governable.sol#56) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Variable IPriceProvidersAggregator.priceProviders(DataTypes.Provider)._priceProvider (contracts/interfaces/core/IPriceProvidersAggregator.sol#12) is too similar to IPriceProvidersAggregator.setPriceProvider(DataTypes.Provider,IPriceProvider).priceProvider_ (contracts/interfaces/core/IPriceProvidersAggregator.sol#55)
Variable SwapperOracle._quote(DataTypes.Provider,address,address,uint256).__amountOut (contracts/periphery/SwapperOracle.sol#119) is too similar to SwapperOracle.quote(address,address,uint256)._amountOutA (contracts/periphery/SwapperOracle.sol#77)
Variable SwapperOracle._quote(DataTypes.Provider,address,address,uint256).__amountOut (contracts/periphery/SwapperOracle.sol#119) is too similar to SwapperOracle.quote(address,address,uint256)._amountOutB (contracts/periphery/SwapperOracle.sol#87)
Variable SwapperOracle._quote(DataTypes.Provider,address,address,uint256).__lastUpdatedAt (contracts/periphery/SwapperOracle.sol#120) is too similar to SwapperOracle.quote(address,address,uint256)._lastUpdatedAtA (contracts/periphery/SwapperOracle.sol#77)
Variable SwapperOracle._quote(DataTypes.Provider,address,address,uint256).__lastUpdatedAt (contracts/periphery/SwapperOracle.sol#120) is too similar to SwapperOracle.quote(address,address,uint256)._lastUpdatedAtB (contracts/periphery/SwapperOracle.sol#87)
Variable SwapperOracle.quote(address,address,uint256)._amountOutA (contracts/periphery/SwapperOracle.sol#77) is too similar to SwapperOracle.quote(address,address,uint256)._amountOutB (contracts/periphery/SwapperOracle.sol#87)
Variable SwapperOracle.quote(address,address,uint256)._lastUpdatedAtA (contracts/periphery/SwapperOracle.sol#77) is too similar to SwapperOracle.quote(address,address,uint256)._lastUpdatedAtB (contracts/periphery/SwapperOracle.sol#87)
Variable SwapperOracle.fallbackProviderA (contracts/periphery/SwapperOracle.sol#25) is too similar to SwapperOracle.fallbackProviderB (contracts/periphery/SwapperOracle.sol#29)
Variable SwapperOracle.updateFallbackProviders(DataTypes.Provider,DataTypes.Provider).fallbackProviderA_ (contracts/periphery/SwapperOracle.sol#131) is too similar to SwapperOracle.updateFallbackProviders(DataTypes.Provider,DataTypes.Provider).fallbackProviderB_ (contracts/periphery/SwapperOracle.sol#131)
Variable SwapperOracle.constructor(IPriceProvidersAggregator,uint256,DataTypes.Provider,DataTypes.Provider).fallbackProviderA_ (contracts/periphery/SwapperOracle.sol#51) is too similar to SwapperOracle.updateFallbackProviders(DataTypes.Provider,DataTypes.Provider).fallbackProviderB_ (contracts/periphery/SwapperOracle.sol#131)
Variable SwapperOracle.constructor(IPriceProvidersAggregator,uint256,DataTypes.Provider,DataTypes.Provider).fallbackProviderA_ (contracts/periphery/SwapperOracle.sol#51) is too similar to SwapperOracle.constructor(IPriceProvidersAggregator,uint256,DataTypes.Provider,DataTypes.Provider).fallbackProviderB_ (contracts/periphery/SwapperOracle.sol#52)
Variable SwapperOracle.updateFallbackProviders(DataTypes.Provider,DataTypes.Provider).fallbackProviderA_ (contracts/periphery/SwapperOracle.sol#131) is too similar to SwapperOracle.constructor(IPriceProvidersAggregator,uint256,DataTypes.Provider,DataTypes.Provider).fallbackProviderB_ (contracts/periphery/SwapperOracle.sol#52)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-are-too-similar

updateFallbackProviders(DataTypes.Provider,DataTypes.Provider) should be declared external:
        - SwapperOracle.updateFallbackProviders(DataTypes.Provider,DataTypes.Provider) (contracts/periphery/SwapperOracle.sol#131-139)
updateProvidersAggregator(IPriceProvidersAggregator) should be declared external:
        - SwapperOracle.updateProvidersAggregator(IPriceProvidersAggregator) (contracts/periphery/SwapperOracle.sol#144-148)
updateStalePeriod(uint256) should be declared external:
        - SwapperOracle.updateStalePeriod(uint256) (contracts/periphery/SwapperOracle.sol#153-156)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

Swapper.sol

```
Swapper._getBestAmountOut(IExchange,address,address,uint256).__amountOut (contracts/swapper/Swapper.sol#199) is a local variable never initialized
Swapper._getBestAmountOut(IExchange,address,address,uint256)._path (contracts/swapper/Swapper.sol#200) is a local variable never initialized
Swapper._getBestAmountIn(IExchange,address,address,uint256).__path (contracts/swapper/Swapper.sol#181) is a local variable never initialized
Swapper._getBestAmountIn(IExchange,address,address,uint256).__amountIn (contracts/swapper/Swapper.sol#180) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

Swapper._getBestAmountIn(IExchange,address,address,uint256) (contracts/swapper/Swapper.sol#173-186) ignores return value by exchange_.getBestAmountIn(tokenIn_,tokenOut_,amountOut_) (contracts/swapper/Swapper.sol#179-185)
Swapper._getBestAmountOut(IExchange,address,address,uint256) (contracts/swapper/Swapper.sol#192-205) ignores return value by exchange_.getBestAmountOut(tokenIn_,tokenOut_,amountIn_) (contracts/swapper/Swapper.sol#198-204)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

Swapper._getBestAmountIn(IExchange,address,address,uint256) (contracts/swapper/Swapper.sol#173-186) has external calls inside a loop: exchange_.getBestAmountIn(tokenIn_,tokenOut_,amountOut_) (contracts/swapper/Swapper.sol#179-185)
Swapper._getBestAmountOut(IExchange,address,address,uint256) (contracts/swapper/Swapper.sol#192-205) has external calls inside a loop: exchange_.getBestAmountOut(tokenIn_,tokenOut_,amountIn_) (contracts/swapper/Swapper.sol#198-204)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#calls-inside-a-loop

Variable 'Swapper._getBestAmountIn(IExchange,address,address,uint256).__amountIn (contracts/swapper/Swapper.sol#180)' in Swapper._getBestAmountIn(IExchange,address,address,uint256) (contracts/swapper/Swapper.sol#173-186) potentially used
 before declaration: _amountIn = __amountIn (contracts/swapper/Swapper.sol#183)
Variable 'Swapper._getBestAmountIn(IExchange,address,address,uint256)._path (contracts/swapper/Swapper.sol#181)' in Swapper._getBestAmountIn(IExchange,address,address,uint256) (contracts/swapper/Swapper.sol#173-186) potentially used bef
ore declaration: _path = __path (contracts/swapper/Swapper.sol#184)
Variable 'Swapper._getBestAmountOut(IExchange,address,address,uint256).__amountOut (contracts/swapper/Swapper.sol#199)' in Swapper._getBestAmountOut(IExchange,address,address,uint256) (contracts/swapper/Swapper.sol#192-205) potentially u
sed before declaration: _amountOut = __amountOut (contracts/swapper/Swapper.sol#202)
Variable 'Swapper._getBestAmountOut(IExchange,address,address,uint256)._path (contracts/swapper/Swapper.sol#200)' in Swapper._getBestAmountOut(IExchange,address,address,uint256) (contracts/swapper/Swapper.sol#192-205) potentially used b
efore declaration: _path = __path (contracts/swapper/Swapper.sol#203)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables

Address.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#201-221) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/utils/Address.sol#213-216)
