QuillAudits

Audit Report
July, 2022

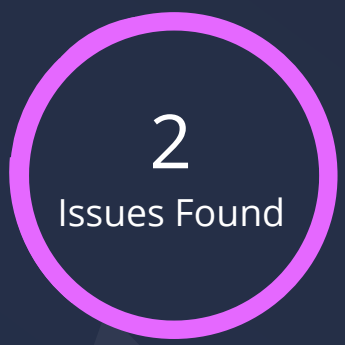
For

NSKSWAP

Table of Content

Executive Summary

Project Name	NSKSWAP
Timeline	June 13th, 2022 to 25th July, 2022
Method	Manual Review, Functional Testing, Automated Testing etc.
Scope of Audit	The scope of this audit was to analyse NSKSWAP codebase for quality, security, and correctness.
Code Base	*https://bscscan.com/address/0x51540AF4E9AD6AF937C188e532c8D0807aa0aFA4#code*



High Medium

Low Informational

	High	Medium	Low	Informational
Open Issues	0	0	0	0
Acknowledged Issues	0	0	0	2
Partially Resolved Issues	0	0	0	0
Resolved Issues	0	0	0	0

Types of Severities

High

A high severity issue or vulnerability means that your smart contract can be exploited. Issues on this level are critical to the smart contract's performance or functionality, and we recommend these issues be fixed before moving to a live environment.

Medium

The issues marked as medium severity usually arise because of errors and deficiencies in the smart contract code. Issues on this level could potentially bring problems, and they should still be fixed.

Low

Low-level severity issues can cause minor impact and or are just warnings that can remain unfixed for now. It would be better to fix these issues at some point in the future.

Informational

These are severity issues that indicate an improvement request, a general question, a cosmetic or documentation error, or a request for information. There is low-to-no impact.

Types of Issues

Open

Security vulnerabilities identified that must be resolved and are currently unresolved.

Resolved

These are the issues identified in the initial audit and have been successfully fixed.

Acknowledged

Vulnerabilities which have been acknowledged but are yet to be resolved.

Partially Resolved

Considerable efforts have been invested to reduce the risk/impact of the security issue, but are not completely resolved.

Checked Vulnerabilities

- Re-entrancy
- Timestamp Dependence
- Gas Limit and Loops
- Exception Disorder
- Gasless Send
- Use of tx.origin
- Compiler version not fixed
- Address hardcoded
- Divide before multiply
- Integer overflow/underflow
- Dangerous strict equalities
- Tautology or contradiction
- Return values of low-level calls
- Missing Zero Address Validation
- Private modifier
- Revert/require functions
- Using block.timestamp
- Multiple Sends
- Using SHA3
- Using suicide
- Using throw
- Using inline assembly

Techniques and Methods

Throughout the audit of smart contract, care was taken to ensure:

- The overall quality of code.
- Use of best practices.
- Code documentation and comments match logic and expected behaviour.
- Token distribution and calculations are as per the intended behaviour mentioned in the whitepaper.
- Implementation of ERC-20 token standards.
- Efficient use of gas.
- Code is safe from re-entrancy and other vulnerabilities.

The following techniques, methods and tools were used to review all the smart contracts.

Structural Analysis

In this step, we have analysed the design patterns and structure of smart contracts. A thorough check was done to ensure the smart contract is structured in a way that will not result in future problems.

Static Analysis

Static analysis of smart contracts was done to identify contract vulnerabilities. In this step, a series of automated tools are used to test the security of smart contracts.

Code Review / Manual Analysis

Manual analysis or review of code was done to identify new vulnerabilities or verify the vulnerabilities found during the static analysis. Contracts were completely manually analysed, their logic was checked and compared with the one described in the whitepaper. Besides, the results of the automated analysis were manually verified.

Gas Consumption

In this step, we have checked the behaviour of smart contracts in production. Checks were done to know how much gas gets consumed and the possibilities of optimization of code to reduce gas consumption.

Tools and Platforms used for Audit

Remix IDE, Truffle, Truffle Team, Solhint, Mythril, Slither, Solidity statistic analysis.

Manual Testing

High Severity Issues

No Issues Found

Medium Severity Issues

No Issues Found

Low Severity Issues

No Issues Found

Informational Issues

A.1 Comments not follow as per Netspac

Description

Contract - Tick.sol
In the library the input parameter @param tickCumulative comment is missing in update function.

Remediation

Include the @param tickCumulative in the comments.

Status

Acknowledged

A.2 Pragma Unlocked

Description

The pragma versions used in the contract are not locked. Consider
using the latest versions among 0.8.13 for deploying the contracts and libraries as it does not compile for any other version and can be confusing for a developer. Solidity source files indicate the versions of the compiler they can be compiled with.

pragma solidity ^0.8.0; // bad: compiles between 0.8.0 and 0.8.13
pragma solidity 0.8.0; // good : compiles w 0.8.0 only but not the latest version
pragma solidity 0.8.13; // best: compiles w 0.8.13

Remediation

Fix the solidity version by removing the caret symbol and use the best compiles method(specified version numbers).