EnumerableSet.values(EnumerableSet.AddressSet) (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#274-283) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#278-280)
EnumerableSet.values(EnumerableSet.UintSet) (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#347-356) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#351-353)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Different versions of Solidity is used:
        - Version used: ['^0.8.9', '<=0.8.9', '^0.8.0', '^0.8.1', '^0.8.2']
        - ^0.8.2 (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#4)
        - ^0.8.1 (node_modules/@openzeppelin/contracts/utils/Address.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#4)
        - 0.8.9 (contracts/access/Governable.sol#3)
        - 0.8.9 (contracts/interfaces/IGovernable.sol#3)
        - 0.8.9 (contracts/interfaces/periphery/ISwapperOracle.sol#3)
        - <=0.8.9 (contracts/interfaces/swapper/IExchange.sol#3)
        - 0.8.9 (contracts/interfaces/swapper/ISwapper.sol#3)
        - 0.8.9 (contracts/swapper/Swapper.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

Address.functionCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#85-87) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#114-120) is never used and should be removed
Address.functionDelegateCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#176-176) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#184-193) is never used and should be removed
Address.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#147-149) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#157-166) is never used and should be removed
Address.sendValue(address,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#60-65) is never used and should be removed
Context._msgData() (node_modules/@openzeppelin/contracts/utils/Context.sol#21-23) is never used and should be removed
EnumerableSet.add(EnumerableSet.Bytes32Set,bytes32) (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#158-160) is never used and should be removed
EnumerableSet.add(EnumerableSet.UintSet,uint256) (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#297-299) is never used and should be removed
EnumerableSet.at(EnumerableSet.Bytes32Set,uint256) (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#196-198) is never used and should be removed
EnumerableSet.at(EnumerableSet.UintSet,uint256) (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#335-337) is never used and should be removed
EnumerableSet.contains(EnumerableSet.AddressSet,address) (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#241-243) is never used and should be removed
EnumerableSet.contains(EnumerableSet.Bytes32Set,bytes32) (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#175-177) is never used and should be removed
EnumerableSet.contains(EnumerableSet.UintSet,uint256) (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#314-316) is never used and should be removed
EnumerableSet.length(EnumerableSet.Bytes32Set) (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#182-184) is never used and should be removed
EnumerableSet.length(EnumerableSet.UintSet) (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#321-323) is never used and should be removed
EnumerableSet.remove(EnumerableSet.Bytes32Set,bytes32) (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#168-170) is never used and should be removed
EnumerableSet.remove(EnumerableSet.UintSet,uint256) (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#307-309) is never used and should be removed
EnumerableSet.values(EnumerableSet.Bytes32Set) (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#208-210) is never used and should be removed
EnumerableSet.values(EnumerableSet.UintSet) (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#347-356) is never used and should be removed
Governable._Governable_init() (contracts/access/Governable.sol#38-42) is never used and should be removed
Initializable._disableInitializers() (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#129-131) is never used and should be removed
Initializable._setInitializedVersion(uint8) (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#133-148) is never used and should be removed
SafeERC20.safeApprove(IERC20,address,uint256) (node_modules/@openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#45-58) is never used and should be removed
SafeERC20.safeDecreaseAllowance(IERC20,address,uint256) (node_modules/@openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#69-80) is never used and should be removed
SafeERC20.safeIncreaseAllowance(IERC20,address,uint256) (node_modules/@openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#60-67) is never used and should be removed
SafeERC20.safeTransfer(IERC20,address,uint256) (node_modules/@openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#21-27) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version^0.8.2 (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#4) allows old versions
Pragma version^0.8.1 (node_modules/@openzeppelin/contracts/utils/Address.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/structs/EnumerableSet.sol#4) allows old versions
Pragma version0.8.9 (contracts/access/Governable.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/interfaces/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 (contracts/interfaces/periphery/ISwapperOracle.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version>=0.8.9 (contracts/interfaces/swapper/IExchange.sol#3) uses a range that
Pragma version0.8.9 (contracts/interfaces/swapper/ISwapper.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/swapper/Swapper.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) (node_modules/@openzeppelin/contracts/utils/Address.sol#60-65):
        - (success) = recipient.call{value: amount}() (node_modules/@openzeppelin/contracts/utils/Address.sol#63)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#128-139):
        - (success,returndata) = target.call{value: value}(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#137)
Low level call in Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#157-166):
        - (success,returndata) = target.staticcall(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#164)
Low level call in Address.functionDelegateCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#184-193):
        - (success,returndata) = target.delegatecall(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#191)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function Governable._Governable_init() (contracts/access/Governable.sol#38-42) is not in mixedCase
Parameter Governable.transferGovernorship(address)._proposedGovernor (contracts/access/Governable.sol#56) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Variable Swapper._getBestAmountOut(IExchange,address,address,uint256).__amountOut (contracts/swapper/Swapper.sol#199) is too similar to Swapper.getBestAmountOut(address,address,uint256)._iAmountOut (contracts/swapper/Swapper.sol#113)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-are-too-similar