Status

Acknowledged

Automated Tests

Slither

```
Compilation warnings/errors on ./UniswapV3Factory.sol:
Warning: Contract code size exceeds 24576 bytes (a limit introduced in Spurious Dragon). This contract may not be deployable on mainnet. Consider enabling the optimizer (with a low "runs" value!), turning off reve
t strings, or using libraries.
  --> ./UniswapV3Pool.sol:30:1:
   |
30 | contract UniswapV3Pool is IUniswapV3Pool, NoDelegateCall {
   | ^ (Relevant source part starts here and spans across multiple lines).

Warning: Contract code size exceeds 24576 bytes (a limit introduced in Spurious Dragon). This contract may not be deployable on mainnet. Consider enabling the optimizer (with a low "runs" value!), turning off reve
t strings, or using libraries.
  --> ./UniswapV3Factory.sol:13:1:
   |
13 | contract UniswapV3Factory is IUniswapV3Factory, UniswapV3PoolDeployer, NoDelegateCall {
   | ^ (Relevant source part starts here and spans across multiple lines).


Compilation warnings/errors on ./UniswapV3Pool.sol:
Warning: Contract code size exceeds 24576 bytes (a limit introduced in Spurious Dragon). This contract may not be deployable on mainnet. Consider enabling the optimizer (with a low "runs" value!), turning off reve
t strings, or using libraries.
  --> ./UniswapV3Pool.sol:30:1:
   |
30 | contract UniswapV3Pool is IUniswapV3Pool, NoDelegateCall {
   | ^ (Relevant source part starts here and spans across multiple lines).


Compilation warnings/errors on ./UniswapV3PoolDeployer.sol:
Warning: Contract code size exceeds 24576 bytes (a limit introduced in Spurious Dragon). This contract may not be deployable on mainnet. Consider enabling the optimizer (with a low "runs" value!), turning off reve
t strings, or using libraries.
  --> ./UniswapV3Pool.sol:30:1:
   |
30 | contract UniswapV3Pool is IUniswapV3Pool, NoDelegateCall {
   | ^ (Relevant source part starts here and spans across multiple lines).



NoDelegateCall.checkNotDelegateCall() (NoDelegateCall.sol#18-20) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
```

```
Reentrancy in UniswapV3Pool.collectProtocol(address,uint128,uint128) (UniswapV3Pool.sol#848-868):
        External calls:
        - TransferHelper.safeTransfer(token0,recipient,amount0) (UniswapV3Pool.sol#859)
        State variables written after the call(s):
        - protocolFees.token1 -= amount1 (UniswapV3Pool.sol#863)
Reentrancy in UniswapV3Pool.increaseObservationCardinalityNext(uint16) (UniswapV3Pool.sol#255-267):
        External calls:
        - observationCardinalityNextNew = observations.grow(observationCardinalityNextOld,observationCardinalityNext) (UniswapV3Pool.sol#262-263)
        State variables written after the call(s):
        - slot0.observationCardinalityNext = observationCardinalityNextNew (UniswapV3Pool.sol#264)
Reentrancy in UniswapV3Pool.initialize(uint160) (UniswapV3Pool.sol#271-289):
        External calls:
        - (cardinality,cardinalityNext) = observations.initialize(_blockTimestamp()) (UniswapV3Pool.sol#276)
        State variables written after the call(s):
        - slot0 = Slot0(sqrtPriceX96,tick,0,cardinality,cardinalityNext,0,true) (UniswapV3Pool.sol#278-286)
Reentrancy in UniswapV3Pool.swap(address,bool,int256,uint160,bytes) (UniswapV3Pool.sol#596-788):
        External calls:
        - (cache.tickCumulative,cache.secondsPerLiquidityCumulativeX128) = observations.observeSingle(cache.blockTimestamp,0,slot0Start.tick,slot0Start.observationIndex,cache.liquidityStart,slot0Start.observationCa
rdinality) (UniswapV3Pool.sol#699-706)
        - (observationIndex,observationCardinality) = observations.write(slot0Start.observationIndex,cache.blockTimestamp,slot0Start.tick,cache.liquidityStart,slot0Start.observationCardinality,slot0Start.observatio
nCardinalityNext) (UniswapV3Pool.sol#734-742)
        State variables written after the call(s):
        - feeGrowthGlobal0X128 = state.feeGrowthGlobalX128 (UniswapV3Pool.sol#760)
        - feeGrowthGlobal1X128 = state.feeGrowthGlobalX128 (UniswapV3Pool.sol#763)
        - liquidity = state.liquidity (UniswapV3Pool.sol#755)
        - (slot0.sqrtPriceX96,slot0.tick,slot0.observationIndex,slot0.observationCardinality) = (state.sqrtPriceX96,state.tick,observationIndex,observationCardinality) (UniswapV3Pool.sol#743-748)
Reentrancy in UniswapV3Pool.swap(address,bool,int256,uint160,bytes) (UniswapV3Pool.sol#596-788):
        External calls:
        - (cache.tickCumulative,cache.secondsPerLiquidityCumulativeX128) = observations.observeSingle(cache.blockTimestamp,0,slot0Start.tick,slot0Start.observationIndex,cache.liquidityStart,slot0Start.observationCa
rdinality) (UniswapV3Pool.sol#699-706)
        State variables written after the call(s):
        - slot0.sqrtPriceX96 = state.sqrtPriceX96 (UniswapV3Pool.sol#751)
Reentrancy in UniswapV3Pool.swap(address,bool,int256,uint160,bytes) (UniswapV3Pool.sol#596-788):
        External calls:
        - (cache.tickCumulative,cache.secondsPerLiquidityCumulativeX128) = observations.observeSingle(cache.blockTimestamp,0,slot0Start.tick,slot0Start.observationIndex,cache.liquidityStart,slot0Start.observationCa
rdinality) (UniswapV3Pool.sol#699-706)
        - (observationIndex,observationCardinality) = observations.write(slot0Start.observationIndex,cache.blockTimestamp,slot0Start.tick,cache.liquidityStart,slot0Start.observationCardinality,slot0Start.observatio
nCardinalityNext) (UniswapV3Pool.sol#734-742)
        - TransferHelper.safeTransfer(token1,recipient,uint256(-amount1)) (UniswapV3Pool.sol#773)
        - balance0Before = balance0() (UniswapV3Pool.sol#775)
                - (success,data) = token0.staticcall(abi.encodeWithSelector(IERC20Minimal.balanceOf.selector,address(this))) (UniswapV3Pool.sol#141-142)
        - IUniswapV3SwapCallback(msg.sender).uniswapV3SwapCallback(amount0,amount1,data) (UniswapV3Pool.sol#776)
        - require(bool,string)(balance0Before.add(uint256(amount0)) <= balance0(),IIA) (UniswapV3Pool.sol#777)
                - (success,data) = token0.staticcall(abi.encodeWithSelector(IERC20Minimal.balanceOf.selector,address(this))) (UniswapV3Pool.sol#141-142)
        - TransferHelper.safeTransfer(token0,recipient,uint256(-amount0)) (UniswapV3Pool.sol#779)
        - balance1Before = balance1() (UniswapV3Pool.sol#781)
                - (success,data) = token1.staticcall(abi.encodeWithSelector(IERC20Minimal.balanceOf.selector,address(this))) (UniswapV3Pool.sol#151-152)
        - IUniswapV3SwapCallback(msg.sender).uniswapV3SwapCallback(amount0,amount1,data) (UniswapV3Pool.sol#782)
        - require(bool,string)(balance1Before.add(uint256(amount1)) <= balance1(),IIA) (UniswapV3Pool.sol#783)
                - (success,data) = token1.staticcall(abi.encodeWithSelector(IERC20Minimal.balanceOf.selector,address(this))) (UniswapV3Pool.sol#151-152)
        State variables written after the call(s):
        - slot0.unlocked = true (UniswapV3Pool.sol#787)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-1

SqrtPriceMath.getNextSqrtPriceFromAmount0RoundingUp(uint160,uint128,uint256,bool).product (libraries/SqrtPriceMath.sol#39) is a local variable never initialized
TickBitmap.nextInitializedTickWithinOneWord(mapping(int16 => uint256),int24,int24,bool).wordPos_scope_0 (libraries/TickBitmap.sol#65) is a local variable never initialized
SqrtPriceMath.getNextSqrtPriceFromAmount0RoundingUp(uint160,uint128,uint256,bool).product_scope_0 (libraries/SqrtPriceMath.sol#49) is a local variable never initialized
UniswapV3Pool._updatePosition(address,int24,int24,int128,int24).flippedUpper (UniswapV3Pool.sol#393) is a local variable never initialized
UniswapV3Pool.mint(address,int24,int24,uint128,bytes).balance0Before (UniswapV3Pool.sol#478) is a local variable never initialized
TickBitmap.nextInitializedTickWithinOneWord(mapping(int16 => uint256),int24,int24,bool).bitPos_scope_1 (libraries/TickBitmap.sol#65) is a local variable never initialized
UniswapV3Pool.mint(address,int24,int24,uint128,bytes).balance1Before (UniswapV3Pool.sol#479) is a local variable never initialized
```

```
NoDelegateCall.checkNotDelegateCall() (NoDelegateCall.sol#18-20) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

UniswapV3Pool.observations (UniswapV3Pool.sol#99) is never initialized. It is used in:
        - UniswapV3Pool.snapshotCumulativesInside(int24,int24) (UniswapV3Pool.sol#158-233)
        - UniswapV3Pool.observe(uint32[]) (UniswapV3Pool.sol#236-252)
        - UniswapV3Pool.increaseObservationCardinalityNext(uint16) (UniswapV3Pool.sol#255-267)
        - UniswapV3Pool.initialize(uint160) (UniswapV3Pool.sol#271-289)
        - UniswapV3Pool._modifyPosition(UniswapV3Pool.ModifyPositionParams) (UniswapV3Pool.sol#306-372)
        - UniswapV3Pool._updatePosition(address,int24,int24,int128,int24) (UniswapV3Pool.sol#379-453)
        - UniswapV3Pool.swap(address,bool,int256,uint160,bytes) (UniswapV3Pool.sol#596-788)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-state-variables

FullMath.mulDiv(uint256,uint256,uint256) (libraries/FullMath.sol#14-106) performs a multiplication on the result of a division:
        -denominator = denominator / twos (libraries/FullMath.sol#67)
        -inv = (3 * denominator) ^ 2 (libraries/FullMath.sol#87)
FullMath.mulDiv(uint256,uint256,uint256) (libraries/FullMath.sol#14-106) performs a multiplication on the result of a division:
        -denominator = denominator / twos (libraries/FullMath.sol#67)
        -inv *= 2 - denominator * inv (libraries/FullMath.sol#91)
FullMath.mulDiv(uint256,uint256,uint256) (libraries/FullMath.sol#14-106) performs a multiplication on the result of a division:
        -denominator = denominator / twos (libraries/FullMath.sol#67)
        -inv *= 2 - denominator * inv (libraries/FullMath.sol#92)
FullMath.mulDiv(uint256,uint256,uint256) (libraries/FullMath.sol#14-106) performs a multiplication on the result of a division:
        -denominator = denominator / twos (libraries/FullMath.sol#67)
        -inv *= 2 - denominator * inv (libraries/FullMath.sol#93)
FullMath.mulDiv(uint256,uint256,uint256) (libraries/FullMath.sol#14-106) performs a multiplication on the result of a division:
        -denominator = denominator / twos (libraries/FullMath.sol#67)
        -inv *= 2 - denominator * inv (libraries/FullMath.sol#94)
FullMath.mulDiv(uint256,uint256,uint256) (libraries/FullMath.sol#14-106) performs a multiplication on the result of a division:
        -denominator = denominator / twos (libraries/FullMath.sol#67)
        -inv *= 2 - denominator * inv (libraries/FullMath.sol#95)
FullMath.mulDiv(uint256,uint256,uint256) (libraries/FullMath.sol#14-106) performs a multiplication on the result of a division:
        -denominator = denominator / twos (libraries/FullMath.sol#67)
        -inv *= 2 - denominator * inv (libraries/FullMath.sol#96)
FullMath.mulDiv(uint256,uint256,uint256) (libraries/FullMath.sol#14-106) performs a multiplication on the result of a division:
        -prod0 = prod0 / twos (libraries/FullMath.sol#72)
        -result = prod0 * inv (libraries/FullMath.sol#104)
Oracle.observeSingle(Oracle.Observation[65535],uint32,uint32,int24,uint16,uint128,uint16) (libraries/Oracle.sol#245-287) performs a multiplication on the result of a division:
        -(beforeOrAt.tickCumulative + ((atOrAfter.tickCumulative - beforeOrAt.tickCumulative) / observationTimeDelta) * targetDelta,beforeOrAt.secondsPerLiquidityCumulativeX128 + uint160((uint256(atOrAfter.secondsP
erLiquidityCumulativeX128 - beforeOrAt.secondsPerLiquidityCumulativeX128) * targetDelta) / observationTimeDelta)) (libraries/Oracle.sol#275-285)
Tick.tickSpacingToMaxLiquidityPerTick(int24) (libraries/Tick.sol#44-49) performs a multiplication on the result of a division:
        -minTick = (TickMath.MIN_TICK / tickSpacing) * tickSpacing (libraries/Tick.sol#45)
Tick.tickSpacingToMaxLiquidityPerTick(int24) (libraries/Tick.sol#44-49) performs a multiplication on the result of a division:
        -maxTick = (TickMath.MAX_TICK / tickSpacing) * tickSpacing (libraries/Tick.sol#46)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#divide-before-multiply

UniswapV3Pool.initialize(uint160) (UniswapV3Pool.sol#271-289) uses a dangerous strict equality:
        - require(bool,string)(slot0.sqrtPriceX96 == 0,AI) (UniswapV3Pool.sol#272)
UniswapV3Pool.swap(address,bool,int256,uint160,bytes) (UniswapV3Pool.sol#596-788) uses a dangerous strict equality:
        - state.sqrtPriceX96 == step.sqrtPriceNextX96 (UniswapV3Pool.sol#693)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dangerous-strict-equalities

Reentrancy in UniswapV3Pool._modifyPosition(UniswapV3Pool.ModifyPositionParams) (UniswapV3Pool.sol#306-372):
        External calls:
        - position = _updatePosition(params.owner,params.tickLower,params.tickUpper,params.liquidityDelta,_slot0.tick) (UniswapV3Pool.sol#319-325)
                - (tickCumulative,secondsPerLiquidityCumulativeX128) = observations.observeSingle(time,0,slot0.tick,slot0.observationIndex,liquidity,slot0.observationCardinality) (UniswapV3Pool.sol#396-404)
        - (slot0.observationIndex,slot0.observationCardinality) = observations.write(_slot0.observationIndex,_blockTimestamp(),_slot0.tick,liquidityBefore,_slot0.observationCardinality,_slot0.observationCardinalityN
ext) (UniswapV3Pool.sol#341-348)
        State variables written after the call(s):
        - liquidity = LiquidityMath.addDelta(liquidityBefore,params.liquidityDelta) (UniswapV3Pool.sol#361)
UniswapV3Pool._updatePosition(address,int24,int24,int128,int24).flippedLower (UniswapV3Pool.sol#392) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

UniswapV3Factory.setOwner(address)._owner (UniswapV3Factory.sol#54) lacks a zero-check on :
                - owner = _owner (UniswapV3Factory.sol#57)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation

UniswapV3Pool.swap(address,bool,int256,uint160,bytes) (UniswapV3Pool.sol#596-788) has external calls inside a loop: (cache.tickCumulative,cache.secondsPerLiquidityCumulativeX128) = observations.observeSingle(cache.
blockTimestamp,0,slot0Start.tick,slot0Start.observationIndex,cache.liquidityStart,slot0Start.observationCardinality) (UniswapV3Pool.sol#699-706)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation/#calls-inside-a-loop

Variable 'UniswapV3Pool.swap(address,bool,int256,uint160,bytes).step (UniswapV3Pool.sol#642)' in UniswapV3Pool.swap(address,bool,int256,uint160,bytes) (UniswapV3Pool.sol#596-788) potentially used before declaratio
n: state.amountSpecifiedRemaining != 0 && state.sqrtPriceX96 != sqrtPriceLimitX96 (UniswapV3Pool.sol#641)
Variable 'TickBitmap.nextInitializedTickWithinOneWord(mapping(int16 => uint256),int24,int24,bool).bitPos (libraries/TickBitmap.sol#52)' in TickBitmap.nextInitializedTickWithinOneWord(mapping(int16 => uint256),int24
,int24,bool) (libraries/TickBitmap.sol#42-77) potentially used before declaration: (wordPos,bitPos) = position(compressed + 1) (libraries/TickBitmap.sol#65)
Variable 'TickBitmap.nextInitializedTickWithinOneWord(mapping(int16 => uint256),int24,int24,bool).wordPos (libraries/TickBitmap.sol#52)' in TickBitmap.nextInitializedTickWithinOneWord(mapping(int16 => uint256),int2
4,int24,bool) (libraries/TickBitmap.sol#42-77) potentially used before declaration: (wordPos,bitPos) = position(compressed + 1) (libraries/TickBitmap.sol#65)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables

Reentrancy in UniswapV3Pool.flash(address,uint256,uint256,bytes) (UniswapV3Pool.sol#791-834):
        External calls:
        - balance0Before = balance0() (UniswapV3Pool.sol#802)
                - (success,data) = token0.staticcall(abi.encodeWithSelector(IERC20Minimal.balanceOf.selector,address(this))) (UniswapV3Pool.sol#141-142)
        - balance1Before = balance1() (UniswapV3Pool.sol#803)
                - (success,data) = token1.staticcall(abi.encodeWithSelector(IERC20Minimal.balanceOf.selector,address(this))) (UniswapV3Pool.sol#151-152)
        - TransferHelper.safeTransfer(token0,recipient,amount0) (UniswapV3Pool.sol#805)
        - TransferHelper.safeTransfer(token1,recipient,amount1) (UniswapV3Pool.sol#806)
        - IUniswapV3FlashCallback(msg.sender).uniswapV3FlashCallback(fee0,fee1,data) (UniswapV3Pool.sol#808)
        - balance0After = balance0() (UniswapV3Pool.sol#810)
                - (success,data) = token0.staticcall(abi.encodeWithSelector(IERC20Minimal.balanceOf.selector,address(this))) (UniswapV3Pool.sol#141-142)
        - balance1After = balance1() (UniswapV3Pool.sol#811)
                - (success,data) = token1.staticcall(abi.encodeWithSelector(IERC20Minimal.balanceOf.selector,address(this))) (UniswapV3Pool.sol#151-152)
        State variables written after the call(s):
        - feeGrowthGlobal0X128 += FullMath.mulDiv(paid0 - fees0,FixedPoint128.Q128,_liquidity) (UniswapV3Pool.sol#824)
        - feeGrowthGlobal1X128 += FullMath.mulDiv(paid1 - fees1,FixedPoint128.Q128,_liquidity) (UniswapV3Pool.sol#830)
        - protocolFees.token0 += uint128(fees0) (UniswapV3Pool.sol#823)
        - protocolFees.token1 += uint128(fees1) (UniswapV3Pool.sol#829)
Reentrancy in UniswapV3Pool.swap(address,bool,int256,uint160,bytes) (UniswapV3Pool.sol#596-788):
        External calls:
        - (cache.tickCumulative,cache.secondsPerLiquidityCumulativeX128) = observations.observeSingle(cache.blockTimestamp,0,slot0Start.tick,slot0Start.observationIndex,cache.liquidityStart,slot0Start.observationCa
rdinality) (UniswapV3Pool.sol#699-706)
        - (observationIndex,observationCardinality) = observations.write(slot0Start.observationIndex,cache.blockTimestamp,slot0Start.tick,cache.liquidityStart,slot0Start.observationCardinality,slot0Start.observatio
nCardinalityNext) (UniswapV3Pool.sol#734-742)
        State variables written after the call(s):
        - protocolFees.token0 += state.protocolFee (UniswapV3Pool.sol#761)
        - protocolFees.token1 += state.protocolFee (UniswapV3Pool.sol#764)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2

Reentrancy in UniswapV3Pool.burn(int24,int24,uint128) (UniswapV3Pool.sol#517-543):
        External calls:
        - (position,amount0Int,amount1Int) = _modifyPosition(ModifyPositionParams(msg.sender,tickLower,tickUpper,- int256(amount).toInt128())) (UniswapV3Pool.sol#522-530)
                - (slot0.observationIndex,slot0.observationCardinality) = observations.write(_slot0.observationIndex,_blockTimestamp(),_slot0.tick,liquidityBefore,_slot0.observationCardinality,_slot0.observationCa
rdinalityNext) (UniswapV3Pool.sol#341-348)
                - (tickCumulative,secondsPerLiquidityCumulativeX128) = observations.observeSingle(time,0,slot0.tick,slot0.observationIndex,liquidity,slot0.observationCardinality) (UniswapV3Pool.sol#396-404)
        Event emitted after the call(s):
        - Burn(msg.sender,tickLower,tickUpper,amount,amount0,amount1) (UniswapV3Pool.sol#542)
Reentrancy in UniswapV3Pool.collect(address,int24,int24,uint128,uint128) (UniswapV3Pool.sol#490-513):
        External calls:
        - TransferHelper.safeTransfer(token0,recipient,amount0) (UniswapV3Pool.sol#505)
        - TransferHelper.safeTransfer(token1,recipient,amount1) (UniswapV3Pool.sol#509)
```

```
        - TransferHelper.safeTransfer(token0,recipient,uint256(- amount0)) (UniswapV3Pool.sol#779)
        - balance1Before = balance1() (UniswapV3Pool.sol#781)
                - (success,data) = token1.staticcall(abi.encodeWithSelector(IERC20Minimal.balanceOf.selector,address(this))) (UniswapV3Pool.sol#151-152)
        - IUniswapV3SwapCallback(msg.sender).uniswapV3SwapCallback(amount0,amount1,data) (UniswapV3Pool.sol#782)
        - require(bool,string)(balance1Before.add(uint256(amount1)) <= balance1(),IIA) (UniswapV3Pool.sol#783)
                - (success,data) = token1.staticcall(abi.encodeWithSelector(IERC20Minimal.balanceOf.selector,address(this))) (UniswapV3Pool.sol#151-152)
        Event emitted after the call(s):
        - Swap(msg.sender,recipient,amount0,amount1,state.sqrtPriceX96,state.liquidity,state.tick) (UniswapV3Pool.sol#786)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

UniswapV3Pool.initialize(uint160) (UniswapV3Pool.sol#271-289) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(slot0.sqrtPriceX96 == 0,AI) (UniswapV3Pool.sol#272)
UniswapV3Pool.swap(address,bool,int256,uint160,bytes) (UniswapV3Pool.sol#596-788) uses timestamp for comparisons
        Dangerous comparisons:
        - state.amountSpecifiedRemaining != 0 && state.sqrtPriceX96 != sqrtPriceLimitX96 (UniswapV3Pool.sol#641)
        - step.tickNext < TickMath.MIN_TICK (UniswapV3Pool.sol#653)
        - step.tickNext > TickMath.MAX_TICK (UniswapV3Pool.sol#655)
        - cache.feeProtocol > 0 (UniswapV3Pool.sol#682)
        - state.liquidity > 0 (UniswapV3Pool.sol#689)
        - state.sqrtPriceX96 == step.sqrtPriceNextX96 (UniswapV3Pool.sol#693)
        - state.sqrtPriceX96 != step.sqrtPriceStartX96 (UniswapV3Pool.sol#726)
        - state.tick != slot0Start.tick (UniswapV3Pool.sol#733)
        - cache.liquidityStart != state.liquidity (UniswapV3Pool.sol#755)
        - state.protocolFee > 0 (UniswapV3Pool.sol#761)
        - state.protocolFee > 0 (UniswapV3Pool.sol#764)
        - amount1 < 0 (UniswapV3Pool.sol#773)
        - require(bool,string)(balance0Before.add(uint256(amount0)) <= balance0(),IIA) (UniswapV3Pool.sol#777)
        - amount0 < 0 (UniswapV3Pool.sol#779)
        - require(bool,string)(balance1Before.add(uint256(amount1)) <= balance1(),IIA) (UniswapV3Pool.sol#783)
        - (step.sqrtPriceNextX96 < sqrtPriceLimitX96) (UniswapV3Pool.sol#663-671)
        - (step.sqrtPriceNextX96 > sqrtPriceLimitX96) (UniswapV3Pool.sol#663-671)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

FullMath.mulDiv(uint256,uint256,uint256) (libraries/FullMath.sol#14-106) uses assembly
        - INLINE ASM (libraries/FullMath.sol#26-30)
        - INLINE ASM (libraries/FullMath.sol#35-37)
        - INLINE ASM (libraries/FullMath.sol#52-54)
        - INLINE ASM (libraries/FullMath.sol#56-59)
        - INLINE ASM (libraries/FullMath.sol#66-68)
        - INLINE ASM (libraries/FullMath.sol#71-73)
        - INLINE ASM (libraries/FullMath.sol#77-79)
TickMath.getTickAtSqrtRatio(uint160) (libraries/TickMath.sol#61-204) uses assembly
        - INLINE ASM (libraries/TickMath.sol#69-73)
        - INLINE ASM (libraries/TickMath.sol#74-78)
        - INLINE ASM (libraries/TickMath.sol#79-83)
        - INLINE ASM (libraries/TickMath.sol#84-88)
        - INLINE ASM (libraries/TickMath.sol#89-93)
        - INLINE ASM (libraries/TickMath.sol#94-98)
        - INLINE ASM (libraries/TickMath.sol#99-103)
        - INLINE ASM (libraries/TickMath.sol#104-107)
        - INLINE ASM (libraries/TickMath.sol#114-119)
        - INLINE ASM (libraries/TickMath.sol#120-125)
        - INLINE ASM (libraries/TickMath.sol#126-131)
        - INLINE ASM (libraries/TickMath.sol#132-137)
        - INLINE ASM (libraries/TickMath.sol#138-143)
        - INLINE ASM (libraries/TickMath.sol#144-149)
        - INLINE ASM (libraries/TickMath.sol#150-155)
        - INLINE ASM (libraries/TickMath.sol#156-161)
        - INLINE ASM (libraries/TickMath.sol#162-167)
        - INLINE ASM (libraries/TickMath.sol#168-173)
        - INLINE ASM (libraries/TickMath.sol#174-179)
        - INLINE ASM (libraries/TickMath.sol#180-185)
        - INLINE ASM (libraries/TickMath.sol#192-196)
UnsafeMath.divRoundingUp(uint256,uint256) (libraries/UnsafeMath.sol#12-16) uses assembly
        - INLINE ASM (libraries/UnsafeMath.sol#13-15)
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', '>=0.4.0<0.8.0', '>=0.5.0', '>=0.5.0<0.8.0', '>=0.6.0', '>=0.7.0']
        - =0.7.6 (NoDelegateCall.sol#2)
        - =0.7.6 (UniswapV3Factory.sol#2)
        - =0.7.6 (UniswapV3Pool.sol#2)
        - =0.7.6 (UniswapV3PoolDeployer.sol#2)
        - >=0.5.0 (interfaces/IERC20Minimal.sol#2)
        - >=0.5.0 (interfaces/IUniswapV3Factory.sol#2)
        - >=0.5.0 (interfaces/IUniswapV3Pool.sol#2)
        - >=0.5.0 (interfaces/IUniswapV3PoolDeployer.sol#2)
        - >=0.5.0 (interfaces/callback/IUniswapV3FlashCallback.sol#2)
        - >=0.5.0 (interfaces/callback/IUniswapV3MintCallback.sol#2)
        - >=0.5.0 (interfaces/callback/IUniswapV3SwapCallback.sol#2)
        - >=0.5.0 (interfaces/pool/IUniswapV3PoolActions.sol#2)
        - >=0.5.0 (interfaces/pool/IUniswapV3PoolDerivedState.sol#2)
        - >=0.5.0 (interfaces/pool/IUniswapV3PoolEvents.sol#2)
        - >=0.5.0 (interfaces/pool/IUniswapV3PoolImmutables.sol#2)
        - >=0.5.0 (interfaces/pool/IUniswapV3PoolOwnerActions.sol#2)
        - >=0.5.0 (interfaces/pool/IUniswapV3PoolState.sol#2)
        - >=0.5.0 (libraries/BitMath.sol#2)
        - >=0.4.0 (libraries/FixedPoint128.sol#2)
        - >=0.4.0 (libraries/FixedPoint96.sol#2)
        - >=0.4.0<0.8.0 (libraries/FullMath.sol#2)
        - >=0.5.0 (libraries/LiquidityMath.sol#2)
        - >=0.7.0 (libraries/LowGasSafeMath.sol#2)
        - >=0.5.0<0.8.0 (libraries/Oracle.sol#2)
        - >=0.5.0<0.8.0 (libraries/Position.sol#2)
        - >=0.5.0 (libraries/SafeCast.sol#2)
        - >=0.5.0 (libraries/SqrtPriceMath.sol#2)
        - >=0.5.0 (libraries/SwapMath.sol#2)
        - >=0.5.0<0.8.0 (libraries/Tick.sol#2)
        - >=0.5.0 (libraries/TickBitmap.sol#2)
        - >=0.5.0<0.8.0 (libraries/TickMath.sol#2)
        - >=0.6.0 (libraries/TransferHelper.sol#2)
        - >=0.5.0 (libraries/UnsafeMath.sol#2)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

LowGasSafeMath.mul(uint256,uint256) (libraries/LowGasSafeMath.sol#27-29) is never used and should be removed
LowGasSafeMath.sub(uint256,uint256) (libraries/LowGasSafeMath.sol#19-21) is never used and should be removed
Oracle.binarySearch(Oracle.Observation[65535],uint32,uint32,uint16,uint16) (libraries/Oracle.sol#153-184) is never used and should be removed
Oracle.getSurroundingObservations(Oracle.Observation[65535],uint32,uint32,int24,uint16,uint128,uint16) (libraries/Oracle.sol#198-230) is never used and should be removed
Oracle.grow(Oracle.Observation[65535],uint16,uint16) (libraries/Oracle.sol#108-120) is never used and should be removed
Oracle.initialize(Oracle.Observation[65535],uint32) (libraries/Oracle.sol#52-63) is never used and should be removed
Oracle.lte(uint32,uint32,uint32) (libraries/Oracle.sol#128-140) is never used and should be removed
Oracle.observe(Oracle.Observation[65535],uint32,uint32[],int24,uint16,uint128,uint16) (libraries/Oracle.sol#300-324) is never used and should be removed
Oracle.observeSingle(Oracle.Observation[65535],uint32,uint32,int24,uint16,uint128,uint16) (libraries/Oracle.sol#245-287) is never used and should be removed
Oracle.transform(Oracle.Observation,uint32,int24,uint128) (libraries/Oracle.sol#30-45) is never used and should be removed
Oracle.write(Oracle.Observation[65535],uint16,uint32,int24,uint128,uint16,uint16) (libraries/Oracle.sol#78-101) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version>=0.5.0 (interfaces/IERC20Minimal.sol#2) allows old versions
Pragma version>=0.5.0 (interfaces/IUniswapV3Factory.sol#2) allows old versions
Pragma version>=0.5.0 (interfaces/IUniswapV3Pool.sol#2) allows old versions
Pragma version>=0.5.0 (interfaces/IUniswapV3PoolDeployer.sol#2) allows old versions
Pragma version>=0.5.0 (interfaces/callback/IUniswapV3FlashCallback.sol#2) allows old versions
Pragma version>=0.5.0 (interfaces/callback/IUniswapV3MintCallback.sol#2) allows old versions
Pragma version>=0.5.0 (interfaces/callback/IUniswapV3SwapCallback.sol#2) allows old versions
Pragma version>=0.5.0 (interfaces/pool/IUniswapV3PoolActions.sol#2) allows old versions
```

```
Pragma version>=0.5.0 (interfaces/pool/IUniswapV3PoolState.sol#2) allows old versions
Pragma version>=0.5.0 (libraries/BitMath.sol#2) allows old versions
Pragma version>=0.4.0 (libraries/FixedPoint128.sol#2) allows old versions
Pragma version>=0.4.0 (libraries/FixedPoint96.sol#2) allows old versions
Pragma version>=0.4.0<0.8.0 (libraries/FullMath.sol#2) is too complex
Pragma version>=0.5.0 (libraries/LiquidityMath.sol#2) allows old versions
Pragma version>=0.7.0 (libraries/LowGasSafeMath.sol#2) allows old versions
Pragma version>=0.5.0<0.8.0 (libraries/Oracle.sol#2) is too complex
Pragma version>=0.5.0<0.8.0 (libraries/Position.sol#2) is too complex
Pragma version>=0.5.0 (libraries/SafeCast.sol#2) allows old versions
Pragma version>=0.5.0 (libraries/SqrtPriceMath.sol#2) allows old versions
Pragma version>=0.5.0 (libraries/SwapMath.sol#2) allows old versions
Pragma version>=0.5.0<0.8.0 (libraries/Tick.sol#2) is too complex
Pragma version>=0.5.0 (libraries/TickBitmap.sol#2) allows old versions
Pragma version>=0.5.0<0.8.0 (libraries/TickMath.sol#2) is too complex
Pragma version>=0.6.0 (libraries/TransferHelper.sol#2) allows old versions
Pragma version>=0.5.0 (libraries/UnsafeMath.sol#2) allows old versions
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in UniswapV3Pool.balance0() (UniswapV3Pool.sol#140-145):
        - (success,data) = token0.staticcall(abi.encodeWithSelector(IERC20Minimal.balanceOf.selector,address(this))) (UniswapV3Pool.sol#141-142)
Low level call in UniswapV3Pool.balance1() (UniswapV3Pool.sol#150-155):
        - (success,data) = token1.staticcall(abi.encodeWithSelector(IERC20Minimal.balanceOf.selector,address(this))) (UniswapV3Pool.sol#151-152)
Low level call in TransferHelper.safeTransfer(address,address,uint256) (libraries/TransferHelper.sol#14-22):
        - (success,data) = token.call(abi.encodeWithSelector(IERC20Minimal.transfer.selector,to,value)) (libraries/TransferHelper.sol#19-20)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Parameter UniswapV3Factory.setOwner(address)._owner (UniswapV3Factory.sol#54) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Variable UniswapV3Pool._updatePosition(address,int24,int24,int128,int24)._feeGrowthGlobal0X128 (UniswapV3Pool.sol#388) is too similar to UniswapV3Pool._updatePosition(address,int24,int24,int128,int24)._feeGrowthGlo
bal1X128 (UniswapV3Pool.sol#389)
Variable UniswapV3Pool.collect(address,int24,int24,uint128,uint128).amount0Requested (UniswapV3Pool.sol#494) is too similar to IUniswapV3PoolOwnerActions.collectProtocol(address,uint128,uint128).amount1Requested (i
nterfaces/pool/IUniswapV3PoolOwnerActions.sol#21)
Variable UniswapV3Pool.collectProtocol(address,uint128,uint128).amount0Requested (UniswapV3Pool.sol#850) is too similar to IUniswapV3PoolOwnerActions.collectProtocol(address,uint128,uint128).amount1Requested (inter
faces/pool/IUniswapV3PoolOwnerActions.sol#21)
Variable UniswapV3Pool.collectProtocol(address,uint128,uint128).amount0Requested (UniswapV3Pool.sol#850) is too similar to UniswapV3Pool.collect(address,int24,int24,uint128,uint128).amount1Requested (UniswapV3Pool.
sol#495)
Variable IUniswapV3PoolActions.collect(address,int24,int24,uint128,uint128).amount0Requested (interfaces/pool/IUniswapV3PoolActions.sol#47) is too similar to IUniswapV3PoolActions.collect(address,int24,int24,uint12
8,uint128).amount1Requested (interfaces/pool/IUniswapV3PoolActions.sol#48)
Variable IUniswapV3PoolActions.collect(address,int24,int24,uint128,uint128).amount0Requested (interfaces/pool/IUniswapV3PoolActions.sol#47) is too similar to IUniswapV3PoolOwnerActions.collectProtocol(address,uint1
28,uint128).amount1Requested (interfaces/pool/IUniswapV3PoolOwnerActions.sol#21)
Variable IUniswapV3PoolActions.collect(address,int24,int24,uint128,uint128).amount0Requested (interfaces/pool/IUniswapV3PoolActions.sol#47) is too similar to UniswapV3Pool.collect(address,int24,int24,uint128,uint12
8).amount1Requested (UniswapV3Pool.sol#495)
Variable UniswapV3Pool.collectProtocol(address,uint128,uint128).amount0Requested (UniswapV3Pool.sol#850) is too similar to IUniswapV3PoolActions.collect(address,int24,int24,uint128,uint128).amount1Requested (inter
aces/pool/IUniswapV3PoolActions.sol#48)
Variable UniswapV3Pool.collect(address,int24,int24,uint128,uint128).amount0Requested (UniswapV3Pool.sol#494) is too similar to UniswapV3Pool.collect(address,int24,int24,uint128,uint128).amount1Requested (UniswapV3P
ool.sol#495)
Variable UniswapV3Pool.collectProtocol(address,uint128,uint128).amount0Requested (UniswapV3Pool.sol#850) is too similar to UniswapV3Pool.collectProtocol(address,uint128,uint128).amount1Requested (UniswapV3Pool.sol#
851)
Variable IUniswapV3PoolOwnerActions.collectProtocol(address,uint128,uint128).amount0Requested (interfaces/pool/IUniswapV3PoolOwnerActions.sol#20) is too similar to IUniswapV3PoolOwnerActions.collectProtocol(address
,uint128,uint128).amount1Requested (interfaces/pool/IUniswapV3PoolOwnerActions.sol#21)
Variable IUniswapV3PoolOwnerActions.collectProtocol(address,uint128,uint128).amount0Requested (interfaces/pool/IUniswapV3PoolOwnerActions.sol#20) is too similar to UniswapV3Pool.collectProtocol(address,uint128,uint
128).amount1Requested (UniswapV3Pool.sol#851)
Variable UniswapV3Pool.collect(address,int24,int24,uint128,uint128).amount0Requested (UniswapV3Pool.sol#494) is too similar to IUniswapV3PoolActions.collect(address,int24,int24,uint128,uint128).amount1Requested (int
erfaces/pool/IUniswapV3PoolActions.sol#48)
Variable IUniswapV3PoolOwnerActions.collectProtocol(address,uint128,uint128).amount0Requested (interfaces/pool/IUniswapV3PoolOwnerActions.sol#20) is too similar to UniswapV3Pool.collect(address,int24,int24,uint128,
uint128).amount1Requested (UniswapV3Pool.sol#495)
Variable IUniswapV3PoolActions.collect(address,int24,int24,uint128,uint128).amount0Requested (interfaces/pool/IUniswapV3PoolActions.sol#47) is too similar to UniswapV3Pool.collectProtocol(address,uint128,uint128).a
mount1Requested (UniswapV3Pool.sol#851)
Variable IUniswapV3PoolActions.collect(address,int24,int24,uint128,uint128).amount0Requested (interfaces/pool/IUniswapV3PoolActions.sol#47) is too similar to IUniswapV3PoolOwnerActions.collectProtocol(address,uint128,
uint128).amount1Requested (interfaces/pool/IUniswapV3PoolOwnerActions.sol#21)
Variable UniswapV3Pool.collect(address,int24,int24,uint128,uint128).amount0Requested (UniswapV3Pool.sol#494) is too similar to UniswapV3Pool.collectProtocol(address,uint128,uint128).amount1Requested (UniswapV3Pool.
```

```
Variable UniswapV3Pool.flash(address,uint256,uint256,bytes).balance0After (UniswapV3Pool.sol#810) is too similar to UniswapV3Pool.flash(address,uint256,uint256,bytes).balance1After (UniswapV3Pool.sol#811)
Variable UniswapV3Pool.swap(address,bool,int256,uint160,bytes).balance0Before (UniswapV3Pool.sol#775) is too similar to UniswapV3Pool.flash(address,uint256,uint256,bytes).balance1Before (UniswapV3Pool.sol#803)
Variable UniswapV3Pool.swap(address,bool,int256,uint160,bytes).balance0Before (UniswapV3Pool.sol#775) is too similar to UniswapV3Pool.mint(address,int24,int24,uint128,bytes).balance1Before (UniswapV3Pool.sol#479)
Variable UniswapV3Pool.flash(address,uint256,uint256,bytes).balance0Before (UniswapV3Pool.sol#802) is too similar to UniswapV3Pool.mint(address,int24,int24,uint128,bytes).balance1Before (UniswapV3Pool.sol#479)
Variable UniswapV3Pool.flash(address,uint256,uint256,bytes).balance0Before (UniswapV3Pool.sol#802) is too similar to UniswapV3Pool.flash(address,uint256,uint256,bytes).balance1Before (UniswapV3Pool.sol#803)
Variable UniswapV3Pool.swap(address,bool,int256,uint160,bytes).balance0Before (UniswapV3Pool.sol#775) is too similar to UniswapV3Pool.swap(address,bool,int256,uint160,bytes).balance1Before (UniswapV3Pool.sol#781)
Variable UniswapV3Pool.mint(address,int24,int24,uint128,bytes).balance0Before (UniswapV3Pool.sol#478) is too similar to UniswapV3Pool.mint(address,int24,int24,uint128,bytes).balance1Before (UniswapV3Pool.sol#479)
Variable UniswapV3Pool.mint(address,int24,int24,uint128,bytes).balance0Before (UniswapV3Pool.sol#478) is too similar to UniswapV3Pool.swap(address,bool,int256,uint160,bytes).balance1Before (UniswapV3Pool.sol#781)
Variable UniswapV3Pool.flash(address,uint256,uint256,bytes).balance0Before (UniswapV3Pool.sol#802) is too similar to UniswapV3Pool.swap(address,bool,int256,uint160,bytes).balance1Before (UniswapV3Pool.sol#781)
Variable UniswapV3Pool.mint(address,int24,int24,uint128,bytes).balance0Before (UniswapV3Pool.sol#478) is too similar to UniswapV3Pool.flash(address,uint256,uint256,bytes).balance1Before (UniswapV3Pool.sol#803)
Variable UniswapV3Pool.feeGrowthGlobal0X128 (UniswapV3Pool.sol#77) is too similar to UniswapV3Pool.feeGrowthGlobal1X128 (UniswapV3Pool.sol#79)
Variable IUniswapV3PoolState.positions(bytes32).feeGrowthInside0LastX128 (interfaces/pool/IUniswapV3PoolState.sol#93) is too similar to IUniswapV3PoolState.positions(bytes32).feeGrowthInside1LastX128 (interfaces/po
ol/IUniswapV3PoolState.sol#94)
Variable UniswapV3Pool._updatePosition(address,int24,int24,int128,int24).feeGrowthInside0X128 (UniswapV3Pool.sol#439) is too similar to UniswapV3Pool._updatePosition(address,int24,int24,int128,int24).feeGrowthInsid
e1X128 (UniswapV3Pool.sol#439)
Variable IUniswapV3PoolState.ticks(int24).feeGrowthOutside0X128 (interfaces/pool/IUniswapV3PoolState.sol#70) is too similar to IUniswapV3PoolState.ticks(int24).feeGrowthOutside1X128 (interfaces/pool/IUniswapV3Pools
tate.sol#71)
Variable IUniswapV3PoolOwnerActions.setFeeProtocol(uint8,uint8).feeProtocol0 (interfaces/pool/IUniswapV3PoolOwnerActions.sol#10) is too similar to UniswapV3Pool.setFeeProtocol(uint8,uint8).feeProtocol1 (UniswapV3Po
ol.sol#837)
Variable UniswapV3Pool.flash(address,uint256,uint256,bytes).feeProtocol0 (UniswapV3Pool.sol#821) is too similar to UniswapV3Pool.flash(address,uint256,uint256,bytes).feeProtocol1 (UniswapV3Pool.sol#827)
Variable UniswapV3Pool.setFeeProtocol(uint8,uint8).feeProtocol0 (UniswapV3Pool.sol#837) is too similar to UniswapV3Pool.setFeeProtocol(uint8,uint8).feeProtocol1 (UniswapV3Pool.sol#837)
Variable UniswapV3Pool.flash(address,uint256,uint256,bytes).feeProtocol0 (UniswapV3Pool.sol#821) is too similar to UniswapV3Pool.setFeeProtocol(uint8,uint8).feeProtocol1 (UniswapV3Pool.sol#837)