```

UniswapV2LikeExchange.sol

```
UniswapV2LikeExchange._getAmountsIn(uint256,address[]).amounts (contracts/swapper/UniswapV2LikeExchange.sol#152) is a local variable never initialized
UniswapV2LikeExchange._getAmountsOut(uint256,address[]).amounts (contracts/swapper/UniswapV2LikeExchange.sol#142) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

UniswapV2LikeExchange._getAmountsOut(uint256,address[]) (contracts/swapper/UniswapV2LikeExchange.sol#141-145) ignores return value by router.getAmountsOut(amountIn_,path_) (contracts/swapper/UniswapV2LikeExchange.sol#142-144)
UniswapV2LikeExchange._getAmountsIn(uint256,address[]) (contracts/swapper/UniswapV2LikeExchange.sol#151-155) ignores return value by router.getAmountsIn(_amountOut,_path) (contracts/swapper/UniswapV2LikeExchange.sol#152-154)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

UniswapV2LikeExchange.constructor(IUniswapV2Router02,address)._wethLike (contracts/swapper/UniswapV2LikeExchange.sol#36) lacks a zero-check on :
        - wethLike = wethLike_ (contracts/swapper/UniswapV2LikeExchange.sol#39)
UniswapV2LikeExchange.updateWethLikeToken(address)._wethLike_ (contracts/swapper/UniswapV2LikeExchange.sol#160) lacks a zero-check on :
        - wethLike = wethLike_ (contracts/swapper/UniswapV2LikeExchange.sol#162)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation

Variable 'UniswapV2LikeExchange._getAmountsOut(uint256,address[]).amounts (contracts/swapper/UniswapV2LikeExchange.sol#142)' in UniswapV2LikeExchange._getAmountsOut(uint256,address[]) (contracts/swapper/UniswapV2LikeExchange.sol#141-145)
 potentially used before declaration: _amountOut = amounts[path_.length - 1] (contracts/swapper/UniswapV2LikeExchange.sol#143)
Variable 'UniswapV2LikeExchange._getAmountsIn(uint256,address[]).amounts (contracts/swapper/UniswapV2LikeExchange.sol#152)' in UniswapV2LikeExchange._getAmountsIn(uint256,address[]) (contracts/swapper/UniswapV2LikeExchange.sol#151-155) p
otentially used before declaration: _amountIn = amounts[0] (contracts/swapper/UniswapV2LikeExchange.sol#153)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables

Address.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#201-221) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/utils/Address.sol#213-216)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Different versions of Solidity is used:
        - Version used: ['^0.8.9', '>=0.8.9', '>=0.6.2', '^0.8.0', '^0.8.1', '^0.8.2']
        - ^0.8.2 (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#4)
        - ^0.8.1 (node_modules/@openzeppelin/contracts/utils/Address.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#4)
        - >=0.6.2 (node_modules/@uniswap/v2-periphery/contracts/interfaces/IUniswapV2Router01.sol#1)
        - >=0.6.2 (node_modules/@uniswap/v2-periphery/contracts/interfaces/IUniswapV2Router02.sol#1)
        - 0.8.9 (contracts/access/Governable.sol#3)
        - 0.8.9 (contracts/interfaces/IGovernable.sol#3)
        - <=0.8.9 (contracts/interfaces/swapper/IExchange.sol#3)
        - 0.8.9 (contracts/swapper/UniswapV2LikeExchange.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

Address.functionCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#85-87) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#114-120) is never used and should be removed
Address.functionDelegateCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#176-176) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#184-193) is never used and should be removed
Address.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#147-149) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#157-166) is never used and should be removed
Address.sendValue(address,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#60-65) is never used and should be removed
Context._msgData() (node_modules/@openzeppelin/contracts/utils/Context.sol#21-23) is never used and should be removed
Governable._Governable_init() (contracts/access/Governable.sol#38-42) is never used and should be removed
Initializable._disableInitializers() (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#129-131) is never used and should be removed
Initializable._setInitializedVersion(uint8) (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#133-148) is never used and should be removed
SafeERC20.safeDecreaseAllowance(IERC20,address,uint256) (node_modules/@openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#69-80) is never used and should be removed
SafeERC20.safeIncreaseAllowance(IERC20,address,uint256) (node_modules/@openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#60-67) is never used and should be removed
SafeERC20.safeTransferFrom(IERC20,address,address,uint256) (node_modules/@openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#29-36) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
```

```
Pragma version^0.8.2 (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/utils/SafeERC20.sol#4) allows old versions
Pragma version^0.8.1 (node_modules/@openzeppelin/contracts/utils/Address.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#4) allows old versions
Pragma version>=0.6.2 (node_modules/@uniswap/v2-periphery/contracts/interfaces/IUniswapV2Router01.sol#1) allows old versions
Pragma version>=0.6.2 (node_modules/@uniswap/v2-periphery/contracts/interfaces/IUniswapV2Router02.sol#1) allows old versions
Pragma version0.8.8 (contracts/access/Governable.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/interfaces/IGovernable.sol#3) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version<=0.8.9 (contracts/interfaces/swapper/IExchange.sol#3) uses lesser than
Pragma version0.8.9 (contracts/swapper/UniswapV2LikeExchange.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) (node_modules/@openzeppelin/contracts/utils/Address.sol#60-65):
        - (success) = recipient.call{value: amount}() (node_modules/@openzeppelin/contracts/utils/Address.sol#63)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#128-139):
        - (success,returndata) = target.call{value: value}(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#137)
Low level call in Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#157-166):
        - (success,returndata) = target.staticcall(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#164)
Low level call in Address.functionDelegateCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#184-193):
        - (success,returndata) = target.delegatecall(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#191)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function IUniswapV2Router01.WETH() (node_modules/@uniswap/v2-periphery/contracts/interfaces/IUniswapV2Router01.sol#5) is not in mixedCase
Function Governable.__Governable_init() (contracts/access/Governable.sol#38-42) is not in mixedCase