Variable IUniswapV3PoolOwnerActions.setFeeProtocol(uint8,uint8).feeProtocol0 (interfaces/pool/IUniswapV3PoolOwnerActions.sol#10) is too similar to IUniswapV3PoolOwnerActions.setFeeProtocol(uint8,uint8).feeProtocol1
 (interfaces/pool/IUniswapV3PoolOwnerActions.sol#10)
Variable UniswapV3Pool.flash(address,uint256,uint256,bytes).feeProtocol0 (UniswapV3Pool.sol#821) is too similar to IUniswapV3PoolOwnerActions.setFeeProtocol(uint8,uint8).feeProtocol1 (interfaces/pool/IUniswapV3Pool
OwnerActions.sol#10)
Variable UniswapV3Pool.setFeeProtocol(uint8,uint8).feeProtocol0 (UniswapV3Pool.sol#837) is too similar to UniswapV3Pool.flash(address,uint256,uint256,bytes).feeProtocol1 (UniswapV3Pool.sol#827)
Variable IUniswapV3PoolOwnerActions.setFeeProtocol(uint8,uint8).feeProtocol0 (interfaces/pool/IUniswapV3PoolOwnerActions.sol#10) is too similar to UniswapV3Pool.flash(address,uint256,uint256,bytes).feeProtocol1 (Un
iswapV3Pool.sol#827)
Variable UniswapV3Pool.setFeeProtocol(uint8,uint8).feeProtocol0 (UniswapV3Pool.sol#837) is too similar to IUniswapV3PoolOwnerActions.setFeeProtocol(uint8,uint8).feeProtocol1 (interfaces/pool/IUniswapV3PoolOwnerActi
ons.sol#10)
Variable UniswapV3Pool.snapshotCumulativesInside(int24,int24).secondsPerLiquidityOutsideLowerX128 (UniswapV3Pool.sol#173) is too similar to UniswapV3Pool.snapshotCumulativesInside(int24,int24).secondsPerLiquidityOu
tsideUpperX128 (UniswapV3Pool.sol#174)
Variable IUniswapV3PoolState.positions(bytes32).tokensOwed0 (interfaces/pool/IUniswapV3PoolState.sol#95) is too similar to IUniswapV3PoolState.positions(bytes32).tokensOwed1 (interfaces/pool/IUniswapV3PoolState.sol
#96)
Variable IUniswapV3MintCallback.uniswapV3MintCallback(uint256,uint256,bytes).amount0Owed (interfaces/callback/IUniswapV3MintCallback.sol#14) is too similar to IUniswapV3MintCallback.uniswapV3MintCallback(uint256,ui
nt256,bytes).amount1Owed (interfaces/callback/IUniswapV3MintCallback.sol#15)
Variable IUniswapV3SwapCallback.uniswapV3SwapCallback(int256,int256,bytes).amount0Delta (interfaces/callback/IUniswapV3SwapCallback.sol#17) is too similar to IUniswapV3SwapCallback.uniswapV3SwapCallback(int256,int2
56,bytes).amount1Delta (interfaces/callback/IUniswapV3SwapCallback.sol#18)
Variable Position.update(Position.Info,int128,uint256,uint256).feeGrowthInside0X128 (libraries/Position.sol#47) is too similar to Position.update(Position.Info,int128,uint256,uint256).feeGrowthInside1X128 (librarie
s/Position.sol#48)
Variable Position.update(Position.Info,int128,uint256,uint256).tokensOwed0 (libraries/Position.sol#61-68) is too similar to Position.update(Position.Info,int128,uint256,uint256).tokensOwed1 (libraries/Position.sol#
69-76)
Variable SqrtPriceMath.getAmount0Delta(uint160,uint160,int128).sqrtRatioAX96 (libraries/SqrtPriceMath.sol#202) is too similar to SqrtPriceMath.getAmount0Delta(uint160,uint160,int128).sqrtRatioBX96 (libraries/SqrtPr
iceMath.sol#203)
Variable SqrtPriceMath.getAmount1Delta(uint160,uint160,uint128,bool).sqrtRatioAX96 (libraries/SqrtPriceMath.sol#183) is too similar to SqrtPriceMath.getAmount1Delta(uint160,uint160,uint128,bool).sqrtRatioBX96 (libr
aries/SqrtPriceMath.sol#184)
Variable SqrtPriceMath.getAmount0Delta(uint160,uint160,int128).sqrtRatioAX96 (libraries/SqrtPriceMath.sol#202) is too similar to SqrtPriceMath.getAmount0Delta(uint160,uint160,uint128,bool).sqrtRatioBX96 (libraries/
SqrtPriceMath.sol#155)
Variable SqrtPriceMath.getAmount0Delta(uint160,uint160,uint128,bool).sqrtRatioAX96 (libraries/SqrtPriceMath.sol#154) is too similar to SqrtPriceMath.getAmount0Delta(uint160,uint160,uint128,bool).sqrtRatioBX96 (libr
aries/SqrtPriceMath.sol#155)
Variable SqrtPriceMath.getAmount1Delta(uint160,uint160,uint128,bool).sqrtRatioAX96 (libraries/SqrtPriceMath.sol#183) is too similar to SqrtPriceMath.getAmount0Delta(uint160,uint160,int128).sqrtRatioBX96 (libraries/
SqrtPriceMath.sol#203)
Variable SqrtPriceMath.getAmount1Delta(uint160,uint160,uint128,bool).sqrtRatioAX96 (libraries/SqrtPriceMath.sol#183) is too similar to SqrtPriceMath.getAmount0Delta(uint160,uint160,uint128,bool).sqrtRatioBX96 (libr
aries/SqrtPriceMath.sol#155)
Variable SqrtPriceMath.getAmount1Delta(uint160,uint160,uint128,bool).sqrtRatioAX96 (libraries/SqrtPriceMath.sol#183) is too similar to SqrtPriceMath.getAmount1Delta(uint160,uint160,int128).sqrtRatioBX96 (libraries/
SqrtPriceMath.sol#219)
Variable SqrtPriceMath.getAmount0Delta(uint160,uint160,int128).sqrtRatioAX96 (libraries/SqrtPriceMath.sol#202) is too similar to SqrtPriceMath.getAmount1Delta(uint160,uint160,int128).sqrtRatioBX96 (libraries/SqrtPr
iceMath.sol#219)
Variable SqrtPriceMath.getAmount1Delta(uint160,uint160,int128).sqrtRatioAX96 (libraries/SqrtPriceMath.sol#218) is too similar to SqrtPriceMath.getAmount0Delta(uint160,uint160,uint128,bool).sqrtRatioBX96 (libraries/
SqrtPriceMath.sol#155)
```

```
Variable SqrtPriceMath.getAmount1Delta(uint160,uint160,int128).sqrtRatioAX96 (libraries/SqrtPriceMath.sol#218) is too similar to SqrtPriceMath.getAmount0Delta(uint160,uint160,uint128,bool).sqrtRatioBX96 (libraries/
SqrtPriceMath.sol#155)
Variable SqrtPriceMath.getAmount0Delta(uint160,uint160,uint128,bool).sqrtRatioAX96 (libraries/SqrtPriceMath.sol#154) is too similar to SqrtPriceMath.getAmount1Delta(uint160,uint160,int128).sqrtRatioBX96 (libraries/
SqrtPriceMath.sol#219)
Variable SqrtPriceMath.getAmount1Delta(uint160,uint160,int128).sqrtRatioAX96 (libraries/SqrtPriceMath.sol#218) is too similar to SqrtPriceMath.getAmount0Delta(uint160,uint160,int128).sqrtRatioBX96 (libraries/SqrtPr
iceMath.sol#203)
Variable SqrtPriceMath.getAmount0Delta(uint160,uint160,int128).sqrtRatioAX96 (libraries/SqrtPriceMath.sol#202) is too similar to SqrtPriceMath.getAmount1Delta(uint160,uint160,uint128,bool).sqrtRatioBX96 (libraries/
SqrtPriceMath.sol#184)
Variable SqrtPriceMath.getAmount0Delta(uint160,uint160,uint128,bool).sqrtRatioAX96 (libraries/SqrtPriceMath.sol#154) is too similar to SqrtPriceMath.getAmount1Delta(uint160,uint160,uint128,bool).sqrtRatioBX96 (libr
aries/SqrtPriceMath.sol#184)
Variable SqrtPriceMath.getAmount1Delta(uint160,uint160,int128).sqrtRatioAX96 (libraries/SqrtPriceMath.sol#218) is too similar to SqrtPriceMath.getAmount1Delta(uint160,uint160,uint128,bool).sqrtRatioBX96 (libraries/
SqrtPriceMath.sol#184)
Variable SqrtPriceMath.getAmount0Delta(uint160,uint160,uint128,bool).sqrtRatioAX96 (libraries/SqrtPriceMath.sol#154) is too similar to SqrtPriceMath.getAmount0Delta(uint160,uint160,int128).sqrtRatioBX96 (libraries/
SqrtPriceMath.sol#203)
Variable SqrtPriceMath.getAmount1Delta(uint160,uint160,int128).sqrtRatioAX96 (libraries/SqrtPriceMath.sol#218) is too similar to SqrtPriceMath.getAmount1Delta(uint160,uint160,int128).sqrtRatioBX96 (libraries/SqrtPr
iceMath.sol#219)
Variable Tick.getFeeGrowthInside(mapping(int24 => Tick.Info),int24,int24,int24,uint256,uint256).feeGrowthAbove0X128 (libraries/Tick.sol#83) is too similar to Tick.getFeeGrowthInside(mapping(int24 => Tick.Info),int2
4,int24,int24,uint256,uint256).feeGrowthAbove1X128 (libraries/Tick.sol#84)
Variable Tick.getFeeGrowthInside(mapping(int24 => Tick.Info),int24,int24,int24,uint256,uint256).feeGrowthBelow0X128 (libraries/Tick.sol#72) is too similar to Tick.getFeeGrowthInside(mapping(int24 => Tick.Info),int2
4,int24,int24,uint256,uint256).feeGrowthBelow1X128 (libraries/Tick.sol#73)
Variable Tick.getFeeGrowthInside(mapping(int24 => Tick.Info),int24,int24,int24,uint256,uint256).feeGrowthGlobal0X128 (libraries/Tick.sol#65) is too similar to Tick.update(mapping(int24 => Tick.Info),int24,int24,int
128,uint256,uint256,uint160,int56,uint32,bool,uint128).feeGrowthGlobal1X128 (libraries/Tick.sol#116)
Variable Tick.update(mapping(int24 => Tick.Info),int24,int24,int128,uint256,uint256,uint160,int56,uint32,bool,uint128).feeGrowthGlobal0X128 (libraries/Tick.sol#115) is too similar to Tick.update(mapping(int24 => Ti
ck.Info),int24,int24,int128,uint256,uint256,uint160,int56,uint32,bool,uint128).feeGrowthGlobal1X128 (libraries/Tick.sol#116)
Variable Tick.getFeeGrowthInside(mapping(int24 => Tick.Info),int24,int24,int24,uint256,uint256).feeGrowthGlobal0X128 (libraries/Tick.sol#65) is too similar to Tick.getFeeGrowthInside(mapping(int24 => Tick.Info),int
24,int24,int24,uint256,uint256).feeGrowthGlobal1X128 (libraries/Tick.sol#66)
Variable Tick.cross(mapping(int24 => Tick.Info),int24,uint256,uint256,uint160,int56,uint32).feeGrowthGlobal0X128 (libraries/Tick.sol#171) is too similar to Tick.update(mapping(int24 => Tick.Info),int24,int24,int128
,uint256,uint256,uint160,int56,uint32,bool,uint128).feeGrowthGlobal1X128 (libraries/Tick.sol#116)
Variable Tick.cross(mapping(int24 => Tick.Info),int24,uint256,uint256,uint160,int56,uint32).feeGrowthGlobal0X128 (libraries/Tick.sol#171) is too similar to Tick.getFeeGrowthInside(mapping(int24 => Tick.Info),int24,
int24,int24,uint256,uint256).feeGrowthGlobal1X128 (libraries/Tick.sol#66)
Variable Tick.update(mapping(int24 => Tick.Info),int24,int24,int128,uint256,uint256,uint160,int56,uint32,bool,uint128).feeGrowthGlobal0X128 (libraries/Tick.sol#115) is too similar to Tick.cross(mapping(int24 => Tic
k.Info),int24,uint256,uint256,uint160,int56,uint32).feeGrowthGlobal1X128 (libraries/Tick.sol#172)
Variable Tick.getFeeGrowthInside(mapping(int24 => Tick.Info),int24,int24,int24,uint256,uint256).feeGrowthGlobal0X128 (libraries/Tick.sol#65) is too similar to Tick.cross(mapping(int24 => Tick.Info),int24,uint256,u
nt256,uint160,int56,uint32).feeGrowthGlobal1X128 (libraries/Tick.sol#172)
Variable Tick.update(mapping(int24 => Tick.Info),int24,int24,int128,uint256,uint256,uint160,int56,uint32,bool,uint128).feeGrowthGlobal0X128 (libraries/Tick.sol#115) is too similar to Tick.getFeeGrowthInside(mapping
(int24 => Tick.Info),int24,int24,int24,uint256,uint256).feeGrowthGlobal1X128 (libraries/Tick.sol#66)
Variable Tick.cross(mapping(int24 => Tick.Info),int24,uint256,uint256,uint160,int56,uint32).feeGrowthGlobal0X128 (libraries/Tick.sol#171) is too similar to Tick.cross(mapping(int24 => Tick.Info),int24,uint256,uint
56,uint160,int56,uint32).feeGrowthGlobal1X128 (libraries/Tick.sol#172)
Variable Tick.getFeeGrowthInside(mapping(int24 => Tick.Info),int24,int24,int24,uint256,uint256).feeGrowthInside0X128 (libraries/Tick.sol#67) is too similar to Tick.getFeeGrowthInside(mapping(int24 => Tick.Info),int
24,int24,int24,uint256,uint256).feeGrowthInside1X128 (libraries/Tick.sol#67)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-are-too-similar

UniswapV3Factory.enableFeeAmount(uint24,int24) (UniswapV3Factory.sol#61-72) uses literals with too many digits:
        - require(bool)(fee < 1000000) (UniswapV3Factory.sol#63)
BitMath.mostSignificantBit(uint256) (libraries/BitMath.sol#13-45) uses literals with too many digits:
        - x >= 0x100000000000000000000000000000000 (libraries/BitMath.sol#16)
BitMath.mostSignificantBit(uint256) (libraries/BitMath.sol#13-45) uses literals with too many digits:
        - x >= 0x10000000000000000 (libraries/BitMath.sol#20)
BitMath.mostSignificantBit(uint256) (libraries/BitMath.sol#13-45) uses literals with too many digits:
        - x >= 0x100000000 (libraries/BitMath.sol#24)
FixedPoint128.slitherConstructorConstantVariables() (libraries/FixedPoint128.sol#6-8) uses literals with too many digits:
        - Q128 = 0x100000000000000000000000000000000 (libraries/FixedPoint128.sol#7)
FixedPoint96.slitherConstructorConstantVariables() (libraries/FixedPoint96.sol#7-10) uses literals with too many digits:
        - Q96 = 0x1000000000000000000000000 (libraries/FixedPoint96.sol#9)
TickMath.getSqrtRatioAtTick(int24) (libraries/TickMath.sol#23-54) uses literals with too many digits:
        - ratio = 0x100000000000000000000000000000000 (libraries/TickMath.sol#27)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits

enableFeeAmount(uint24,int24) should be declared external:
        - UniswapV3Factory.enableFeeAmount(uint24,int24) (UniswapV3Factory.sol#61-72)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external

FullMath.mulDiv(uint256,uint256,uint256) (libraries/FullMath.sol#14-106) performs a multiplication on the result of a division:
        -denominator = denominator / twos (libraries/FullMath.sol#67)
        -inv = (3 * denominator) ^ 2 (libraries/FullMath.sol#87)
FullMath.mulDiv(uint256,uint256,uint256) (libraries/FullMath.sol#14-106) performs a multiplication on the result of a division:
        -denominator = denominator / twos (libraries/FullMath.sol#67)
        -inv *= 2 - denominator * inv (libraries/FullMath.sol#91)
FullMath.mulDiv(uint256,uint256,uint256) (libraries/FullMath.sol#14-106) performs a multiplication on the result of a division:
        -denominator = denominator / twos (libraries/FullMath.sol#67)
        -inv *= 2 - denominator * inv (libraries/FullMath.sol#92)
FullMath.mulDiv(uint256,uint256,uint256) (libraries/FullMath.sol#14-106) performs a multiplication on the result of a division:
        -denominator = denominator / twos (libraries/FullMath.sol#67)
        -inv *= 2 - denominator * inv (libraries/FullMath.sol#93)
FullMath.mulDiv(uint256,uint256,uint256) (libraries/FullMath.sol#14-106) performs a multiplication on the result of a division:
        -denominator = denominator / twos (libraries/FullMath.sol#67)
        -inv *= 2 - denominator * inv (libraries/FullMath.sol#94)
FullMath.mulDiv(uint256,uint256,uint256) (libraries/FullMath.sol#14-106) performs a multiplication on the result of a division:
        -denominator = denominator / twos (libraries/FullMath.sol#67)
        -inv *= 2 - denominator * inv (libraries/FullMath.sol#95)
FullMath.mulDiv(uint256,uint256,uint256) (libraries/FullMath.sol#14-106) performs a multiplication on the result of a division:
        -denominator = denominator / twos (libraries/FullMath.sol#67)
        -inv *= 2 - denominator * inv (libraries/FullMath.sol#96)
FullMath.mulDiv(uint256,uint256,uint256) (libraries/FullMath.sol#14-106) performs a multiplication on the result of a division:
        -prod0 = prod0 / twos (libraries/FullMath.sol#72)
        -result = prod0 * inv (libraries/FullMath.sol#104)
Oracle.observeSingle(Oracle.Observation[65535],uint32,uint32,int24,uint16,uint128,uint16) (libraries/Oracle.sol#245-287) performs a multiplication on the result of a division:
        -(beforeOrAt.tickCumulative + ((atOrAfter.tickCumulative - beforeOrAt.tickCumulative) / observationTimeDelta) * targetDelta,beforeOrAt.secondsPerLiquidityCumulativeX128 + uint160((uint256(atOrAfter.secondsP
erLiquidityCumulativeX128 - beforeOrAt.secondsPerLiquidityCumulativeX128) * targetDelta) / observationTimeDelta)) (libraries/Oracle.sol#275-285)
Tick.tickSpacingToMaxLiquidityPerTick(int24) (libraries/Tick.sol#44-49) performs a multiplication on the result of a division:
        -minTick = (TickMath.MIN_TICK / tickSpacing) * tickSpacing (libraries/Tick.sol#45)
Tick.tickSpacingToMaxLiquidityPerTick(int24) (libraries/Tick.sol#44-49) performs a multiplication on the result of a division:
        -maxTick = (TickMath.MAX_TICK / tickSpacing) * tickSpacing (libraries/Tick.sol#46)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#divide-before-multiply

UniswapV3Pool.initialize(uint160) (UniswapV3Pool.sol#271-289) uses a dangerous strict equality:
        - require(bool,string)(slot0.sqrtPriceX96 == 0,AI) (UniswapV3Pool.sol#272)
UniswapV3Pool.swap(address,bool,int256,uint160,bytes) (UniswapV3Pool.sol#596-788) uses a dangerous strict equality:
        - state.sqrtPriceX96 == step.sqrtPriceNextX96 (UniswapV3Pool.sol#693)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dangerous-strict-equalities

Reentrancy in UniswapV3Pool._modifyPosition(UniswapV3Pool.ModifyPositionParams) (UniswapV3Pool.sol#306-372):
        External calls:
        - position = _updatePosition(params.owner,params.tickLower,params.tickUpper,params.liquidityDelta,_slot0.tick) (UniswapV3Pool.sol#319-325)
                - (tickCumulative,secondsPerLiquidityCumulativeX128) = observations.observeSingle(time,0,slot0.tick,slot0.observationIndex,liquidity,slot0.observationCardinality) (UniswapV3Pool.sol#396-404)
        - (slot0.observationIndex,slot0.observationCardinality) = observations.write(_slot0.observationIndex,_blockTimestamp(),_slot0.tick,liquidityBefore,_slot0.observationCardinality,_slot0.observationCardinalityN
ext) (UniswapV3Pool.sol#341-348)
        State variables written after the call(s):
        - liquidity = LiquidityMath.addDelta(liquidityBefore,params.liquidityDelta) (UniswapV3Pool.sol#361)
        - (slot0.observationIndex,slot0.observationCardinality) = observations.write(_slot0.observationIndex,_blockTimestamp(),_slot0.tick,liquidityBefore,_slot0.observationCardinality,_slot0.observationCardinalityN
ext) (UniswapV3Pool.sol#341-348)
Reentrancy in UniswapV3Pool.collectProtocol(address,uint128,uint128) (UniswapV3Pool.sol#848-868):
        External calls:
        - TransferHelper.safeTransfer(token0,recipient,amount0) (UniswapV3Pool.sol#859)
        State variables written after the call(s):
        - protocolFees.token1 -= amount1 (UniswapV3Pool.sol#863)
Reentrancy in UniswapV3Pool.increaseObservationCardinalityNext(uint16) (UniswapV3Pool.sol#255-267):
        External calls:
        - observationCardinalityNextNew = observations.grow(observationCardinalityNextOld,observationCardinalityNext) (UniswapV3Pool.sol#262-263)
        State variables written after the call(s):
        - slot0.observationCardinalityNext = observationCardinalityNextNew (UniswapV3Pool.sol#264)
Reentrancy in UniswapV3Pool.initialize(uint160) (UniswapV3Pool.sol#271-289):
        External calls:
        - (cardinality,cardinalityNext) = observations.initialize(_blockTimestamp()) (UniswapV3Pool.sol#276)
```

```
Reentrancy in UniswapV3Pool.initialize(uint160) (UniswapV3Pool.sol#271-289):
        External calls:
        - (cardinality,cardinalityNext) = observations.initialize(_blockTimestamp()) (UniswapV3Pool.sol#276)
        State variables written after the call(s):
        - slot0 = Slot0(sqrtPriceX96,tick,0,cardinality,cardinalityNext,0,true) (UniswapV3Pool.sol#278-286)
Reentrancy in UniswapV3Pool.swap(address,bool,int256,uint160,bytes) (UniswapV3Pool.sol#596-788):
        External calls:
        - (cache.tickCumulative,cache.secondsPerLiquidityCumulativeX128) = observations.observeSingle(cache.blockTimestamp,0,slot0Start.tick,slot0Start.observationIndex,cache.liquidityStart,slot0Start.observationCa
rdinality) (UniswapV3Pool.sol#699-706)
        - (observationIndex,observationCardinality) = observations.write(slot0Start.observationIndex,cache.blockTimestamp,slot0Start.tick,cache.liquidityStart,slot0Start.observationCardinality,slot0Start.observatio
nCardinalityNext) (UniswapV3Pool.sol#734-742)
        State variables written after the call(s):
        - feeGrowthGlobal0X128 = state.feeGrowthGlobalX128 (UniswapV3Pool.sol#760)
        - feeGrowthGlobal1X128 = state.feeGrowthGlobalX128 (UniswapV3Pool.sol#763)