Parameter Governable_transferGovernorship(address)._proposedGovernor (contracts/access/Governable.sol#56) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Variable IUniswapV2Router01.addLiquidity(address,address,uint256,uint256,uint256,address,uint256).amountADesired (node_modules/@uniswap/v2-periphery/contracts/interfaces/IUniswapV2Router01.sol#10) is too similar to IUniswapV2Router01.addLiquidity(address,address,uint256,uint256,uint256,address,uint256).amountBDesired (node_modules/@uniswap/v2-periphery/contracts/interfaces/IUniswapV2Router01.sol#11)
Variable UniswapV2LikeExchange.getBestAmountOut(address,address,uint256)._amountOutA (contracts/swapper/UniswapV2LikeExchange.sol#86) is too similar to UniswapV2LikeExchange.getBestAmountOut(address,address,uint256)._amountOutB (contracts/swapper/UniswapV2LikeExchange.sol#99)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-are-too-similar
```

CTokenOracle.sol

```
CTokenOracle.getPriceInUsd(IERC20) (contracts/periphery/synth/CTokenOracle.sol#35-48) performs a multiplication on the result of a division:
        -_underlyingAmount = (ONE_CTOKEN * ICToken(address(_asset)).exchangeRateStored()) / 1e18 (contracts/periphery/synth/CTokenOracle.sol#46)
        -_priceInUsd = (_underlyingPriceInUsd * _underlyingAmount) / 10 ** IERC20Metadata(_underlyingAddress).decimals() (contracts/periphery/synth/CTokenOracle.sol#47)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#divide-before-multiply

CTokenOracle.getPriceInUsd(IERC20)._underlyingAddress (contracts/periphery/synth/CTokenOracle.sol#36) is a local variable never initialized
CTokenOracle.getPriceInUsd(IERC20)._underlying (contracts/periphery/synth/CTokenOracle.sol#38) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

CTokenOracle.getPriceInUsd(IERC20) (contracts/periphery/synth/CTokenOracle.sol#35-48) ignores return value by ICToken(address(_asset)).underlying() (contracts/periphery/synth/CTokenOracle.sol#38-40)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

CTokenOracle.constructor(ISynthOracle,address)._wethLike (contracts/periphery/synth/CTokenOracle.sol#25) lacks a zero-check on :
        - wethLike = _wethLike (contracts/periphery/synth/CTokenOracle.sol#27)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation

Variable 'CTokenOracle.getPriceInUsd(IERC20)._underlying (contracts/periphery/synth/CTokenOracle.sol#38)' in CTokenOracle.getPriceInUsd(IERC20) (contracts/periphery/synth/CTokenOracle.sol#35-48) potentially used before declaration: _underlyingAddress = _underlying (contracts/periphery/synth/CTokenOracle.sol#39)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables

Address.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#201-221) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/utils/Address.sol#213-216)
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', '^0.8.1', '^0.8.2']
        - ^0.8.2 (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol#4)
        - ^0.8.1 (node_modules/@openzeppelin/contracts/utils/Address.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#4)
        - 0.8.9 (contracts/access/Governable.sol#3)
        - 0.8.9 (contracts/interfaces/IGovernable.sol#3)
        - 0.8.9 (contracts/interfaces/external/ICToken.sol#3)
        - 0.8.9 (contracts/interfaces/periphery/synth/ISynthOracle.sol#3)
        - 0.8.9 (contracts/periphery/synth/CTokenOracle.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

Address.functionCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#85-87) is never used and should be removed
Address.functionCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#95-101) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#114-120) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#128-139) is never used and should be removed
Address.functionDelegateCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#174-176) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#184-193) is never used and should be removed
Address.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#147-149) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#157-166) is never used and should be removed
Address.isContract(address) (node_modules/@openzeppelin/contracts/utils/Address.sol#36-42) is never used and should be removed
Address.sendValue(address,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#60-65) is never used and should be removed
Address.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#201-221) is never used and should be removed
Context._msgData() (node_modules/@openzeppelin/contracts/utils/Context.sol#21-23) is never used and should be removed
Governable.__Governable_init() (contracts/access/Governable.sol#38-42) is never used and should be removed
Initializable._disableInitializers() (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#129-131) is never used and should be removed
Initializable._setInitializedVersion(uint8) (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#133-148) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version^0.8.2 (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol#4) allows old versions
Pragma version^0.8.1 (node_modules/@openzeppelin/contracts/utils/Address.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#4) allows old versions
Pragma version0.8.9 (contracts/access/Governable.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/interfaces/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 (contracts/interfaces/external/ICToken.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/interfaces/periphery/synth/ISynthOracle.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/periphery/synth/CTokenOracle.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) (node_modules/@openzeppelin/contracts/utils/Address.sol#60-65):
        - (success) = recipient.call{value: amount}() (node_modules/@openzeppelin/contracts/utils/Address.sol#63)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#128-139):
        - (success,returndata) = target.call{value: value}(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#137)
Low level call in Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#157-166):
        - (success,returndata) = target.staticcall(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#164)
Low level call in Address.functionDelegateCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#184-193):
        - (success,returndata) = target.delegatecall(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#191)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function Governable.__Governable_init() (contracts/access/Governable.sol#38-42) is not in mixedCase
Parameter Governable_transferGovernorship(address)._proposedGovernor (contracts/access/Governable.sol#56) is not in mixedCase
Parameter CTokenOracle.getPriceInUsd(IERC20)._asset (contracts/periphery/synth/CTokenOracle.sol#35) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
```

CurveLpTokenOracle.sol

```
Address.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#201-221) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/utils/Address.sol#213-216)
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', '^0.8.1', '^0.8.2']
        - ^0.8.2 (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4)
        - ^0.8.1 (node_modules/@openzeppelin/contracts/utils/Address.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#4)
        - 0.8.9 (contracts/access/Governable.sol#3)
        - 0.8.9 (contracts/interfaces/IGovernable.sol#3)
        - 0.8.9 (contracts/interfaces/external/ICurveAddressProvider.sol#3)
        - 0.8.9 (contracts/interfaces/external/ICurveRegistry.sol#3)
        - 0.8.9 (contracts/interfaces/periphery/synth/ISynthOracle.sol#3)
        - 0.8.9 (contracts/periphery/synth/CurveLpTokenOracle.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

Address.functionCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#85-87) is never used and should be removed
Address.functionCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#95-101) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#114-120) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#128-139) is never used and should be removed
Address.functionDelegateCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#174-176) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#184-193) is never used and should be removed
Address.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#147-149) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#157-166) is never used and should be removed
Address.isContract(address) (node_modules/@openzeppelin/contracts/utils/Address.sol#36-42) is never used and should be removed
Address.sendValue(address,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#60-65) is never used and should be removed
Address.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#201-221) is never used and should be removed
Context._msgData() (node_modules/@openzeppelin/contracts/utils/Context.sol#21-23) is never used and should be removed
Governable.__Governable_init() (contracts/access/Governable.sol#38-42) is never used and should be removed
Initializable._disableInitializers() (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#129-131) is never used and should be removed
Initializable._setInitializedVersion(uint8) (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#133-148) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version^0.8.2 (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4) allows old versions
Pragma version^0.8.1 (node_modules/@openzeppelin/contracts/utils/Address.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#4) allows old versions
Pragma version0.8.9 (contracts/access/Governable.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/interfaces/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 (contracts/interfaces/external/ICurveAddressProvider.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/interfaces/external/ICurveRegistry.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/interfaces/periphery/synth/ISynthOracle.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/periphery/synth/CurveLpTokenOracle.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) (node_modules/@openzeppelin/contracts/utils/Address.sol#60-65):
        - (success) = recipient.call{value: amount}() (node_modules/@openzeppelin/contracts/utils/Address.sol#63)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#128-139):
        - (success,returndata) = target.call{value: value}(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#137)
Low level call in Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#157-166):
        - (success,returndata) = target.staticcall(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#164)
Low level call in Address.functionDelegateCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#184-193):
        - (success,returndata) = target.delegatecall(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#191)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function Governable.__Governable_init() (contracts/access/Governable.sol#38-42) is not in mixedCase
Parameter Governable.transferGovernorship(address)._proposedGovernor (contracts/access/Governable.sol#56) is not in mixedCase
Function ICurveAddressProvider.get_registry() (contracts/interfaces/external/ICurveAddressProvider.sol#8) is not in mixedCase
Function ICurveRegistry.get_virtual_price_from_lp_token(address) (contracts/interfaces/external/ICurveRegistry.sol#6) is not in mixedCase
Parameter CurveLpTokenOracle.getPriceInUsd(IERC20)._asset (contracts/periphery/synth/CurveLpTokenOracle.sol#24) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
```

SynthDefaultOracle.sol

```
SynthDefaultOracle.getPriceInUsd(IERC20) (contracts/periphery/synth/SynthDefaultOracle.sol#85-94) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(_assetData.stalePeriod > block.timestamp - _lastUpdatedAt,price-is-stale) (contracts/periphery/synth/SynthDefaultOracle.sol#93)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

Address.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#201-221) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/utils/Address.sol#213-216)
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', '^0.8.1', '^0.8.2']
        - ^0.8.2 (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol#4)
        - ^0.8.1 (node_modules/@openzeppelin/contracts/utils/Address.sol#4)
        - ^0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#4)
        - 0.8.9 (contracts/access/Governable.sol#3)
        - 0.8.9 (contracts/interfaces/IGovernable.sol#3)
        - 0.8.9 (contracts/interfaces/core/IChainlinkPriceProvider.sol#3)
        - 0.8.9 (contracts/interfaces/core/IPriceProvider.sol#3)
        - 0.8.9 (contracts/interfaces/periphery/synth/ISynthOracle.sol#3)
        - 0.8.9 (contracts/periphery/synth/SynthDefaultOracle.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

Address.functionCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#85-87) is never used and should be removed
Address.functionCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#95-101) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#114-120) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#123-139) is never used and should be removed
Address.functionDelegateCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#174-176) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#184-193) is never used and should be removed
Address.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#147-149) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#157-166) is never used and should be removed
Address.isContract(address) (node_modules/@openzeppelin/contracts/utils/Address.sol#36-42) is never used and should be removed
Address.sendValue(address,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#60-65) is never used and should be removed
Address.verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#201-221) is never used and should be removed
Context._msgData() (node_modules/@openzeppelin/contracts/utils/Context.sol#21-23) is never used and should be removed
Governable.__Governable_init() (contracts/access/Governable.sol#38-42) is never used and should be removed
Initializable._disableInitializers() (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#129-131) is never used and should be removed