        - liquidity = state.liquidity (UniswapV3Pool.sol#755)
        - (slot0.sqrtPriceX96,slot0.tick,slot0.observationIndex,slot0.observationCardinality) = (state.sqrtPriceX96,state.tick,observationIndex,observationCardinality) (UniswapV3Pool.sol#743-748)
Reentrancy in UniswapV3Pool.swap(address,bool,int256,uint160,bytes) (UniswapV3Pool.sol#596-788):
        External calls:
        - (cache.tickCumulative,cache.secondsPerLiquidityCumulativeX128) = observations.observeSingle(cache.blockTimestamp,0,slot0Start.tick,slot0Start.observationIndex,cache.liquidityStart,slot0Start.observationCa
rdinality) (UniswapV3Pool.sol#699-706)
        State variables written after the call(s):
        - slot0.sqrtPriceX96 = state.sqrtPriceX96 (UniswapV3Pool.sol#751)
Reentrancy in UniswapV3Pool.swap(address,bool,int256,uint160,bytes) (UniswapV3Pool.sol#596-788):
        External calls:
        - (cache.tickCumulative,cache.secondsPerLiquidityCumulativeX128) = observations.observeSingle(cache.blockTimestamp,0,slot0Start.tick,slot0Start.observationIndex,cache.liquidityStart,slot0Start.observationCa
rdinality) (UniswapV3Pool.sol#699-706)
        - (observationIndex,observationCardinality) = observations.write(slot0Start.observationIndex,cache.blockTimestamp,slot0Start.tick,cache.liquidityStart,slot0Start.observationCardinality,slot0Start.observatio
nCardinalityNext) (UniswapV3Pool.sol#734-742)
        - TransferHelper.safeTransfer(token1,recipient,uint256(-amount1)) (UniswapV3Pool.sol#773)
        - balance0Before = balance0() (UniswapV3Pool.sol#775)
                - (success,data) = token0.staticcall(abi.encodeWithSelector(IERC20Minimal.balanceOf.selector,address(this))) (UniswapV3Pool.sol#141-142)
        - IUniswapV3SwapCallback(msg.sender).uniswapV3SwapCallback(amount0,amount1,data) (UniswapV3Pool.sol#776)
        - require(bool,string)(balance0Before.add(uint256(amount0)) <= balance0(),IIA) (UniswapV3Pool.sol#777)
                - (success,data) = token0.staticcall(abi.encodeWithSelector(IERC20Minimal.balanceOf.selector,address(this))) (UniswapV3Pool.sol#141-142)
        - TransferHelper.safeTransfer(token0,recipient,uint256(-amount0)) (UniswapV3Pool.sol#779)
        - balance1Before = balance1() (UniswapV3Pool.sol#781)
                - (success,data) = token1.staticcall(abi.encodeWithSelector(IERC20Minimal.balanceOf.selector,address(this))) (UniswapV3Pool.sol#151-152)
        - IUniswapV3SwapCallback(msg.sender).uniswapV3SwapCallback(amount0,amount1,data) (UniswapV3Pool.sol#782)
        - require(bool,string)(balance1Before.add(uint256(amount1)) <= balance1(),IIA) (UniswapV3Pool.sol#783)
                - (success,data) = token1.staticcall(abi.encodeWithSelector(IERC20Minimal.balanceOf.selector,address(this))) (UniswapV3Pool.sol#151-152)
        State variables written after the call(s):
        - slot0.unlocked = true (UniswapV3Pool.sol#787)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-1

UniswapV3Pool.mint(address,int24,int24,uint128,bytes).balance0Before (UniswapV3Pool.sol#478) is a local variable never initialized
UniswapV3Pool.mint(address,int24,int24,uint128,bytes).balance1Before (UniswapV3Pool.sol#479) is a local variable never initialized
TickBitmap.nextInitializedTickWithinOneWord(mapping(int16 => uint256),int24,int24,bool).wordPos_scope_0 (libraries/TickBitmap.sol#65) is a local variable never initialized
UniswapV3Pool._updatePosition(address,int24,int24,int128,int24).flippedLower (UniswapV3Pool.sol#392) is a local variable never initialized
UniswapV3Pool._updatePosition(address,int24,int24,int128,int24).flippedUpper (UniswapV3Pool.sol#393) is a local variable never initialized
TickBitmap.nextInitializedTickWithinOneWord(mapping(int16 => uint256),int24,int24,bool).bitPos_scope_1 (libraries/TickBitmap.sol#65) is a local variable never initialized
SqrtPriceMath.getNextSqrtPriceFromAmount0RoundingUp(uint160,uint128,uint256,bool).product_scope_0 (libraries/SqrtPriceMath.sol#49) is a local variable never initialized
SqrtPriceMath.getNextSqrtPriceFromAmount0RoundingUp(uint160,uint128,uint256,bool).product (libraries/SqrtPriceMath.sol#39) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

UniswapV3Pool.swap(address,bool,int256,uint160,bytes) (UniswapV3Pool.sol#596-788) has external calls inside a loop: (cache.tickCumulative,cache.secondsPerLiquidityCumulativeX128) = observations.observeSingle(cache.
blockTimestamp,0,slot0Start.tick,slot0Start.observationIndex,cache.liquidityStart,slot0Start.observationCardinality) (UniswapV3Pool.sol#699-706)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation/#calls-inside-a-loop

Variable 'UniswapV3Pool.swap(address,bool,int256,uint160,bytes).step (UniswapV3Pool.sol#642)' in UniswapV3Pool.swap(address,bool,int256,uint160,bytes) (UniswapV3Pool.sol#596-788) potentially used before declaratio
: state.amountSpecifiedRemaining != 0 && state.sqrtPriceX96 != sqrtPriceLimitX96 (UniswapV3Pool.sol#641)
Variable 'TickBitmap.nextInitializedTickWithinOneWord(mapping(int16 => uint256),int24,int24,bool).bitPos (libraries/TickBitmap.sol#52)' in TickBitmap.nextInitializedTickWithinOneWord(mapping(int16 => uint256),int2
,int24,bool) (libraries/TickBitmap.sol#42-77) potentially used before declaration: (wordPos,bitPos) = position(compressed + 1) (libraries/TickBitmap.sol#65)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables

Reentrancy in UniswapV3Pool.flash(address,uint256,uint256,bytes) (UniswapV3Pool.sol#791-834):
        External calls:
        - balance0Before = balance0() (UniswapV3Pool.sol#802)
                - (success,data) = token0.staticcall(abi.encodeWithSelector(IERC20Minimal.balanceOf.selector,address(this))) (UniswapV3Pool.sol#141-142)
        - balance1Before = balance1() (UniswapV3Pool.sol#803)
                - (success,data) = token1.staticcall(abi.encodeWithSelector(IERC20Minimal.balanceOf.selector,address(this))) (UniswapV3Pool.sol#151-152)
        - TransferHelper.safeTransfer(token0,recipient,amount0) (UniswapV3Pool.sol#805)
        - TransferHelper.safeTransfer(token1,recipient,amount1) (UniswapV3Pool.sol#806)
        - IUniswapV3FlashCallback(msg.sender).uniswapV3FlashCallback(fee0,fee1,data) (UniswapV3Pool.sol#808)
        - balance0After = balance0() (UniswapV3Pool.sol#810)
                - (success,data) = token0.staticcall(abi.encodeWithSelector(IERC20Minimal.balanceOf.selector,address(this))) (UniswapV3Pool.sol#141-142)
        - balance1After = balance1() (UniswapV3Pool.sol#811)
                - (success,data) = token1.staticcall(abi.encodeWithSelector(IERC20Minimal.balanceOf.selector,address(this))) (UniswapV3Pool.sol#151-152)
        State variables written after the call(s):
        - feeGrowthGlobal0X128 += FullMath.mulDiv(paid0 - fees0,FixedPoint128.Q128,_liquidity) (UniswapV3Pool.sol#824)
        - feeGrowthGlobal1X128 += FullMath.mulDiv(paid1 - fees1,FixedPoint128.Q128,_liquidity) (UniswapV3Pool.sol#830)
        - protocolFees.token0 += uint128(fees0) (UniswapV3Pool.sol#823)
        - protocolFees.token1 += uint128(fees1) (UniswapV3Pool.sol#829)
Reentrancy in UniswapV3Pool.swap(address,bool,int256,uint160,bytes) (UniswapV3Pool.sol#596-788):
        External calls:
        - (cache.tickCumulative,cache.secondsPerLiquidityCumulativeX128) = observations.observeSingle(cache.blockTimestamp,0,slot0Start.tick,slot0Start.observationIndex,cache.liquidityStart,slot0Start.observationCa
rdinality) (UniswapV3Pool.sol#699-706)
        - (observationIndex,observationCardinality) = observations.write(slot0Start.observationIndex,cache.blockTimestamp,slot0Start.tick,cache.liquidityStart,slot0Start.observationCardinality,slot0Start.observatio
nCardinalityNext) (UniswapV3Pool.sol#734-742)
        State variables written after the call(s):
        - protocolFees.token0 += state.protocolFee (UniswapV3Pool.sol#761)
        - protocolFees.token1 += state.protocolFee (UniswapV3Pool.sol#764)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2

Reentrancy in UniswapV3Pool.burn(int24,int24,uint128) (UniswapV3Pool.sol#517-543):
        External calls:
        - (position,amount0Int,amount1Int) = _modifyPosition(ModifyPositionParams(msg.sender,tickLower,tickUpper,-int256(amount).toInt128())) (UniswapV3Pool.sol#522-530)
                - (slot0.observationIndex,slot0.observationCardinality) = observations.write(_slot0.observationIndex,_blockTimestamp(),_slot0.tick,liquidityBefore,_slot0.observationCardinality,_slot0.observationCa
rdinalityNext) (UniswapV3Pool.sol#341-348)
                - (tickCumulative,secondsPerLiquidityCumulativeX128) = observations.observeSingle(time,0,slot0.tick,slot0.observationIndex,liquidity,slot0.observationCardinality) (UniswapV3Pool.sol#396-404)
        Event emitted after the call(s):
        - Burn(msg.sender,tickLower,tickUpper,amount,amount0,amount1) (UniswapV3Pool.sol#542)
Reentrancy in UniswapV3Pool.collect(address,int24,int24,uint128,uint128) (UniswapV3Pool.sol#490-513):
        External calls:
        - TransferHelper.safeTransfer(token0,recipient,amount0) (UniswapV3Pool.sol#505)
        - TransferHelper.safeTransfer(token1,recipient,amount1) (UniswapV3Pool.sol#509)
        Event emitted after the call(s):
        - Collect(msg.sender,recipient,tickLower,tickUpper,amount0,amount1) (UniswapV3Pool.sol#512)
Reentrancy in UniswapV3Pool.collectProtocol(address,uint128,uint128) (UniswapV3Pool.sol#848-868):
        External calls:
        - TransferHelper.safeTransfer(token0,recipient,amount0) (UniswapV3Pool.sol#859)
        - TransferHelper.safeTransfer(token1,recipient,amount1) (UniswapV3Pool.sol#864)
        Event emitted after the call(s):
        - CollectProtocol(msg.sender,recipient,amount0,amount1) (UniswapV3Pool.sol#867)
Reentrancy in UniswapV3Pool.flash(address,uint256,uint256,bytes) (UniswapV3Pool.sol#791-834):
        External calls:
        - balance0Before = balance0() (UniswapV3Pool.sol#802)
                - (success,data) = token0.staticcall(abi.encodeWithSelector(IERC20Minimal.balanceOf.selector,address(this))) (UniswapV3Pool.sol#141-142)
        - balance1Before = balance1() (UniswapV3Pool.sol#803)
                - (success,data) = token1.staticcall(abi.encodeWithSelector(IERC20Minimal.balanceOf.selector,address(this))) (UniswapV3Pool.sol#151-152)
        - TransferHelper.safeTransfer(token0,recipient,amount0) (UniswapV3Pool.sol#805)
        - TransferHelper.safeTransfer(token1,recipient,amount1) (UniswapV3Pool.sol#806)
        - IUniswapV3FlashCallback(msg.sender).uniswapV3FlashCallback(fee0,fee1,data) (UniswapV3Pool.sol#808)
        - balance0After = balance0() (UniswapV3Pool.sol#810)
                - (success,data) = token0.staticcall(abi.encodeWithSelector(IERC20Minimal.balanceOf.selector,address(this))) (UniswapV3Pool.sol#141-142)
        - balance1After = balance1() (UniswapV3Pool.sol#811)
```

```
                - (success,data) = token1.staticcall(abi.encodeWithSelector(IERC20Minimal.balanceOf.selector,address(this))) (UniswapV3Pool.sol#151-152)
        Event emitted after the call(s):
        - Flash(msg.sender,recipient,amount0,amount1,paid0,paid1) (UniswapV3Pool.sol#833)
Reentrancy in UniswapV3Pool.increaseObservationCardinalityNext(uint16) (UniswapV3Pool.sol#255-267):
        External calls:
        - observationCardinalityNextNew = observations.grow(observationCardinalityNextOld,observationCardinalityNext) (UniswapV3Pool.sol#262-263)
        Event emitted after the call(s):
        - IncreaseObservationCardinalityNext(observationCardinalityNextOld,observationCardinalityNextNew) (UniswapV3Pool.sol#266)
Reentrancy in UniswapV3Pool.initialize(uint160) (UniswapV3Pool.sol#271-289):
        External calls:
        - (cardinality,cardinalityNext) = observations.initialize(_blockTimestamp()) (UniswapV3Pool.sol#276)
        Event emitted after the call(s):
        - Initialize(sqrtPriceX96,tick) (UniswapV3Pool.sol#288)
Reentrancy in UniswapV3Pool.mint(address,int24,int24,uint128,bytes) (UniswapV3Pool.sol#457-487):
        External calls:
        - (amount0Int,amount1Int) = _modifyPosition(ModifyPositionParams(recipient,tickLower,tickUpper,int256(amount).toInt128())) (UniswapV3Pool.sol#465-473)
                - (slot0.observationIndex,slot0.observationCardinality) = observations.write(_slot0.observationIndex,_blockTimestamp(),_slot0.tick,liquidityBefore,_slot0.observationCardinality,_slot0.observationCar
dinalityNext) (UniswapV3Pool.sol#341-348)
                - (tickCumulative,secondsPerLiquidityCumulativeX128) = observations.observeSingle(time,0,slot0.tick,slot0.observationIndex,liquidity,slot0.observationCardinality) (UniswapV3Pool.sol#396-404)
        - balance0Before = balance0() (UniswapV3Pool.sol#480)
                - (success,data) = token0.staticcall(abi.encodeWithSelector(IERC20Minimal.balanceOf.selector,address(this))) (UniswapV3Pool.sol#141-142)
        - balance1Before = balance1() (UniswapV3Pool.sol#481)
                - (success,data) = token1.staticcall(abi.encodeWithSelector(IERC20Minimal.balanceOf.selector,address(this))) (UniswapV3Pool.sol#151-152)
        - IUniswapV3MintCallback(msg.sender).uniswapV3MintCallback(amount0,amount1,data) (UniswapV3Pool.sol#482)
        - require(bool,string)(balance0Before.add(amount0) <= balance0(),M0) (UniswapV3Pool.sol#483)
                - (success,data) = token0.staticcall(abi.encodeWithSelector(IERC20Minimal.balanceOf.selector,address(this))) (UniswapV3Pool.sol#141-142)
        - require(bool,string)(balance1Before.add(amount1) <= balance1(),M1) (UniswapV3Pool.sol#484)
                - (success,data) = token1.staticcall(abi.encodeWithSelector(IERC20Minimal.balanceOf.selector,address(this))) (UniswapV3Pool.sol#151-152)
        Event emitted after the call(s):
        - Mint(msg.sender,recipient,tickLower,tickUpper,amount,amount0,amount1) (UniswapV3Pool.sol#486)
Reentrancy in UniswapV3Pool.swap(address,bool,int256,uint160,bytes) (UniswapV3Pool.sol#596-788):
        External calls:
        - (cache.tickCumulative,cache.secondsPerLiquidityCumulativeX128) = observations.observeSingle(cache.blockTimestamp,0,slot0Start.tick,slot0Start.observationIndex,cache.liquidityStart,slot0Start.observationCa
rdinality) (UniswapV3Pool.sol#699-706)
        - (observationIndex,observationCardinality) = observations.write(slot0Start.observationIndex,cache.blockTimestamp,slot0Start.tick,cache.liquidityStart,slot0Start.observationCardinality,slot0Start.observatio
nCardinalityNext) (UniswapV3Pool.sol#734-742)
        - TransferHelper.safeTransfer(token1,recipient,uint256(-amount1)) (UniswapV3Pool.sol#773)
        - balance0Before = balance0() (UniswapV3Pool.sol#775)
                - (success,data) = token0.staticcall(abi.encodeWithSelector(IERC20Minimal.balanceOf.selector,address(this))) (UniswapV3Pool.sol#141-142)
        - IUniswapV3SwapCallback(msg.sender).uniswapV3SwapCallback(amount0,amount1,data) (UniswapV3Pool.sol#776)
        - require(bool,string)(balance0Before.add(uint256(amount0)) <= balance0(),IIA) (UniswapV3Pool.sol#777)
                - (success,data) = token0.staticcall(abi.encodeWithSelector(IERC20Minimal.balanceOf.selector,address(this))) (UniswapV3Pool.sol#141-142)
        - TransferHelper.safeTransfer(token0,recipient,uint256(-amount0)) (UniswapV3Pool.sol#779)
        - balance1Before = balance1() (UniswapV3Pool.sol#781)
                - (success,data) = token1.staticcall(abi.encodeWithSelector(IERC20Minimal.balanceOf.selector,address(this))) (UniswapV3Pool.sol#151-152)
        - IUniswapV3SwapCallback(msg.sender).uniswapV3SwapCallback(amount0,amount1,data) (UniswapV3Pool.sol#782)
        - require(bool,string)(balance1Before.add(uint256(amount1)) <= balance1(),IIA) (UniswapV3Pool.sol#783)
                - (success,data) = token1.staticcall(abi.encodeWithSelector(IERC20Minimal.balanceOf.selector,address(this))) (UniswapV3Pool.sol#151-152)
        Event emitted after the call(s):
        - Swap(msg.sender,recipient,amount0,amount1,state.sqrtPriceX96,state.liquidity,state.tick) (UniswapV3Pool.sol#786)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

UniswapV3Pool.initialize(uint160) (UniswapV3Pool.sol#271-289) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(slot0.sqrtPriceX96 == 0,AI) (UniswapV3Pool.sol#272)
UniswapV3Pool.swap(address,bool,int256,uint160,bytes) (UniswapV3Pool.sol#596-788) uses timestamp for comparisons
        Dangerous comparisons:
        - state.amountSpecifiedRemaining != 0 && state.sqrtPriceX96 != sqrtPriceLimitX96 (UniswapV3Pool.sol#641)
        - step.tickNext < TickMath.MIN_TICK (UniswapV3Pool.sol#653)
        - step.tickNext > TickMath.MAX_TICK (UniswapV3Pool.sol#655)
        - cache.feeProtocol > 0 (UniswapV3Pool.sol#682)
        - state.liquidity > 0 (UniswapV3Pool.sol#689)
        - state.sqrtPriceX96 == step.sqrtPriceNextX96 (UniswapV3Pool.sol#693)
        - state.sqrtPriceX96 != step.sqrtPriceStartX96 (UniswapV3Pool.sol#726)
        - state.tick != slot0Start.tick (UniswapV3Pool.sol#733)
        - cache.liquidityStart != state.liquidity (UniswapV3Pool.sol#755)
        - state.protocolFee > 0 (UniswapV3Pool.sol#761)
        - state.protocolFee > 0 (UniswapV3Pool.sol#764)
        - amount1 < 0 (UniswapV3Pool.sol#773)
        - require(bool,string)(balance0Before.add(uint256(amount0)) <= balance0(),IIA) (UniswapV3Pool.sol#777)
        - amount0 < 0 (UniswapV3Pool.sol#779)
        - require(bool,string)(balance1Before.add(uint256(amount1)) <= balance1(),IIA) (UniswapV3Pool.sol#783)
        - (step.sqrtPriceNextX96 < sqrtPriceLimitX96) (UniswapV3Pool.sol#663-671)
        - (step.sqrtPriceNextX96 > sqrtPriceLimitX96) (UniswapV3Pool.sol#663-671)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

FullMath.mulDiv(uint256,uint256,uint256) (libraries/FullMath.sol#14-106) uses assembly
        - INLINE ASM (libraries/FullMath.sol#26-30)
        - INLINE ASM (libraries/FullMath.sol#35-37)
        - INLINE ASM (libraries/FullMath.sol#52-54)
        - INLINE ASM (libraries/FullMath.sol#56-59)
        - INLINE ASM (libraries/FullMath.sol#66-68)
        - INLINE ASM (libraries/FullMath.sol#71-73)
        - INLINE ASM (libraries/FullMath.sol#77-79)
TickMath.getTickAtSqrtRatio(uint160) (libraries/TickMath.sol#61-204) uses assembly
        - INLINE ASM (libraries/TickMath.sol#69-73)
        - INLINE ASM (libraries/TickMath.sol#74-78)
        - INLINE ASM (libraries/TickMath.sol#79-83)
        - INLINE ASM (libraries/TickMath.sol#84-88)
        - INLINE ASM (libraries/TickMath.sol#89-93)
        - INLINE ASM (libraries/TickMath.sol#94-98)
        - INLINE ASM (libraries/TickMath.sol#99-103)
        - INLINE ASM (libraries/TickMath.sol#104-107)
        - INLINE ASM (libraries/TickMath.sol#114-119)
        - INLINE ASM (libraries/TickMath.sol#120-125)
        - INLINE ASM (libraries/TickMath.sol#126-131)
        - INLINE ASM (libraries/TickMath.sol#132-137)
        - INLINE ASM (libraries/TickMath.sol#138-143)
        - INLINE ASM (libraries/TickMath.sol#144-149)