Initializable._setInitializedVersion(uint8) (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#133-148) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version^0.8.2 (node_modules/@openzeppelin/contracts/proxy/utils/Initializable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/IERC20.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC20/extensions/IERC20Metadata.sol#4) allows old versions
Pragma version^0.8.1 (node_modules/@openzeppelin/contracts/utils/Address.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#4) allows old versions
Pragma version0.8.9 (contracts/access/Governable.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/interfaces/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 (contracts/interfaces/core/IChainlinkPriceProvider.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/interfaces/core/IPriceProvider.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/interfaces/periphery/synth/ISynthOracle.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/periphery/synth/SynthDefaultOracle.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) (node_modules/@openzeppelin/contracts/utils/Address.sol#60-65):
        - (success) = recipient.call{value: amount}() (node_modules/@openzeppelin/contracts/utils/Address.sol#63)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#128-139):
        - (success,returndata) = target.call{value: value}(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#137)
Low level call in Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#157-166):
        - (success,returndata) = target.staticcall(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#164)
Low level call in Address.functionDelegateCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#184-193):
        - (success,returndata) = target.delegatecall(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#191)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function Governable.__Governable_init() (contracts/access/Governable.sol#38-42) is not in mixedCase
Parameter Governable.transferGovernorship(address)._proposedGovernor (contracts/access/Governable.sol#56) is not in mixedCase
Parameter SynthDefaultOracle.addOrUpdateUsdAsset(IERC20)._asset (contracts/periphery/synth/SynthDefaultOracle.sol#66) is not in mixedCase
Parameter SynthDefaultOracle.addOrUpdateAssetThatUsesChainlink(IERC20,address,uint256)._asset (contracts/periphery/synth/SynthDefaultOracle.sol#77) is not in mixedCase
Parameter SynthDefaultOracle.addOrUpdateAssetThatUsesChainlink(IERC20,address,uint256)._underlyingAsset (contracts/periphery/synth/SynthDefaultOracle.sol#78) is not in mixedCase
Parameter SynthDefaultOracle.addOrUpdateAssetThatUsesChainlink(IERC20,address,uint256)._stalePeriod (contracts/periphery/synth/SynthDefaultOracle.sol#79) is not in mixedCase
Parameter SynthDefaultOracle.getPriceInUsd(IERC20)._asset (contracts/periphery/synth/SynthDefaultOracle.sol#85) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
```

UniswapV3CrossPoolOracle.sol

```
FullMath.mulDiv(uint256,uint256,uint256) (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#14-106) performs a multiplication on the result of a division:
        -denominator = denominator / twos (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#67)
        -inv = (3 * denominator) ^ 2 (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#87)
FullMath.mulDiv(uint256,uint256,uint256) (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#14-106) performs a multiplication on the result of a division:
        -denominator = denominator / twos (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#67)
        -inv *= 2 - denominator * inv (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#88)
FullMath.mulDiv(uint256,uint256,uint256) (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#14-106) performs a multiplication on the result of a division:
        -denominator = denominator / twos (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#67)
        -inv *= 2 - denominator * inv (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#89)
FullMath.mulDiv(uint256,uint256,uint256) (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#14-106) performs a multiplication on the result of a division:
        -denominator = denominator / twos (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#67)
        -inv *= 2 - denominator * inv (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#90)
FullMath.mulDiv(uint256,uint256,uint256) (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#14-106) performs a multiplication on the result of a division:
        -denominator = denominator / twos (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#67)
        -inv *= 2 - denominator * inv (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#91)
FullMath.mulDiv(uint256,uint256,uint256) (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#14-106) performs a multiplication on the result of a division:
        -denominator = denominator / twos (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#67)
        -inv *= 2 - denominator * inv (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#96)
FullMath.mulDiv(uint256,uint256,uint256) (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#14-106) performs a multiplication on the result of a division:
        -prod0 = prod0 / twos (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#72)
        -result = prod0 * inv (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#104)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#divide-before-multiply

OracleLibrary.getWeightedArithmeticMeanTick(OracleLibrary.WeightedTickData[]).i (node_modules/@uniswap/v3-periphery/contracts/libraries/OracleLibrary.sol#152) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

UniswapV3CrossPoolOracle.constructor(address).nativeToken_ (contracts/utils/UniswapV3CrossPoolOracle.sol#18) lacks a zero-check on :
        - nativeToken = nativeToken_ (contracts/utils/UniswapV3CrossPoolOracle.sol#19)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation

OracleLibrary.getBlockStartingTickAndLiquidity(address) (node_modules/@uniswap/v3-periphery/contracts/libraries/OracleLibrary.sol#93-126) uses timestamp for comparisons
        Dangerous comparisons:
        - observationTimestamp != uint32(block.timestamp) (node_modules/@uniswap/v3-periphery/contracts/libraries/OracleLibrary.sol#104)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

FullMath.mulDiv(uint256,uint256,uint256) (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#14-106) uses assembly
        - INLINE ASM (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#26-30)
        - INLINE ASM (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#35-37)
        - INLINE ASM (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#52-54)
        - INLINE ASM (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#56-59)
        - INLINE ASM (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#66-68)
        - INLINE ASM (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#71-73)
        - INLINE ASM (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#77-79)
```

AUTOMATED TESTING

```
TickMath.getTickAtSqrtRatio(uint160) (node_modules/@uniswap/v3-core/contracts/libraries/TickMath.sol#61-204) uses assembly
        - INLINE ASM (node_modules/@uniswap/v3-core/contracts/libraries/TickMath.sol#69-73)
        - INLINE ASM (node_modules/@uniswap/v3-core/contracts/libraries/TickMath.sol#74-78)
        - INLINE ASM (node_modules/@uniswap/v3-core/contracts/libraries/TickMath.sol#79-83)
        - INLINE ASM (node_modules/@uniswap/v3-core/contracts/libraries/TickMath.sol#84-88)
        - INLINE ASM (node_modules/@uniswap/v3-core/contracts/libraries/TickMath.sol#89-93)
        - INLINE ASM (node_modules/@uniswap/v3-core/contracts/libraries/TickMath.sol#94-98)
        - INLINE ASM (node_modules/@uniswap/v3-core/contracts/libraries/TickMath.sol#99-103)
        - INLINE ASM (node_modules/@uniswap/v3-core/contracts/libraries/TickMath.sol#104-107)
        - INLINE ASM (node_modules/@uniswap/v3-core/contracts/libraries/TickMath.sol#114-119)
        - INLINE ASM (node_modules/@uniswap/v3-core/contracts/libraries/TickMath.sol#120-125)
        - INLINE ASM (node_modules/@uniswap/v3-core/contracts/libraries/TickMath.sol#126-131)
        - INLINE ASM (node_modules/@uniswap/v3-core/contracts/libraries/TickMath.sol#132-137)
        - INLINE ASM (node_modules/@uniswap/v3-core/contracts/libraries/TickMath.sol#138-143)
        - INLINE ASM (node_modules/@uniswap/v3-core/contracts/libraries/TickMath.sol#144-149)
        - INLINE ASM (node_modules/@uniswap/v3-core/contracts/libraries/TickMath.sol#150-155)
        - INLINE ASM (node_modules/@uniswap/v3-core/contracts/libraries/TickMath.sol#156-161)
        - INLINE ASM (node_modules/@uniswap/v3-core/contracts/libraries/TickMath.sol#162-167)
        - INLINE ASM (node_modules/@uniswap/v3-core/contracts/libraries/TickMath.sol#168-173)
        - INLINE ASM (node_modules/@uniswap/v3-core/contracts/libraries/TickMath.sol#174-179)
        - INLINE ASM (node_modules/@uniswap/v3-core/contracts/libraries/TickMath.sol#180-185)
        - INLINE ASM (node_modules/@uniswap/v3-core/contracts/libraries/TickMath.sol#186-191)
        - INLINE ASM (node_modules/@uniswap/v3-core/contracts/libraries/TickMath.sol#192-196)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Different versions of Solidity is used:
        - Version used: ['0.7.6', '>=0.4.0', '>=0.5.0', '>=0.5.0<0.8.0', '>=0.7.6']
        - >=0.5.0 (node_modules/@uniswap/v3-core/contracts/interfaces/IUniswapV3Pool.sol#2)
        - >=0.5.0 (node_modules/@uniswap/v3-core/contracts/interfaces/pool/IUniswapV3PoolActions.sol#2)
        - >=0.5.0 (node_modules/@uniswap/v3-core/contracts/interfaces/pool/IUniswapV3PoolDerivedState.sol#2)
        - >=0.5.0 (node_modules/@uniswap/v3-core/contracts/interfaces/pool/IUniswapV3PoolEvents.sol#2)
        - >=0.5.0 (node_modules/@uniswap/v3-core/contracts/interfaces/pool/IUniswapV3PoolImmutables.sol#2)
        - >=0.5.0 (node_modules/@uniswap/v3-core/contracts/interfaces/pool/IUniswapV3PoolOwnerActions.sol#2)
        - >=0.5.0 (node_modules/@uniswap/v3-core/contracts/interfaces/pool/IUniswapV3PoolState.sol#2)
        - >=0.4.0 (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#2)
        - >=0.5.0 (node_modules/@uniswap/v3-core/contracts/libraries/TickMath.sol#2)
        - >=0.5.0<0.8.0 (node_modules/@uniswap/v3-periphery/contracts/libraries/OracleLibrary.sol#2)
        - >=0.5.0 (node_modules/@uniswap/v3-periphery/contracts/libraries/PoolAddress.sol#2)
        - >=0.7.6 (contracts/libraries/SafeUint128.sol#3)
        - 0.7.6 (contracts/utils/UniswapV3CrossPoolOracle.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

FullMath.mulDivRoundingUp(uint256,uint256,uint256) (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#113-123) is never used and should be removed
OracleLibrary.getBlockStartingTickAndLiquidity(address) (node_modules/@uniswap/v3-periphery/contracts/libraries/OracleLibrary.sol#93-126) is never used and should be removed
OracleLibrary.getChainedPrice(address[],int24[]) (node_modules/@uniswap/v3-periphery/contracts/libraries/OracleLibrary.sol#168-179) is never used and should be removed
OracleLibrary.getOldestObservationSecondsAgo(address) (node_modules/@uniswap/v3-periphery/contracts/libraries/OracleLibrary.sol#74-88) is never used and should be removed
OracleLibrary.getWeightedArithmeticMeanTick(OracleLibrary.WeightedTickData[]) (node_modules/@uniswap/v3-periphery/contracts/libraries/OracleLibrary.sol#140-160) is never used and should be removed
TickMath.getTickAtSqrtRatio(uint160) (node_modules/@uniswap/v3-core/contracts/libraries/TickMath.sol#61-204) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version>=0.5.0 (node_modules/@uniswap/v3-core/contracts/interfaces/IUniswapV3Pool.sol#2) allows old versions
Pragma version>=0.5.0 (node_modules/@uniswap/v3-core/contracts/interfaces/pool/IUniswapV3PoolActions.sol#2) allows old versions
Pragma version>=0.5.0 (node_modules/@uniswap/v3-core/contracts/interfaces/pool/IUniswapV3PoolDerivedState.sol#2) allows old versions
Pragma version>=0.5.0 (node_modules/@uniswap/v3-core/contracts/interfaces/pool/IUniswapV3PoolEvents.sol#2) allows old versions
Pragma version>=0.5.0 (node_modules/@uniswap/v3-core/contracts/interfaces/pool/IUniswapV3PoolImmutables.sol#2) allows old versions
Pragma version>=0.5.0 (node_modules/@uniswap/v3-core/contracts/interfaces/pool/IUniswapV3PoolOwnerActions.sol#2) allows old versions
Pragma version>=0.5.0 (node_modules/@uniswap/v3-core/contracts/interfaces/pool/IUniswapV3PoolState.sol#2) allows old versions
Pragma version>=0.4.0 (node_modules/@uniswap/v3-core/contracts/libraries/FullMath.sol#2) allows old versions
Pragma version>=0.5.0 (node_modules/@uniswap/v3-core/contracts/libraries/TickMath.sol#2) allows old versions
Pragma version>=0.5.0<0.8.0 (node_modules/@uniswap/v3-periphery/contracts/libraries/OracleLibrary.sol#2) is too complex
Pragma version>=0.5.0 (node_modules/@uniswap/v3-periphery/contracts/libraries/PoolAddress.sol#2) allows old versions
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Parameter UniswapV3CrossPoolOracle.ethToAsset(uint256,address,uint24,uint32)._ethAmountIn (contracts/utils/UniswapV3CrossPoolOracle.sol#32) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Variable IUniswapV3PoolActions.collect(address,int24,int24,uint128,uint128).amount0Requested (node_modules/@uniswap/v3-core/contracts/interfaces/pool/IUniswapV3PoolActions.sol#47) is too similar to IUniswapV3PoolActions.collect(address,i
nt24,int24,uint128,uint128).amount1Requested (node_modules/@uniswap/v3-core/contracts/interfaces/pool/IUniswapV3PoolActions.sol#48)
Variable IUniswapV3PoolOwnerActions.collectProtocol(address,uint128,uint128).amount0Requested (node_modules/@uniswap/v3-core/contracts/interfaces/pool/IUniswapV3PoolOwnerActions.sol#20) is too similar to IUniswapV3PoolOwnerActions.collec
tProtocol(address,uint128,uint128).amount1Requested (node_modules/@uniswap/v3-core/contracts/interfaces/pool/IUniswapV3PoolOwnerActions.sol#21)
Variable IUniswapV3PoolActions.collect(address,int24,int24,uint128,uint128).amount0Requested (node_modules/@uniswap/v3-core/contracts/interfaces/pool/IUniswapV3PoolActions.sol#47) is too similar to IUniswapV3PoolOwnerActions.collectProto