        - INLINE ASM (libraries/TickMath.sol#150-155)
        - INLINE ASM (libraries/TickMath.sol#156-161)
        - INLINE ASM (libraries/TickMath.sol#162-167)
        - INLINE ASM (libraries/TickMath.sol#168-173)
        - INLINE ASM (libraries/TickMath.sol#174-179)
        - INLINE ASM (libraries/TickMath.sol#180-185)
        - INLINE ASM (libraries/TickMath.sol#186-191)
        - INLINE ASM (libraries/TickMath.sol#192-196)
UnsafeMath.divRoundingUp(uint256,uint256) (libraries/UnsafeMath.sol#12-16) uses assembly
        - INLINE ASM (libraries/UnsafeMath.sol#13-15)
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', '>=0.4.0<0.8.0', '>=0.5.0', '>=0.5.0<0.8.0', '>=0.6.0', '>=0.7.0']
        - =0.7.6 (NoDelegateCall.sol#2)
        - =0.7.6 (UniswapV3Pool.sol#2)
        - >=0.5.0 (interfaces/IERC20Minimal.sol#2)
        - >=0.5.0 (interfaces/IUniswapV3Factory.sol#2)
        - >=0.5.0 (interfaces/IUniswapV3Pool.sol#2)
        - >=0.5.0 (interfaces/IUniswapV3PoolDeployer.sol#2)
        - >=0.5.0 (interfaces/callback/IUniswapV3FlashCallback.sol#2)
        - >=0.5.0 (interfaces/callback/IUniswapV3MintCallback.sol#2)
        - >=0.5.0 (interfaces/callback/IUniswapV3SwapCallback.sol#2)
        - >=0.5.0 (interfaces/pool/IUniswapV3PoolActions.sol#2)
        - >=0.5.0 (interfaces/pool/IUniswapV3PoolDerivedState.sol#2)
```

```
Pragma version>=0.5.0 (libraries/SqrtPriceMath.sol#2) allows old versions
Pragma version>=0.5.0 (libraries/SwapMath.sol#2) allows old versions
Pragma version>=0.5.0<0.8.0 (libraries/Tick.sol#2) is too complex
Pragma version>=0.5.0 (libraries/TickBitmap.sol#2) allows old versions
Pragma version>=0.5.0<0.8.0 (libraries/TickMath.sol#2) is too complex
Pragma version>=0.6.0 (libraries/TransferHelper.sol#2) allows old versions
Pragma version>=0.5.0 (libraries/UnsafeMath.sol#2) allows old versions
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in UniswapV3Pool.balance0() (UniswapV3Pool.sol#140-145):
        - (success,data) = token0.staticcall(abi.encodeWithSelector(IERC20Minimal.balanceOf.selector,address(this))) (UniswapV3Pool.sol#141-142)
Low level call in UniswapV3Pool.balance1() (UniswapV3Pool.sol#150-155):
        - (success,data) = token1.staticcall(abi.encodeWithSelector(IERC20Minimal.balanceOf.selector,address(this))) (UniswapV3Pool.sol#151-152)
Low level call in TransferHelper.safeTransfer(address,address,uint256) (libraries/TransferHelper.sol#14-22):
        - (success,data) = token.call(abi.encodeWithSelector(IERC20Minimal.transfer.selector,to,value)) (libraries/TransferHelper.sol#19-20)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Variable UniswapV3Pool._updatePosition(address,int24,int24,int128,int24)._feeGrowthGlobal0X128 (UniswapV3Pool.sol#388) is too similar to UniswapV3Pool._updatePosition(address,int24,int24,int128,int24)._feeGrowthGlo
bal1X128 (UniswapV3Pool.sol#389)
Variable UniswapV3Pool.collect(address,int24,int24,uint128,uint128).amount0Requested (UniswapV3Pool.sol#494) is too similar to UniswapV3Pool.collect(address,int24,int24,uint128,uint128).amount1Requested (UniswapV3P
ool.sol#495)
Variable UniswapV3Pool.collect(address,int24,int24,uint128,uint128).amount0Requested (UniswapV3Pool.sol#494) is too similar to IUniswapV3PoolOwnerActions.collectProtocol(address,uint128,uint128).amount1Requested (i
nterfaces/pool/IUniswapV3PoolOwnerActions.sol#21)
Variable IUniswapV3PoolActions.collect(address,int24,int24,uint128,uint128).amount0Requested (interfaces/pool/IUniswapV3PoolActions.sol#47) is too similar to IUniswapV3PoolActions.collect(address,int24,int24,uint12
8,uint128).amount1Requested (interfaces/pool/IUniswapV3PoolActions.sol#48)
Variable IUniswapV3PoolOwnerActions.collectProtocol(address,uint128,uint128).amount0Requested (interfaces/pool/IUniswapV3PoolOwnerActions.sol#20) is too similar to IUniswapV3PoolOwnerActions.collectProtocol(address
,uint128,uint128).amount1Requested (interfaces/pool/IUniswapV3PoolOwnerActions.sol#21)
Variable IUniswapV3PoolActions.collect(address,int24,int24,uint128,uint128).amount0Requested (interfaces/pool/IUniswapV3PoolActions.sol#47) is too similar to UniswapV3Pool.collect(address,int24,int24,uint128,uint12
8).amount1Requested (UniswapV3Pool.sol#495)
Variable UniswapV3Pool.collectProtocol(address,uint128,uint128).amount0Requested (UniswapV3Pool.sol#850) is too similar to IUniswapV3PoolActions.collect(address,int24,int24,uint128,uint128).amount1Requested (inter
aces/pool/IUniswapV3PoolActions.sol#48)
Variable IUniswapV3PoolActions.collect(address,int24,int24,uint128,uint128).amount0Requested (interfaces/pool/IUniswapV3PoolActions.sol#47) is too similar to IUniswapV3PoolOwnerActions.collectProtocol(address,uint1
28,uint128).amount1Requested (interfaces/pool/IUniswapV3PoolOwnerActions.sol#21)
Variable UniswapV3Pool.collectProtocol(address,uint128,uint128).amount0Requested (UniswapV3Pool.sol#850) is too similar to UniswapV3Pool.collect(address,int24,int24,uint128,uint128).amount1Requested (UniswapV3Pool.
sol#495)
Variable UniswapV3Pool.collectProtocol(address,uint128,uint128).amount0Requested (UniswapV3Pool.sol#850) is too similar to IUniswapV3PoolOwnerActions.collectProtocol(address,uint128,uint128).amount1Requested (inter
faces/pool/IUniswapV3PoolOwnerActions.sol#21)
Variable UniswapV3Pool.collectProtocol(address,uint128,uint128).amount0Requested (UniswapV3Pool.sol#850) is too similar to UniswapV3Pool.collectProtocol(address,uint128,uint128).amount1Requested (UniswapV3Pool.sol#
851)
Variable IUniswapV3PoolOwnerActions.collectProtocol(address,uint128,uint128).amount0Requested (interfaces/pool/IUniswapV3PoolOwnerActions.sol#20) is too similar to IUniswapV3PoolActions.collect(address,int24,int24,
uint128,uint128).amount1Requested (interfaces/pool/IUniswapV3PoolActions.sol#48)
Variable IUniswapV3PoolOwnerActions.collectProtocol(address,uint128,uint128).amount0Requested (interfaces/pool/IUniswapV3PoolOwnerActions.sol#20) is too similar to UniswapV3Pool.collect(address,int24,int24,uint128,
uint128).amount1Requested (UniswapV3Pool.sol#495)
Variable UniswapV3Pool.collect(address,int24,int24,uint128,uint128).amount0Requested (UniswapV3Pool.sol#494) is too similar to IUniswapV3PoolActions.collect(address,int24,int24,uint128,uint128).amount1Requested (in
terfaces/pool/IUniswapV3PoolActions.sol#48)
Variable IUniswapV3PoolActions.collect(address,int24,int24,uint128,uint128).amount0Requested (interfaces/pool/IUniswapV3PoolActions.sol#47) is too similar to UniswapV3Pool.collectProtocol(address,uint128,uint128).a
mount1Requested (UniswapV3Pool.sol#851)
Variable UniswapV3Pool.collect(address,int24,int24,uint128,uint128).amount0Requested (UniswapV3Pool.sol#494) is too similar to UniswapV3Pool.collectProtocol(address,uint128,uint128).amount1Requested (UniswapV3Pool.
sol#851)
Variable IUniswapV3PoolOwnerActions.collectProtocol(address,uint128,uint128).amount0Requested (interfaces/pool/IUniswapV3PoolOwnerActions.sol#20) is too similar to UniswapV3Pool.collectProtocol(address,uint128,uint
128).amount1Requested (UniswapV3Pool.sol#851)
Variable UniswapV3Pool.swap(address,bool,int256,uint160,bytes).balance0Before (UniswapV3Pool.sol#775) is too similar to UniswapV3Pool.swap(address,bool,int256,uint160,bytes).balance1Before (UniswapV3Pool.sol#781)
Variable UniswapV3Pool.mint(address,int24,int24,uint128,bytes).balance0Before (UniswapV3Pool.sol#478) is too similar to UniswapV3Pool.flash(address,uint256,uint256,bytes).balance1Before (UniswapV3Pool.sol#803)
Variable UniswapV3Pool.swap(address,bool,int256,uint160,bytes).balance0Before (UniswapV3Pool.sol#775) is too similar to UniswapV3Pool.mint(address,int24,int24,uint128,bytes).balance1Before (UniswapV3Pool.sol#479)
Variable UniswapV3Pool.flash(address,uint256,uint256,bytes).balance0Before (UniswapV3Pool.sol#802) is too similar to UniswapV3Pool.flash(address,uint256,uint256,bytes).balance1Before (UniswapV3Pool.sol#803)
Variable UniswapV3Pool.mint(address,int24,int24,uint128,bytes).balance0Before (UniswapV3Pool.sol#478) is too similar to UniswapV3Pool.mint(address,int24,int24,uint128,bytes).balance1Before (UniswapV3Pool.sol#479)
Variable UniswapV3Pool.swap(address,bool,int256,uint160,bytes).balance0Before (UniswapV3Pool.sol#775) is too similar to UniswapV3Pool.flash(address,uint256,uint256,bytes).balance1Before (UniswapV3Pool.sol#803)
Variable UniswapV3Pool.flash(address,uint256,uint256,bytes).balance0After (UniswapV3Pool.sol#810) is too similar to UniswapV3Pool.flash(address,uint256,uint256,bytes).balance1After (UniswapV3Pool.sol#811)
        - >=0.5.0 (interfaces/IUniswapV3Pool.sol#2)
        - >=0.5.0 (interfaces/IUniswapV3PoolDeployer.sol#2)
        - >=0.5.0 (interfaces/callback/IUniswapV3FlashCallback.sol#2)
        - >=0.5.0 (interfaces/callback/IUniswapV3MintCallback.sol#2)
        - >=0.5.0 (interfaces/callback/IUniswapV3SwapCallback.sol#2)
        - >=0.5.0 (interfaces/pool/IUniswapV3PoolActions.sol#2)
        - >=0.5.0 (interfaces/pool/IUniswapV3PoolDerivedState.sol#2)
        - >=0.5.0 (interfaces/pool/IUniswapV3PoolEvents.sol#2)
        - >=0.5.0 (interfaces/pool/IUniswapV3PoolImmutables.sol#2)
        - >=0.5.0 (interfaces/pool/IUniswapV3PoolOwnerActions.sol#2)
        - >=0.5.0 (interfaces/pool/IUniswapV3PoolState.sol#2)
        - >=0.5.0 (libraries/BitMath.sol#2)
        - >=0.4.0 (libraries/FixedPoint128.sol#2)
        - >=0.4.0 (libraries/FixedPoint96.sol#2)
        - >=0.4.0<0.8.0 (libraries/FullMath.sol#2)
        - >=0.5.0 (libraries/LiquidityMath.sol#2)
        - >=0.7.0 (libraries/LowGasSafeMath.sol#2)
        - >=0.5.0<0.8.0 (libraries/Oracle.sol#2)
        - >=0.5.0<0.8.0 (libraries/Position.sol#2)
        - >=0.5.0 (libraries/SafeCast.sol#2)
        - >=0.5.0 (libraries/SqrtPriceMath.sol#2)
        - >=0.5.0 (libraries/SwapMath.sol#2)
        - >=0.5.0<0.8.0 (libraries/Tick.sol#2)
        - >=0.5.0 (libraries/TickBitmap.sol#2)
        - >=0.5.0<0.8.0 (libraries/TickMath.sol#2)
        - >=0.6.0 (libraries/TransferHelper.sol#2)
        - >=0.5.0 (libraries/UnsafeMath.sol#2)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

LowGasSafeMath.mul(uint256,uint256) (libraries/LowGasSafeMath.sol#27-29) is never used and should be removed
LowGasSafeMath.sub(uint256,uint256) (libraries/LowGasSafeMath.sol#19-21) is never used and should be removed
Oracle.binarySearch(Oracle.Observation[65535],uint32,uint32,uint16,uint16) (libraries/Oracle.sol#153-184) is never used and should be removed
Oracle.getSurroundingObservations(Oracle.Observation[65535],uint32,uint32,int24,uint16,uint128,uint16) (libraries/Oracle.sol#198-230) is never used and should be removed
Oracle.grow(Oracle.Observation[65535],uint16,uint16) (libraries/Oracle.sol#108-120) is never used and should be removed
Oracle.initialize(Oracle.Observation[65535],uint32) (libraries/Oracle.sol#52-63) is never used and should be removed
Oracle.lte(uint32,uint32,uint32) (libraries/Oracle.sol#128-140) is never used and should be removed
Oracle.observe(Oracle.Observation[65535],uint32,uint32[],int24,uint16,uint128,uint16) (libraries/Oracle.sol#300-324) is never used and should be removed
Oracle.observeSingle(Oracle.Observation[65535],uint32,uint32,int24,uint16,uint128,uint16) (libraries/Oracle.sol#245-287) is never used and should be removed
Oracle.transform(Oracle.Observation,uint32,int24,uint128) (libraries/Oracle.sol#30-45) is never used and should be removed
Oracle.write(Oracle.Observation[65535],uint16,uint32,int24,uint128,uint16,uint16) (libraries/Oracle.sol#78-101) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version>=0.5.0 (interfaces/IERC20Minimal.sol#2) allows old versions
Pragma version>=0.5.0 (interfaces/IUniswapV3Factory.sol#2) allows old versions
Pragma version>=0.5.0 (interfaces/IUniswapV3Pool.sol#2) allows old versions
Pragma version>=0.5.0 (interfaces/IUniswapV3PoolDeployer.sol#2) allows old versions
Pragma version>=0.5.0 (interfaces/callback/IUniswapV3FlashCallback.sol#2) allows old versions
Pragma version>=0.5.0 (interfaces/callback/IUniswapV3MintCallback.sol#2) allows old versions
Pragma version>=0.5.0 (interfaces/callback/IUniswapV3SwapCallback.sol#2) allows old versions
Pragma version>=0.5.0 (interfaces/pool/IUniswapV3PoolActions.sol#2) allows old versions
Pragma version>=0.5.0 (interfaces/pool/IUniswapV3PoolDerivedState.sol#2) allows old versions
Pragma version>=0.5.0 (interfaces/pool/IUniswapV3PoolEvents.sol#2) allows old versions
Pragma version>=0.5.0 (interfaces/pool/IUniswapV3PoolImmutables.sol#2) allows old versions
Pragma version>=0.5.0 (interfaces/pool/IUniswapV3PoolOwnerActions.sol#2) allows old versions
Pragma version>=0.5.0 (interfaces/pool/IUniswapV3PoolState.sol#2) allows old versions
Pragma version>=0.5.0 (libraries/BitMath.sol#2) allows old versions
Pragma version>=0.4.0 (libraries/FixedPoint128.sol#2) allows old versions
Pragma version>=0.4.0 (libraries/FixedPoint96.sol#2) allows old versions
Pragma version>=0.4.0<0.8.0 (libraries/FullMath.sol#2) is too complex
Pragma version>=0.5.0 (libraries/LiquidityMath.sol#2) allows old versions
Pragma version>=0.7.0 (libraries/LowGasSafeMath.sol#2) allows old versions
Pragma version>=0.5.0<0.8.0 (libraries/Oracle.sol#2) is too complex
Pragma version>=0.5.0<0.8.0 (libraries/Position.sol#2) is too complex
```

```
Variable UniswapV3Pool.setFeeProtocol(uint8,uint8).feeProtocol0 (UniswapV3Pool.sol#837) is too similar to IUniswapV3PoolOwnerActions.setFeeProtocol(uint8,uint8).feeProtocol1 (interfaces/pool/IUniswapV3PoolOwnerAct
ons.sol#10)
Variable IUniswapV3PoolOwnerActions.setFeeProtocol(uint8,uint8).feeProtocol0 (interfaces/pool/IUniswapV3PoolOwnerActions.sol#10) is too similar to UniswapV3Pool.setFeeProtocol(uint8,uint8).feeProtocol1 (UniswapV3Po
ol.sol#837)
Variable UniswapV3Pool.setFeeProtocol(uint8,uint8).feeProtocol0 (UniswapV3Pool.sol#837) is too similar to UniswapV3Pool.setFeeProtocol(uint8,uint8).feeProtocol1 (UniswapV3Pool.sol#837)
Variable UniswapV3Pool.setFeeProtocol(uint8,uint8).feeProtocol0 (UniswapV3Pool.sol#837) is too similar to UniswapV3Pool.flash(address,uint256,uint256,bytes).feeProtocol1 (UniswapV3Pool.sol#827)
Variable UniswapV3Pool.snapshotCumulativesInside(int24,int24).secondsPerLiquidityOutsideLowerX128 (UniswapV3Pool.sol#173) is too similar to UniswapV3Pool.snapshotCumulativesInside(int24,int24).secondsPerLiquidityO
tsideUpperX128 (UniswapV3Pool.sol#174)
Variable IUniswapV3PoolState.positions(bytes32).tokensOwed0 (interfaces/pool/IUniswapV3PoolState.sol#95) is too similar to IUniswapV3PoolState.positions(bytes32).tokensOwed1 (interfaces/pool/IUniswapV3PoolState.sol
#96)
Variable IUniswapV3MintCallback.uniswapV3MintCallback(uint256,uint256,bytes).amount0Owed (interfaces/callback/IUniswapV3MintCallback.sol#14) is too similar to IUniswapV3MintCallback.uniswapV3MintCallback(uint256,ui
nt256,bytes).amount1Owed (interfaces/callback/IUniswapV3MintCallback.sol#15)
Variable IUniswapV3SwapCallback.uniswapV3SwapCallback(int256,int256,bytes).amount0Delta (interfaces/callback/IUniswapV3SwapCallback.sol#17) is too similar to IUniswapV3SwapCallback.uniswapV3SwapCallback(int256,int2
56,bytes).amount1Delta (interfaces/callback/IUniswapV3SwapCallback.sol#18)
Variable Position.update(Position.Info,int128,uint256,uint256).feeGrowthInside0X128 (libraries/Position.sol#47) is too similar to Position.update(Position.Info,int128,uint256,uint256).feeGrowthInside1X128 (librarie
s/Position.sol#48)
Variable Position.update(Position.Info,int128,uint256,uint256).tokensOwed0 (libraries/Position.sol#61-68) is too similar to Position.update(Position.Info,int128,uint256,uint256).tokensOwed1 (libraries/Position.sol#
69-76)
Variable SqrtPriceMath.getAmount1Delta(uint160,uint160,uint128,bool).sqrtRatioAX96 (libraries/SqrtPriceMath.sol#183) is too similar to SqrtPriceMath.getAmount0Delta(uint160,uint160,int128).sqrtRatioBX96 (libraries/
SqrtPriceMath.sol#203)
Variable SqrtPriceMath.getAmount1Delta(uint160,uint160,uint128,bool).sqrtRatioAX96 (libraries/SqrtPriceMath.sol#183) is too similar to SqrtPriceMath.getAmount1Delta(uint160,uint160,int128).sqrtRatioBX96 (libraries/
SqrtPriceMath.sol#219)
Variable SqrtPriceMath.getAmount0Delta(uint160,uint160,int128).sqrtRatioAX96 (libraries/SqrtPriceMath.sol#202) is too similar to SqrtPriceMath.getAmount0Delta(uint160,uint160,uint128,bool).sqrtRatioBX96 (libraries/
SqrtPriceMath.sol#155)
Variable SqrtPriceMath.getAmount0Delta(uint160,uint160,uint128,bool).sqrtRatioAX96 (libraries/SqrtPriceMath.sol#154) is too similar to SqrtPriceMath.getAmount0Delta(uint160,uint160,uint128,bool).sqrtRatioBX96 (libr
aries/SqrtPriceMath.sol#155)
Variable SqrtPriceMath.getAmount0Delta(uint160,uint160,uint128,bool).sqrtRatioAX96 (libraries/SqrtPriceMath.sol#154) is too similar to SqrtPriceMath.getAmount1Delta(uint160,uint160,int128).sqrtRatioBX96 (libraries/
SqrtPriceMath.sol#219)
Variable SqrtPriceMath.getAmount0Delta(uint160,uint160,int128).sqrtRatioAX96 (libraries/SqrtPriceMath.sol#202) is too similar to SqrtPriceMath.getAmount0Delta(uint160,uint160,int128).sqrtRatioBX96 (libraries/SqrtPr
iceMath.sol#203)
Variable SqrtPriceMath.getAmount0Delta(uint160,uint160,int128).sqrtRatioAX96 (libraries/SqrtPriceMath.sol#202) is too similar to SqrtPriceMath.getAmount1Delta(uint160,uint160,int128).sqrtRatioBX96 (libraries/SqrtPr
iceMath.sol#219)
Variable SqrtPriceMath.getAmount1Delta(uint160,uint160,uint128,bool).sqrtRatioAX96 (libraries/SqrtPriceMath.sol#183) is too similar to SqrtPriceMath.getAmount0Delta(uint160,uint160,uint128,bool).sqrtRatioBX96 (libr
aries/SqrtPriceMath.sol#155)
Variable SqrtPriceMath.getAmount1Delta(uint160,uint160,int128).sqrtRatioAX96 (libraries/SqrtPriceMath.sol#218) is too similar to SqrtPriceMath.getAmount1Delta(uint160,uint160,int128).sqrtRatioBX96 (libraries/SqrtPr
iceMath.sol#219)
Variable SqrtPriceMath.getAmount1Delta(uint160,uint160,uint128,bool).sqrtRatioAX96 (libraries/SqrtPriceMath.sol#183) is too similar to SqrtPriceMath.getAmount1Delta(uint160,uint160,uint128,bool).sqrtRatioBX96 (libr
aries/SqrtPriceMath.sol#184)
Variable SqrtPriceMath.getAmount0Delta(uint160,uint160,int128).sqrtRatioAX96 (libraries/SqrtPriceMath.sol#202) is too similar to SqrtPriceMath.getAmount1Delta(uint160,uint160,uint128,bool).sqrtRatioBX96 (libraries/
SqrtPriceMath.sol#184)
Variable SqrtPriceMath.getAmount1Delta(uint160,uint160,int128).sqrtRatioAX96 (libraries/SqrtPriceMath.sol#218) is too similar to SqrtPriceMath.getAmount0Delta(uint160,uint160,int128).sqrtRatioBX96 (libraries/SqrtPr
iceMath.sol#203)
Variable SqrtPriceMath.getAmount0Delta(uint160,uint160,uint128,bool).sqrtRatioAX96 (libraries/SqrtPriceMath.sol#154) is too similar to SqrtPriceMath.getAmount0Delta(uint160,uint160,int128).sqrtRatioBX96 (libraries/
SqrtPriceMath.sol#203)
Variable SqrtPriceMath.getAmount1Delta(uint160,uint160,int128).sqrtRatioAX96 (libraries/SqrtPriceMath.sol#218) is too similar to SqrtPriceMath.getAmount1Delta(uint160,uint160,uint128,bool).sqrtRatioBX96 (libraries/
SqrtPriceMath.sol#184)