col(address,uint128,uint128).amount1Requested (node_modules/@uniswap/v3-core/contracts/interfaces/pool/IUniswapV3PoolOwnerActions.sol#21)
Variable IUniswapV3PoolOwnerActions.collectProtocol(address,uint128,uint128).amount0Requested (node_modules/@uniswap/v3-core/contracts/interfaces/pool/IUniswapV3PoolOwnerActions.sol#20) is too similar to IUniswapV3PoolActions.collect(add
ress,int24,int24,uint128,uint128).amount1Requested (node_modules/@uniswap/v3-core/contracts/interfaces/pool/IUniswapV3PoolActions.sol#48)
Variable IUniswapV3PoolState.positions(bytes32).feeGrowthInside0LastX128 (node_modules/@uniswap/v3-core/contracts/interfaces/pool/IUniswapV3PoolState.sol#93) is too similar to IUniswapV3PoolState.positions(bytes32).feeGrowthInside1LastX1
28 (node_modules/@uniswap/v3-core/contracts/interfaces/pool/IUniswapV3PoolState.sol#94)
Variable IUniswapV3PoolState.ticks(int24).feeGrowthOutside0X128 (node_modules/@uniswap/v3-core/contracts/interfaces/pool/IUniswapV3PoolState.sol#70) is too similar to IUniswapV3PoolState.ticks(int24).feeGrowthOutside1X128 (node_modules/@
uniswap/v3-core/contracts/interfaces/pool/IUniswapV3PoolState.sol#71)
Variable IUniswapV3PoolOwnerActions.setFeeProtocol(uint8,uint8).feeProtocol0 (node_modules/@uniswap/v3-core/contracts/interfaces/pool/IUniswapV3PoolOwnerActions.sol#10) is too similar to IUniswapV3PoolOwnerActions.setFeeProtocol(uint8,ui
nt8).feeProtocol1 (node_modules/@uniswap/v3-core/contracts/interfaces/pool/IUniswapV3PoolOwnerActions.sol#10)
Variable IUniswapV3PoolState.positions(bytes32).tokensOwed0 (node_modules/@uniswap/v3-core/contracts/interfaces/pool/IUniswapV3PoolState.sol#95) is too similar to IUniswapV3PoolState.positions(bytes32).tokensOwed1 (node_modules/@uniswap/
v3-core/contracts/interfaces/pool/IUniswapV3PoolState.sol#96)
Variable UniswapV3CrossPoolOracle.assetToAssetThruRoute(address,uint256,address,uint32,address,uint24[])._routeThruToken (contracts/utils/UniswapV3CrossPoolOracle.sol#68) is too similar to UniswapV3CrossPoolOracle.assetToAssetThruRoute(
address,uint256,address,uint32,address,uint24[]).routeThruToken_ (contracts/utils/UniswapV3CrossPoolOracle.sol#62)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-are-too-similar

TickMath.getSqrtRatioAtTick(int24) (node_modules/@uniswap/v3-core/contracts/libraries/TickMath.sol#23-54) uses literals with too many digits:
        - ratio = 0x100000000000000000000000000000000 (node_modules/@uniswap/v3-core/contracts/libraries/TickMath.sol#27)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits

assetToAssetThruRoute(address,uint256,address,uint32,address,uint24[]) should be declared external:
        - UniswapV3CrossPoolOracle.assetToAssetThruRoute(address,uint256,address,uint32,address,uint24[]) (contracts/utils/UniswapV3CrossPoolOracle.sol#57-83)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

- Some unused returns were correctly flagged by Slither.
- The weak PRNG flagged by Slither is a false positive.
- 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:

Governable.sol

No issues found by MythX.

ChainlinkArbitrumPriceProvider.sol

Report for contracts/core/ChainlinkArbitrumPriceProvider.sol
https://dashboard.mythx.io/#/console/analyses/d22980c1-95c3-4c59-9a67-4e83ffa9f38d

Line	SWC Title	Severity	Short Description
10	(SWC-123) Requirement Violation	Low	Requirement violation.

ChainlinkAvalanchePriceProvider.sol

Report for contracts/core/ChainlinkAvalanchePriceProvider.sol
https://dashboard.mythx.io/#/console/analyses/d690ffbe-f5dd-417b-8f45-9218d72d0957

Line	SWC Title	Severity	Short Description
10	(SWC-123) Requirement Violation	Low	Requirement violation.

ChainlinkMainnetPriceProvider.sol

No issues found by MythX.

ChainlinkPolygonPriceProvider.sol

Report for contracts/core/ChainlinkPolygonPriceProvider.sol
https://dashboard.mythx.io/#/console/analyses/fb18196d-c62a-442a-bfba-8db934f64646

Line	SWC Title	Severity	Short Description
10	(SWC-123) Requirement Violation	Low	Requirement violation.

ChainlinkPriceProvider.sol

Report for contracts/core/ChainlinkPriceProvider.sol
https://dashboard.mythx.io/#/console/analyses/3df59a56-9594-47af-a044-4d7800c70187

Line	SWC Title	Severity	Short Description
16	(SWC-123) Requirement Violation	Low	Requirement violation.
86	(SWC-123) Requirement Violation	Low	Requirement violation.

PriceProvidersAggregator.sol

No issues found by MythX.

UniswapV2LikePriceProvider.sol

Report for contracts/core/UniswapV2LikePriceProvider.sol
https://dashboard.mythx.io/#/console/analyses/c399d6c2-2540-4ef4-9def-96207b0ecab3

Line	SWC Title	Severity	Short Description
199	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
206	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
206	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
209	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
209	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered

UniswapV3PriceProvider.sol

No issues found by MythX.

Oracle.sol

No issues found by MythX.

SwapperOracle.sol

Report for contracts/periphery/SwapperOracle.sol
https://dashboard.mythx.io/#/console/analyses/61c53b2b-714d-4360-80d8-205fc0453e51

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

Swapper.sol

Report for contracts/swapper/Swapper.sol
https://dashboard.mythx.io/#/console/analyses/d66039ec-898c-4a17-b3bf-d290a3b39ec9

Line	SWC Title	Severity	Short Description
73	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
90	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered
90	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "+" discovered
90	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
111	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "++" discovered
128	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "*" discovered
128	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "-" discovered
128	(SWC-101) Integer Overflow and Underflow	Unknown	Arithmetic operation "/" discovered

UniswapV2LikeExchange.sol

Report for contracts/swapper/UniswapV2LikeExchange.sol
https://dashboard.mythx.io/#/console/analyses/0e1bb03e-5302-4b4e-bc9e-58ba657dc87b

Line	SWC Title	Severity	Short Description
14	(SWC-123) Requirement Violation	Low	Requirement violation.

CTokenOracle.sol

Report for contracts/periphery/synth/CTokenOracle.sol
https://dashboard.mythx.io/#/console/analyses/40cb84a4-395e-4b66-8226-5f3ba1679b46

Line	SWC Title	Severity	Short Description
13	(SWC-123) Requirement Violation	Low	Requirement violation.
38	(SWC-123) Requirement Violation	Low	Requirement violation.

CurveLpTokenOracle.sol

No issues found by MythX.

SynthDefaultOracle.sol

Report for contracts/periphery/synth/SynthDefaultOracle.sol
https://dashboard.mythx.io/#/console/analyses/0517de04-ed66-4b35-8ddd-7d62dfde5e57

Line	SWC Title	Severity	Short Description
93	(SWC-116) Timestamp Dependence	Low	A control flow decision is made based on The block.timestamp environment variable.

UniswapV3CrossPoolOracle.sol

No issues found by MythX.

- 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