Variable SqrtPriceMath.getAmount0Delta(uint160,uint160,uint128,bool).sqrtRatioAX96 (libraries/SqrtPriceMath.sol#154) is too similar to SqrtPriceMath.getAmount1Delta(uint160,uint160,uint128,bool).sqrtRatioBX96 (libr
aries/SqrtPriceMath.sol#184)
Variable SqrtPriceMath.getAmount1Delta(uint160,uint160,int128).sqrtRatioAX96 (libraries/SqrtPriceMath.sol#218) is too similar to SqrtPriceMath.getAmount0Delta(uint160,uint160,uint128,bool).sqrtRatioBX96 (libraries/
SqrtPriceMath.sol#155)
Variable Tick.getFeeGrowthInside(mapping(int24 => Tick.Info),int24,int24,int24,uint256,uint256).feeGrowthAbove0X128 (libraries/Tick.sol#83) is too similar to Tick.getFeeGrowthInside(mapping(int24 => Tick.Info),int2
4,int24,int24,uint256,uint256).feeGrowthAbove1X128 (libraries/Tick.sol#84)
Variable Tick.getFeeGrowthInside(mapping(int24 => Tick.Info),int24,int24,int24,uint256,uint256).feeGrowthBelow0X128 (libraries/Tick.sol#72) is too similar to Tick.getFeeGrowthInside(mapping(int24 => Tick.Info),int2
4,int24,int24,uint256,uint256).feeGrowthBelow1X128 (libraries/Tick.sol#73)
Variable Tick.getFeeGrowthInside(mapping(int24 => Tick.Info),int24,int24,int24,uint256,uint256).feeGrowthGlobal0X128 (libraries/Tick.sol#65) is too similar to Tick.getFeeGrowthInside(mapping(int24 => Tick.Info),int
24,int24,int24,uint256,uint256).feeGrowthGlobal1X128 (libraries/Tick.sol#66)
Variable Tick.cross(mapping(int24 => Tick.Info),int24,uint256,uint256,uint160,int56,uint32).feeGrowthGlobal0X128 (libraries/Tick.sol#171) is too similar to Tick.cross(mapping(int24 => Tick.Info),int24,uint256,uint2
56,uint160,int56,uint32).feeGrowthGlobal1X128 (libraries/Tick.sol#172)
Variable Tick.update(mapping(int24 => Tick.Info),int24,int24,int128,uint256,uint256,uint160,int56,uint32,bool,uint128).feeGrowthGlobal0X128 (libraries/Tick.sol#115) is too similar to Tick.getFeeGrowthInside(mapping
(int24 => Tick.Info),int24,int24,int24,uint256,uint256).feeGrowthGlobal1X128 (libraries/Tick.sol#66)
Variable Tick.cross(mapping(int24 => Tick.Info),int24,uint256,uint256,uint160,int56,uint32).feeGrowthGlobal0X128 (libraries/Tick.sol#171) is too similar to Tick.getFeeGrowthInside(mapping(int24 => Tick.Info),int24,
int24,int24,uint256,uint256).feeGrowthGlobal1X128 (libraries/Tick.sol#66)
Variable Tick.update(mapping(int24 => Tick.Info),int24,int24,int128,uint256,uint256,uint160,int56,uint32,bool,uint128).feeGrowthGlobal0X128 (libraries/Tick.sol#115) is too similar to Tick.cross(mapping(int24 => Tic
```

```
        State variables written after the call(s):
        - slot0.sqrtPriceX96 = state.sqrtPriceX96 (UniswapV3Pool.sol#751)
Reentrancy in UniswapV3Pool.swap(address,bool,int256,uint160,bytes) (UniswapV3Pool.sol#596-788):
        External calls:
        - (cache.tickCumulative,cache.secondsPerLiquidityCumulativeX128) = observations.observeSingle(cache.blockTimestamp,0,slot0Start.tick,slot0Start.observationIndex,cache.liquidityStart,slot0Start.observationCa
rdinality) (UniswapV3Pool.sol#699-706)
        - (observationIndex,observationCardinality) = observations.write(slot0Start.observationIndex,cache.blockTimestamp,slot0Start.tick,cache.liquidityStart,slot0Start.observationCardinality,slot0Start.observatio
nCardinalityNext) (UniswapV3Pool.sol#734-742)
        - TransferHelper.safeTransfer(token1,recipient,uint256(-amount1)) (UniswapV3Pool.sol#773)
        - balance0Before = balance0() (UniswapV3Pool.sol#775)
                - (success,data) = token0.staticcall(abi.encodeWithSelector(IERC20Minimal.balanceOf.selector,address(this))) (UniswapV3Pool.sol#141-142)
        - IUniswapV3SwapCallback(msg.sender).uniswapV3SwapCallback(amount0,amount1,data) (UniswapV3Pool.sol#776)
        - require(bool,string)(balance0Before.add(uint256(amount0)) <= balance0(),IIA) (UniswapV3Pool.sol#777)
                - (success,data) = token0.staticcall(abi.encodeWithSelector(IERC20Minimal.balanceOf.selector,address(this))) (UniswapV3Pool.sol#141-142)
        - TransferHelper.safeTransfer(token0,recipient,uint256(-amount0)) (UniswapV3Pool.sol#779)
        - balance1Before = balance1() (UniswapV3Pool.sol#781)
                - (success,data) = token1.staticcall(abi.encodeWithSelector(IERC20Minimal.balanceOf.selector,address(this))) (UniswapV3Pool.sol#151-152)
        - IUniswapV3SwapCallback(msg.sender).uniswapV3SwapCallback(amount0,amount1,data) (UniswapV3Pool.sol#782)
        - require(bool,string)(balance1Before.add(uint256(amount1)) <= balance1(),IIA) (UniswapV3Pool.sol#783)
                - (success,data) = token1.staticcall(abi.encodeWithSelector(IERC20Minimal.balanceOf.selector,address(this))) (UniswapV3Pool.sol#151-152)
        State variables written after the call(s):
        - slot0.unlocked = true (UniswapV3Pool.sol#787)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-1

SqrtPriceMath.getNextSqrtPriceFromAmount0RoundingUp(uint160,uint128,uint256,bool).product_scope_0 (libraries/SqrtPriceMath.sol#49) is a local variable never initialized
TickBitmap.nextInitializedTickWithinOneWord(mapping(int16 => uint256),int24,int24,bool).bitPos_scope_1 (libraries/TickBitmap.sol#65) is a local variable never initialized
UniswapV3Pool.mint(address,int24,int24,uint128,bytes).balance0Before (UniswapV3Pool.sol#478) is a local variable never initialized
UniswapV3Pool._updatePosition(address,int24,int24,int128,int24).flippedUpper (UniswapV3Pool.sol#393) is a local variable never initialized
TickBitmap.nextInitializedTickWithinOneWord(mapping(int16 => uint256),int24,int24,bool).wordPos_scope_0 (libraries/TickBitmap.sol#65) is a local variable never initialized
SqrtPriceMath.getNextSqrtPriceFromAmount0RoundingUp(uint160,uint128,uint256,bool).product (libraries/SqrtPriceMath.sol#39) is a local variable never initialized
UniswapV3Pool.mint(address,int24,int24,uint128,bytes).balance1Before (UniswapV3Pool.sol#479) is a local variable never initialized
UniswapV3Pool._updatePosition(address,int24,int24,int128,int24).flippedLower (UniswapV3Pool.sol#392) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

UniswapV3Pool.swap(address,bool,int256,uint160,bytes) (UniswapV3Pool.sol#596-788) has external calls inside a loop: (cache.tickCumulative,cache.secondsPerLiquidityCumulativeX128) = observations.observeSingle(cache.
blockTimestamp,0,slot0Start.tick,slot0Start.observationIndex,cache.liquidityStart,slot0Start.observationCardinality) (UniswapV3Pool.sol#699-706)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation/#calls-inside-a-loop

Variable 'UniswapV3Pool.swap(address,bool,int256,uint160,bytes).step (UniswapV3Pool.sol#642)' in UniswapV3Pool.swap(address,bool,int256,uint160,bytes) (UniswapV3Pool.sol#596-788) potentially used before declaratio
: state.amountSpecifiedRemaining != 0 && state.sqrtPriceX96 != sqrtPriceLimitX96 (UniswapV3Pool.sol#641)
Variable 'TickBitmap.nextInitializedTickWithinOneWord(mapping(int16 => uint256),int24,int24,bool).bitPos (libraries/TickBitmap.sol#52)' in TickBitmap.nextInitializedTickWithinOneWord(mapping(int16 => uint256),int2
,int24,bool) (libraries/TickBitmap.sol#42-77) potentially used before declaration: (wordPos,bitPos) = position(compressed + 1) (libraries/TickBitmap.sol#65)
Variable 'TickBitmap.nextInitializedTickWithinOneWord(mapping(int16 => uint256),int24,int24,bool).wordPos (libraries/TickBitmap.sol#52)' in TickBitmap.nextInitializedTickWithinOneWord(mapping(int16 => uint256),int2
4,int24,bool) (libraries/TickBitmap.sol#42-77) potentially used before declaration: (wordPos,bitPos) = position(compressed + 1) (libraries/TickBitmap.sol#65)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables

Reentrancy in UniswapV3Pool.flash(address,uint256,uint256,bytes) (UniswapV3Pool.sol#791-834):
        External calls:
        - balance0Before = balance0() (UniswapV3Pool.sol#802)
                - (success,data) = token0.staticcall(abi.encodeWithSelector(IERC20Minimal.balanceOf.selector,address(this))) (UniswapV3Pool.sol#141-142)
        - balance1Before = balance1() (UniswapV3Pool.sol#803)
                - (success,data) = token1.staticcall(abi.encodeWithSelector(IERC20Minimal.balanceOf.selector,address(this))) (UniswapV3Pool.sol#151-152)
        - TransferHelper.safeTransfer(token0,recipient,amount0) (UniswapV3Pool.sol#805)
        - TransferHelper.safeTransfer(token1,recipient,amount1) (UniswapV3Pool.sol#806)
        - IUniswapV3FlashCallback(msg.sender).uniswapV3FlashCallback(fee0,fee1,data) (UniswapV3Pool.sol#808)
        - balance0After = balance0() (UniswapV3Pool.sol#810)
                - (success,data) = token0.staticcall(abi.encodeWithSelector(IERC20Minimal.balanceOf.selector,address(this))) (UniswapV3Pool.sol#141-142)
        - balance1After = balance1() (UniswapV3Pool.sol#811)
                - (success,data) = token1.staticcall(abi.encodeWithSelector(IERC20Minimal.balanceOf.selector,address(this))) (UniswapV3Pool.sol#151-152)
        State variables written after the call(s):
        - feeGrowthGlobal0X128 += FullMath.mulDiv(paid0 - fees0,FixedPoint128.Q128,_liquidity) (UniswapV3Pool.sol#824)
        - feeGrowthGlobal1X128 += FullMath.mulDiv(paid1 - fees1,FixedPoint128.Q128,_liquidity) (UniswapV3Pool.sol#830)
        - protocolFees.token0 += uint128(fees0) (UniswapV3Pool.sol#823)
```

```
        -denominator = denominator / twos (libraries/FullMath.sol#67)
        -inv *= 2 - denominator * inv (libraries/FullMath.sol#95)
FullMath.mulDiv(uint256,uint256,uint256) (libraries/FullMath.sol#14-106) performs a multiplication on the result of a division:
        -denominator = denominator / twos (libraries/FullMath.sol#67)
        -inv *= 2 - denominator * inv (libraries/FullMath.sol#96)
FullMath.mulDiv(uint256,uint256,uint256) (libraries/FullMath.sol#14-106) performs a multiplication on the result of a division:
        -prod0 = prod0 / twos (libraries/FullMath.sol#72)
        -result = prod0 * inv (libraries/FullMath.sol#104)
Oracle.observeSingle(Oracle.Observation[65535],uint32,uint32,int24,uint16,uint128,uint16) (libraries/Oracle.sol#245-287) performs a multiplication on the result of a division:
        -(beforeOrAt.tickCumulative + ((atOrAfter.tickCumulative - beforeOrAt.tickCumulative) / observationTimeDelta) * targetDelta,beforeOrAt.secondsPerLiquidityCumulativeX128 + uint160((uint256(atOrAfter.secondsP
erLiquidityCumulativeX128 - beforeOrAt.secondsPerLiquidityCumulativeX128) * targetDelta) / observationTimeDelta)) (libraries/Oracle.sol#275-285)
Tick.tickSpacingToMaxLiquidityPerTick(int24) (libraries/Tick.sol#44-49) performs a multiplication on the result of a division:
        -minTick = (TickMath.MIN_TICK / tickSpacing) * tickSpacing (libraries/Tick.sol#45)
Tick.tickSpacingToMaxLiquidityPerTick(int24) (libraries/Tick.sol#44-49) performs a multiplication on the result of a division:
        -maxTick = (TickMath.MAX_TICK / tickSpacing) * tickSpacing (libraries/Tick.sol#46)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#divide-before-multiply

UniswapV3Pool.initialize(uint160) (UniswapV3Pool.sol#271-289) uses a dangerous strict equality:
        - require(bool,string)(slot0.sqrtPriceX96 == 0,AI) (UniswapV3Pool.sol#272)
UniswapV3Pool.swap(address,bool,int256,uint160,bytes) (UniswapV3Pool.sol#596-788) uses a dangerous strict equality:
        - state.sqrtPriceX96 == step.sqrtPriceNextX96 (UniswapV3Pool.sol#693)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dangerous-strict-equalities

Reentrancy in UniswapV3Pool._modifyPosition(UniswapV3Pool.ModifyPositionParams) (UniswapV3Pool.sol#306-372):
        External calls:
        - position = _updatePosition(params.owner,params.tickLower,params.tickUpper,params.liquidityDelta,_slot0.tick) (UniswapV3Pool.sol#319-325)
                - (tickCumulative,secondsPerLiquidityCumulativeX128) = observations.observeSingle(time,0,slot0.tick,slot0.observationIndex,liquidity,slot0.observationCardinality) (UniswapV3Pool.sol#396-404)
        - (slot0.observationIndex,slot0.observationCardinality) = observations.write(_slot0.observationIndex,_blockTimestamp(),_slot0.tick,liquidityBefore,_slot0.observationCardinality,_slot0.observationCardinalityN
ext) (UniswapV3Pool.sol#341-348)
        State variables written after the call(s):
        - liquidity = LiquidityMath.addDelta(liquidityBefore,params.liquidityDelta) (UniswapV3Pool.sol#361)
        - (slot0.observationIndex,slot0.observationCardinality) = observations.write(_slot0.observationIndex,_blockTimestamp(),_slot0.tick,liquidityBefore,_slot0.observationCardinality,_slot0.observationCardinalityN
ext) (UniswapV3Pool.sol#341-348)
Reentrancy in UniswapV3Pool.collectProtocol(address,uint128,uint128) (UniswapV3Pool.sol#848-868):
        External calls:
        - TransferHelper.safeTransfer(token0,recipient,amount0) (UniswapV3Pool.sol#859)
        State variables written after the call(s):
        - protocolFees.token1 -= amount1 (UniswapV3Pool.sol#863)
Reentrancy in UniswapV3Pool.increaseObservationCardinalityNext(uint16) (UniswapV3Pool.sol#255-267):
        External calls:
        - observationCardinalityNextNew = observations.grow(observationCardinalityNextOld,observationCardinalityNext) (UniswapV3Pool.sol#262-263)
        State variables written after the call(s):
        - slot0.observationCardinalityNext = observationCardinalityNextNew (UniswapV3Pool.sol#264)
Reentrancy in UniswapV3Pool.initialize(uint160) (UniswapV3Pool.sol#271-289):
        External calls:
        - (cardinality,cardinalityNext) = observations.initialize(_blockTimestamp()) (UniswapV3Pool.sol#276)
        State variables written after the call(s):
        - slot0 = Slot0(sqrtPriceX96,tick,0,cardinality,cardinalityNext,0,true) (UniswapV3Pool.sol#278-286)
Reentrancy in UniswapV3Pool.swap(address,bool,int256,uint160,bytes) (UniswapV3Pool.sol#596-788):
        External calls:
        - (cache.tickCumulative,cache.secondsPerLiquidityCumulativeX128) = observations.observeSingle(cache.blockTimestamp,0,slot0Start.tick,slot0Start.observationIndex,cache.liquidityStart,slot0Start.observationCa
rdinality) (UniswapV3Pool.sol#699-706)
        - (observationIndex,observationCardinality) = observations.write(slot0Start.observationIndex,cache.blockTimestamp,slot0Start.tick,cache.liquidityStart,slot0Start.observationCardinality,slot0Start.observatio
nCardinalityNext) (UniswapV3Pool.sol#734-742)
        State variables written after the call(s):
        - feeGrowthGlobal0X128 = state.feeGrowthGlobalX128 (UniswapV3Pool.sol#760)
        - feeGrowthGlobal1X128 = state.feeGrowthGlobalX128 (UniswapV3Pool.sol#763)
        - liquidity = state.liquidity (UniswapV3Pool.sol#755)
        - (slot0.sqrtPriceX96,slot0.tick,slot0.observationIndex,slot0.observationCardinality) = (state.sqrtPriceX96,state.tick,observationIndex,observationCardinality) (UniswapV3Pool.sol#743-748)
```

```
nCardinalityNext) (UniswapV3Pool.sol#734-742)
        State variables written after the call(s):
        - protocolFees.token0 += state.protocolFee (UniswapV3Pool.sol#761)
        - protocolFees.token1 += state.protocolFee (UniswapV3Pool.sol#764)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2

Reentrancy in UniswapV3Pool.burn(int24,int24,uint128) (UniswapV3Pool.sol#517-543):
        External calls:
        - (position,amount0Int,amount1Int) = _modifyPosition(ModifyPositionParams(msg.sender,tickLower,tickUpper,- int256(amount).toInt128())) (UniswapV3Pool.sol#522-530)
                - (slot0.observationIndex,slot0.observationCardinality) = observations.write(_slot0.observationIndex,_blockTimestamp(),_slot0.tick,liquidityBefore,_slot0.observationCardinality,_slot0.observationCar
dinalityNext) (UniswapV3Pool.sol#341-348)
                - (tickCumulative,secondsPerLiquidityCumulativeX128) = observations.observeSingle(time,0,slot0.tick,slot0.observationIndex,liquidity,slot0.observationCardinality) (UniswapV3Pool.sol#396-404)
        Event emitted after the call(s):
        - Burn(msg.sender,tickLower,tickUpper,amount,amount0,amount1) (UniswapV3Pool.sol#542)
Reentrancy in UniswapV3Pool.collect(address,int24,int24,uint128,uint128) (UniswapV3Pool.sol#490-513):
        External calls:
        - TransferHelper.safeTransfer(token0,recipient,amount0) (UniswapV3Pool.sol#505)
        - TransferHelper.safeTransfer(token1,recipient,amount1) (UniswapV3Pool.sol#509)
        Event emitted after the call(s):
        - Collect(msg.sender,recipient,tickLower,tickUpper,amount0,amount1) (UniswapV3Pool.sol#512)
Reentrancy in UniswapV3Pool.collectProtocol(address,uint128,uint128) (UniswapV3Pool.sol#848-868):
        External calls:
        - TransferHelper.safeTransfer(token0,recipient,amount0) (UniswapV3Pool.sol#859)
        - TransferHelper.safeTransfer(token1,recipient,amount1) (UniswapV3Pool.sol#864)
        Event emitted after the call(s):
        - CollectProtocol(msg.sender,recipient,amount0,amount1) (UniswapV3Pool.sol#867)
Reentrancy in UniswapV3Pool.flash(address,uint256,uint256,bytes) (UniswapV3Pool.sol#791-834):
        External calls:
        - balance0Before = balance0() (UniswapV3Pool.sol#802)
                - (success,data) = token0.staticcall(abi.encodeWithSelector(IERC20Minimal.balanceOf.selector,address(this))) (UniswapV3Pool.sol#141-142)
        - balance1Before = balance1() (UniswapV3Pool.sol#803)
                - (success,data) = token1.staticcall(abi.encodeWithSelector(IERC20Minimal.balanceOf.selector,address(this))) (UniswapV3Pool.sol#151-152)
        - TransferHelper.safeTransfer(token0,recipient,amount0) (UniswapV3Pool.sol#805)
        - TransferHelper.safeTransfer(token1,recipient,amount1) (UniswapV3Pool.sol#806)
        - IUniswapV3FlashCallback(msg.sender).uniswapV3FlashCallback(fee0,fee1,data) (UniswapV3Pool.sol#808)
        - balance0After = balance0() (UniswapV3Pool.sol#810)
                - (success,data) = token0.staticcall(abi.encodeWithSelector(IERC20Minimal.balanceOf.selector,address(this))) (UniswapV3Pool.sol#141-142)
        - balance1After = balance1() (UniswapV3Pool.sol#811)
                - (success,data) = token1.staticcall(abi.encodeWithSelector(IERC20Minimal.balanceOf.selector,address(this))) (UniswapV3Pool.sol#151-152)
        Event emitted after the call(s):
        - Flash(msg.sender,recipient,amount0,amount1,paid0,paid1) (UniswapV3Pool.sol#833)
Reentrancy in UniswapV3Pool.increaseObservationCardinalityNext(uint16) (UniswapV3Pool.sol#255-267):
        External calls:
        - observationCardinalityNextNew = observations.grow(observationCardinalityNextOld,observationCardinalityNext) (UniswapV3Pool.sol#262-263)
        Event emitted after the call(s):
        - IncreaseObservationCardinalityNext(observationCardinalityNextOld,observationCardinalityNextNew) (UniswapV3Pool.sol#266)
Reentrancy in UniswapV3Pool.initialize(uint160) (UniswapV3Pool.sol#271-289):
        External calls:
        - (cardinality,cardinalityNext) = observations.initialize(_blockTimestamp()) (UniswapV3Pool.sol#276)
        Event emitted after the call(s):
        - Initialize(sqrtPriceX96,tick) (UniswapV3Pool.sol#288)
Reentrancy in UniswapV3Pool.mint(address,int24,int24,uint128,bytes) (UniswapV3Pool.sol#457-487):
        External calls:
        - (amount0Int,amount1Int) = _modifyPosition(ModifyPositionParams(recipient,tickLower,tickUpper,int256(amount).toInt128())) (UniswapV3Pool.sol#465-473)
                - (slot0.observationIndex,slot0.observationCardinality) = observations.write(_slot0.observationIndex,_blockTimestamp(),_slot0.tick,liquidityBefore,_slot0.observationCardinality,_slot0.observationCar
dinalityNext) (UniswapV3Pool.sol#341-348)
                - (tickCumulative,secondsPerLiquidityCumulativeX128) = observations.observeSingle(time,0,slot0.tick,slot0.observationIndex,liquidity,slot0.observationCardinality) (UniswapV3Pool.sol#396-404)
        - balance0Before = balance0() (UniswapV3Pool.sol#480)
                - (success,data) = token0.staticcall(abi.encodeWithSelector(IERC20Minimal.balanceOf.selector,address(this))) (UniswapV3Pool.sol#141-142)
```

```
Reentrancy in UniswapV3Pool.swap(address,bool,int256,uint160,bytes) (UniswapV3Pool.sol#596-788):
        External calls:
        - (cache.tickCumulative,cache.secondsPerLiquidityCumulativeX128) = observations.observeSingle(cache.blockTimestamp,0,slot0Start.tick,slot0Start.observationIndex,cache.liquidityStart,slot0Start.observationCa
rdinality) (UniswapV3Pool.sol#699-706)
        - (observationIndex,observationCardinality) = observations.write(slot0Start.observationIndex,cache.blockTimestamp,slot0Start.tick,cache.liquidityStart,slot0Start.observationCardinality,slot0Start.observatio
nCardinalityNext) (UniswapV3Pool.sol#734-742)
        - TransferHelper.safeTransfer(token1,recipient,uint256(-amount1)) (UniswapV3Pool.sol#773)
        - balance0Before = balance0() (UniswapV3Pool.sol#775)
                - (success,data) = token0.staticcall(abi.encodeWithSelector(IERC20Minimal.balanceOf.selector,address(this))) (UniswapV3Pool.sol#141-142)
        - IUniswapV3SwapCallback(msg.sender).uniswapV3SwapCallback(amount0,amount1,data) (UniswapV3Pool.sol#776)
        - require(bool,string)(balance0Before.add(uint256(amount0)) <= balance0(),IIA) (UniswapV3Pool.sol#777)
                - (success,data) = token0.staticcall(abi.encodeWithSelector(IERC20Minimal.balanceOf.selector,address(this))) (UniswapV3Pool.sol#141-142)
        - TransferHelper.safeTransfer(token0,recipient,uint256(-amount0)) (UniswapV3Pool.sol#779)
        - balance1Before = balance1() (UniswapV3Pool.sol#781)
                - (success,data) = token1.staticcall(abi.encodeWithSelector(IERC20Minimal.balanceOf.selector,address(this))) (UniswapV3Pool.sol#151-152)
        - IUniswapV3SwapCallback(msg.sender).uniswapV3SwapCallback(amount0,amount1,data) (UniswapV3Pool.sol#782)
        - require(bool,string)(balance1Before.add(uint256(amount1)) <= balance1(),IIA) (UniswapV3Pool.sol#783)
                - (success,data) = token1.staticcall(abi.encodeWithSelector(IERC20Minimal.balanceOf.selector,address(this))) (UniswapV3Pool.sol#151-152)
        Event emitted after the call(s):
        - Swap(msg.sender,recipient,amount0,amount1,state.sqrtPriceX96,state.liquidity,state.tick) (UniswapV3Pool.sol#786)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

UniswapV3Pool.initialize(uint160) (UniswapV3Pool.sol#271-289) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(slot0.sqrtPriceX96 == 0,AI) (UniswapV3Pool.sol#272)
UniswapV3Pool.swap(address,bool,int256,uint160,bytes) (UniswapV3Pool.sol#596-788) uses timestamp for comparisons
        Dangerous comparisons:
        - state.amountSpecifiedRemaining != 0 && state.sqrtPriceX96 != sqrtPriceLimitX96 (UniswapV3Pool.sol#641)
        - step.tickNext < TickMath.MIN_TICK (UniswapV3Pool.sol#653)
        - step.tickNext > TickMath.MAX_TICK (UniswapV3Pool.sol#655)
        - cache.feeProtocol > 0 (UniswapV3Pool.sol#682)
        - state.liquidity > 0 (UniswapV3Pool.sol#689)
        - state.sqrtPriceX96 == step.sqrtPriceNextX96 (UniswapV3Pool.sol#693)
        - state.sqrtPriceX96 != step.sqrtPriceStartX96 (UniswapV3Pool.sol#726)
        - state.tick != slot0Start.tick (UniswapV3Pool.sol#733)
        - cache.liquidityStart != state.liquidity (UniswapV3Pool.sol#755)
        - state.protocolFee > 0 (UniswapV3Pool.sol#761)
        - state.protocolFee > 0 (UniswapV3Pool.sol#764)
        - amount1 < 0 (UniswapV3Pool.sol#773)
        - require(bool,string)(balance0Before.add(uint256(amount0)) <= balance0(),IIA) (UniswapV3Pool.sol#777)
        - amount0 < 0 (UniswapV3Pool.sol#779)
        - require(bool,string)(balance1Before.add(uint256(amount1)) <= balance1(),IIA) (UniswapV3Pool.sol#783)
        - (step.sqrtPriceNextX96 < sqrtPriceLimitX96) (UniswapV3Pool.sol#663-671)
        - (step.sqrtPriceNextX96 > sqrtPriceLimitX96) (UniswapV3Pool.sol#663-671)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

FullMath.mulDiv(uint256,uint256,uint256) (libraries/FullMath.sol#14-106) uses assembly
        - INLINE ASM (libraries/FullMath.sol#26-30)
        - INLINE ASM (libraries/FullMath.sol#35-37)
        - INLINE ASM (libraries/FullMath.sol#52-54)
        - INLINE ASM (libraries/FullMath.sol#56-59)
        - INLINE ASM (libraries/FullMath.sol#66-68)
        - INLINE ASM (libraries/FullMath.sol#71-73)
        - INLINE ASM (libraries/FullMath.sol#77-79)
TickMath.getTickAtSqrtRatio(uint160) (libraries/TickMath.sol#61-204) uses assembly
        - INLINE ASM (libraries/TickMath.sol#69-73)
        - INLINE ASM (libraries/TickMath.sol#74-78)
        - INLINE ASM (libraries/TickMath.sol#79-83)
        - INLINE ASM (libraries/TickMath.sol#84-88)
        - INLINE ASM (libraries/TickMath.sol#89-93)
        - INLINE ASM (libraries/TickMath.sol#94-98)
        - INLINE ASM (libraries/TickMath.sol#99-103)
        - INLINE ASM (libraries/TickMath.sol#104-107)
        - INLINE ASM (libraries/TickMath.sol#126-131)
        - INLINE ASM (libraries/TickMath.sol#132-137)
        - INLINE ASM (libraries/TickMath.sol#138-143)
        - INLINE ASM (libraries/TickMath.sol#144-149)
        - INLINE ASM (libraries/TickMath.sol#150-155)
        - INLINE ASM (libraries/TickMath.sol#156-161)
        - INLINE ASM (libraries/TickMath.sol#162-167)
        - INLINE ASM (libraries/TickMath.sol#168-173)
        - INLINE ASM (libraries/TickMath.sol#174-179)
        - INLINE ASM (libraries/TickMath.sol#180-185)
        - INLINE ASM (libraries/TickMath.sol#186-191)
        - INLINE ASM (libraries/TickMath.sol#192-196)
UnsafeMath.divRoundingUp(uint256,uint256) (libraries/UnsafeMath.sol#12-16) uses assembly
        - INLINE ASM (libraries/UnsafeMath.sol#13-15)
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', '>=0.4.0<0.8.0', '>=0.5.0', '>=0.5.0<0.8.0', '>=0.6.0', '>=0.7.0']
        - =0.7.6 (NoDelegateCall.sol#2)
        - =0.7.6 (UniswapV3Pool.sol#2)
        - >=0.5.0 (interfaces/IERC20Minimal.sol#2)
        - >=0.5.0 (interfaces/IUniswapV3Factory.sol#2)
        - >=0.5.0 (interfaces/IUniswapV3Pool.sol#2)
        - >=0.5.0 (interfaces/IUniswapV3PoolDeployer.sol#2)
        - >=0.5.0 (interfaces/callback/IUniswapV3FlashCallback.sol#2)
        - >=0.5.0 (interfaces/callback/IUniswapV3MintCallback.sol#2)
        - >=0.5.0 (interfaces/callback/IUniswapV3SwapCallback.sol#2)
        - >=0.5.0 (interfaces/pool/IUniswapV3PoolActions.sol#2)
        - >=0.5.0 (interfaces/pool/IUniswapV3PoolDerivedState.sol#2)
        - >=0.5.0 (interfaces/pool/IUniswapV3PoolEvents.sol#2)
        - >=0.5.0 (interfaces/pool/IUniswapV3PoolImmutables.sol#2)
        - >=0.5.0 (interfaces/pool/IUniswapV3PoolOwnerActions.sol#2)
        - >=0.5.0 (interfaces/pool/IUniswapV3PoolState.sol#2)
        - >=0.5.0 (libraries/BitMath.sol#2)
        - >=0.4.0 (libraries/FixedPoint128.sol#2)
        - >=0.4.0 (libraries/FixedPoint96.sol#2)
        - >=0.4.0<0.8.0 (libraries/FullMath.sol#2)
        - >=0.5.0 (libraries/LiquidityMath.sol#2)
        - >=0.7.0 (libraries/LowGasSafeMath.sol#2)
        - >=0.5.0<0.8.0 (libraries/Oracle.sol#2)
        - >=0.5.0<0.8.0 (libraries/Position.sol#2)
        - >=0.5.0 (libraries/SafeCast.sol#2)
        - >=0.5.0 (libraries/SqrtPriceMath.sol#2)
        - >=0.5.0 (libraries/SwapMath.sol#2)
        - >=0.5.0<0.8.0 (libraries/Tick.sol#2)
        - >=0.5.0 (libraries/TickBitmap.sol#2)
        - >=0.5.0<0.8.0 (libraries/TickMath.sol#2)
        - >=0.6.0 (libraries/TransferHelper.sol#2)
        - >=0.5.0 (libraries/UnsafeMath.sol#2)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

LowGasSafeMath.mul(uint256,uint256) (libraries/LowGasSafeMath.sol#27-29) is never used and should be removed
LowGasSafeMath.sub(uint256,uint256) (libraries/LowGasSafeMath.sol#19-21) is never used and should be removed
Oracle.binarySearch(Oracle.Observation[65535],uint32,uint32,uint16,uint16) (libraries/Oracle.sol#153-184) is never used and should be removed
Oracle.getSurroundingObservations(Oracle.Observation[65535],uint32,uint32,int24,uint16,uint128,uint16) (libraries/Oracle.sol#198-230) is never used and should be removed
Oracle.grow(Oracle.Observation[65535],uint16,uint16) (libraries/Oracle.sol#108-120) is never used and should be removed
Oracle.initialize(Oracle.Observation[65535],uint32) (libraries/Oracle.sol#52-63) is never used and should be removed
Oracle.lte(uint32,uint32,uint32) (libraries/Oracle.sol#128-140) is never used and should be removed
Oracle.observe(Oracle.Observation[65535],uint32,uint32[],int24,uint16,uint128,uint16) (libraries/Oracle.sol#300-324) is never used and should be removed
Oracle.observeSingle(Oracle.Observation[65535],uint32,uint32,int24,uint16,uint128,uint16) (libraries/Oracle.sol#245-287) is never used and should be removed
Oracle.transform(Oracle.Observation,uint32,int24,uint128) (libraries/Oracle.sol#30-45) is never used and should be removed
Oracle.write(Oracle.Observation[65535],uint16,uint32,int24,uint128,uint16,uint16) (libraries/Oracle.sol#70-101) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
```

```
Variable SqrtPriceMath.getAmount1Delta(uint160,uint160,uint128,bool).sqrtRatioAX96 (libraries/SqrtPriceMath.sol#183) is too similar to SqrtPriceMath.getAmount1Delta(uint160,uint160,uint128,bool).sqrtRatioBX96 (lib
aries/SqrtPriceMath.sol#184)
Variable SqrtPriceMath.getAmount1Delta(uint160,uint160,int128).sqrtRatioAX96 (libraries/SqrtPriceMath.sol#218) is too similar to SqrtPriceMath.getAmount0Delta(uint160,uint160,int128).sqrtRatioBX96 (libraries/SqrtP
iceMath.sol#203)
Variable SqrtPriceMath.getAmount1Delta(uint160,uint160,uint128,bool).sqrtRatioAX96 (libraries/SqrtPriceMath.sol#183) is too similar to SqrtPriceMath.getAmount0Delta(uint160,uint160,int128).sqrtRatioBX96 (libraries,
SqrtPriceMath.sol#203)
Variable SqrtPriceMath.getAmount0Delta(uint160,uint160,int128).sqrtRatioAX96 (libraries/SqrtPriceMath.sol#202) is too similar to SqrtPriceMath.getAmount1Delta(uint160,uint160,uint128,bool).sqrtRatioBX96 (libraries,
SqrtPriceMath.sol#184)
Variable SqrtPriceMath.getAmount0Delta(uint160,uint160,uint128,bool).sqrtRatioAX96 (libraries/SqrtPriceMath.sol#154) is too similar to SqrtPriceMath.getAmount1Delta(uint160,uint160,int128).sqrtRatioBX96 (libraries,
SqrtPriceMath.sol#219)
Variable SqrtPriceMath.getAmount1Delta(uint160,uint160,int128).sqrtRatioAX96 (libraries/SqrtPriceMath.sol#218) is too similar to SqrtPriceMath.getAmount1Delta(uint160,uint160,uint128,bool).sqrtRatioBX96 (libraries,
SqrtPriceMath.sol#184)
Variable SqrtPriceMath.getAmount1Delta(uint160,uint160,uint128,bool).sqrtRatioAX96 (libraries/SqrtPriceMath.sol#183) is too similar to SqrtPriceMath.getAmount1Delta(uint160,uint160,int128).sqrtRatioBX96 (libraries,
SqrtPriceMath.sol#219)
Variable SqrtPriceMath.getAmount0Delta(uint160,uint160,int128).sqrtRatioAX96 (libraries/SqrtPriceMath.sol#202) is too similar to SqrtPriceMath.getAmount1Delta(uint160,uint160,int128).sqrtRatioBX96 (libraries/SqrtP
iceMath.sol#219)
Variable SqrtPriceMath.getAmount0Delta(uint160,uint160,uint128,bool).sqrtRatioAX96 (libraries/SqrtPriceMath.sol#154) is too similar to SqrtPriceMath.getAmount0Delta(uint160,uint160,int128).sqrtRatioBX96 (libraries,
SqrtPriceMath.sol#203)
Variable SqrtPriceMath.getAmount1Delta(uint160,uint160,int128).sqrtRatioAX96 (libraries/SqrtPriceMath.sol#218) is too similar to SqrtPriceMath.getAmount1Delta(uint160,uint160,int128).sqrtRatioBX96 (libraries/SqrtP
iceMath.sol#219)
Variable SqrtPriceMath.getAmount0Delta(uint160,uint160,uint128,bool).sqrtRatioAX96 (libraries/SqrtPriceMath.sol#154) is too similar to SqrtPriceMath.getAmount1Delta(uint160,uint160,uint128,bool).sqrtRatioBX96 (lib
aries/SqrtPriceMath.sol#184)
Variable Tick.getFeeGrowthInside(mapping(int24 => Tick.Info),int24,int24,int24,uint256,uint256).feeGrowthAbove0X128 (libraries/Tick.sol#83) is too similar to Tick.getFeeGrowthInside(mapping(int24 => Tick.Info),int2
4,int24,int24,uint256,uint256).feeGrowthAbove1X128 (libraries/Tick.sol#84)
Variable Tick.getFeeGrowthInside(mapping(int24 => Tick.Info),int24,int24,int24,uint256,uint256).feeGrowthBelow0X128 (libraries/Tick.sol#72) is too similar to Tick.getFeeGrowthInside(mapping(int24 => Tick.Info),int2
4,int24,int24,uint256,uint256).feeGrowthBelow1X128 (libraries/Tick.sol#73)
Variable Tick.getFeeGrowthInside(mapping(int24 => Tick.Info),int24,int24,int24,uint256,uint256).feeGrowthGlobal0X128 (libraries/Tick.sol#65) is too similar to Tick.getFeeGrowthInside(mapping(int24 => Tick.Info),in
24,int24,int24,uint256,uint256).feeGrowthGlobal1X128 (libraries/Tick.sol#66)
Variable Tick.cross(mapping(int24 => Tick.Info),int24,uint256,uint256,uint160,int56,uint32).feeGrowthGlobal0X128 (libraries/Tick.sol#171) is too similar to Tick.cross(mapping(int24 => Tick.Info),int24,uint256,uint2
56,uint160,int56,uint32).feeGrowthGlobal1X128 (libraries/Tick.sol#172)
Variable Tick.cross(mapping(int24 => Tick.Info),int24,uint256,uint256,uint160,int56,uint32).feeGrowthGlobal0X128 (libraries/Tick.sol#171) is too similar to Tick.getFeeGrowthInside(mapping(int24 => Tick.Info),int24
int24,int24,uint256,uint256).feeGrowthGlobal1X128 (libraries/Tick.sol#66)
Variable Tick.update(mapping(int24 => Tick.Info),int24,int24,int128,uint256,uint256,uint160,int56,uint32,bool,uint128).feeGrowthGlobal0X128 (libraries/Tick.sol#115) is too similar to Tick.cross(mapping(int24 => Ti
k.Info),int24,uint256,uint256,uint160,int56,uint32).feeGrowthGlobal1X128 (libraries/Tick.sol#172)
Variable Tick.getFeeGrowthInside(mapping(int24 => Tick.Info),int24,int24,int24,uint256,uint256).feeGrowthGlobal0X128 (libraries/Tick.sol#65) is too similar to Tick.update(mapping(int24 => Tick.Info),int24,int24,in
128,uint256,uint256,uint160,int56,uint32,bool,uint128).feeGrowthGlobal1X128 (libraries/Tick.sol#116)
Variable Tick.cross(mapping(int24 => Tick.Info),int24,uint256,uint256,uint160,int56,uint32).feeGrowthGlobal0X128 (libraries/Tick.sol#171) is too similar to Tick.update(mapping(int24 => Tick.Info),int24,int24,int12
,uint256,uint256,uint160,int56,uint32,bool,uint128).feeGrowthGlobal1X128 (libraries/Tick.sol#116)
Variable Tick.update(mapping(int24 => Tick.Info),int24,int24,int128,uint256,uint256,uint160,int56,uint32,bool,uint128).feeGrowthGlobal0X128 (libraries/Tick.sol#115) is too similar to Tick.getFeeGrowthInside(mappin
(int24 => Tick.Info),int24,int24,int24,uint256,uint256).feeGrowthGlobal1X128 (libraries/Tick.sol#66)
Variable Tick.getFeeGrowthInside(mapping(int24 => Tick.Info),int24,int24,int24,uint256,uint256).feeGrowthGlobal0X128 (libraries/Tick.sol#65) is too similar to Tick.cross(mapping(int24 => Tick.Info),int24,uint256,u
nt256,uint160,int56,uint32).feeGrowthGlobal1X128 (libraries/Tick.sol#172)
Variable Tick.update(mapping(int24 => Tick.Info),int24,int24,int128,uint256,uint256,uint160,int56,uint32,bool,uint128).feeGrowthGlobal0X128 (libraries/Tick.sol#115) is too similar to Tick.update(mapping(int24 => T
ck.Info),int24,int24,int128,uint256,uint256,uint160,int56,uint32,bool,uint128).feeGrowthGlobal1X128 (libraries/Tick.sol#116)
Variable Tick.getFeeGrowthInside(mapping(int24 => Tick.Info),int24,int24,int24,uint256,uint256).feeGrowthInside0X128 (libraries/Tick.sol#67) is too similar to Tick.getFeeGrowthInside(mapping(int24 => Tick.Info),in
24,int24,int24,uint256,uint256).feeGrowthInside1X128 (libraries/Tick.sol#67)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-are-too-similar

BitMath.mostSignificantBit(uint256) (libraries/BitMath.sol#13-45) uses literals with too many digits:
        - x >= 0x100000000000000000000000000000000 (libraries/BitMath.sol#16)
BitMath.mostSignificantBit(uint256) (libraries/BitMath.sol#13-45) uses literals with too many digits:
        - x >= 0x10000000000000000 (libraries/BitMath.sol#20)
BitMath.mostSignificantBit(uint256) (libraries/BitMath.sol#13-45) uses literals with too many digits:
        - x >= 0x100000000 (libraries/BitMath.sol#24)
FixedPoint128.slitherConstructorConstantVariables() (libraries/FixedPoint128.sol#6-8) uses literals with too many digits:
        - Q128 = 0x100000000000000000000000000000000 (libraries/FixedPoint128.sol#7)
FixedPoint96.slitherConstructorConstantVariables() (libraries/FixedPoint96.sol#7-10) uses literals with too many digits:
        - Q96 = 0x1000000000000000000000000 (libraries/FixedPoint96.sol#9)
TickMath.getSqrtRatioAtTick(int24) (libraries/TickMath.sol#23-54) uses literals with too many digits:
        - ratio = 0x100000000000000000000000000000000 (libraries/TickMath.sol#27)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits
. analyzed (97 contracts with 75 detectors), 522 result(s) found
ethsecm59c4e323a4a:/code/v3-core/contracts$
```

Results

Some false positive errors were reported by the tool. All the other issues have been categorized above according to their level of severity.

Closing Summary

Overall, smart contracts are well written and it's a fork of Uniswap-V3.

Some information issues were raised and the NSK Swap Team acknowledged it.

Disclaimer

Quillhash audit is not a security warranty, investment advice, or an endorsement of the NSK SWAP. This audit does not provide a security or correctness guarantee of the audited smart contracts. The statements made in this document should not be interpreted as investment or legal advice, nor should its authors be held accountable for decisions made based on them. Securing smart contracts is a multistep process. One audit cannot be considered enough. We recommend that the NSK SWAP Team put in place a bug bounty program to encourage further analysis of the smart contract by other third parties.

About QuillAudits

QuillAudits is a secure smart contracts audit platform designed by QuillHash Technologies. We are a team of dedicated blockchain security experts and smart contract auditors determined to ensure that Smart Contract-based Web3 projects can avail the latest and best security solutions to operate in a trustworthy and risk-free ecosystem.

500+
Audits Completed

$15B
Secured

500K
Lines of Code Audited

Follow Our Journey

Audit Report
July, 2022

For

NSKSWAP

QuillAudits

Canada, India, Singapore, United Kingdom

audits.quillhash.com

audits@quillhash.com