CYBER UNIT
Cyber Security Strategic Partner
www.cyberunit.tech

UNN Smart Contract Audit

Date: March 24, 2021
Report for: UNN Finance
By: CyberUnit.Tech

This document may contain confidential information about IT systems and the customer's intellectual property and information about potential vulnerabilities and exploitation methods.
The report contains confidential information. This information can be used internally by the customer. The customer can release the information after fixing all vulnerabilities.

Document

Name	UNN OC Protections
Date	24/03/21
Commit	ced026a06b4126167c3f95861059f367d8e8e237

Table of contents

Introduction

This report presents the Customer` s smart contract's security assessment findings and its code review conducted between March 9 – March 24, 2021.

Scope

The scope of the project is UNN OC Protections smart contracts.

We have scanned this smart contract for commonly known and more specific vulnerabilities. Here are some of the widely known vulnerabilities that are considered (the complete list includes them but does not limit by them):

- Reentrancy
- Timestamp Dependence
- Gas Limit and Loops
- DoS with (Unexpected) Throw
- DoS with Block Gas Limit
- Transaction-Ordering Dependence
- Style guide violation
- Transfer forwards all gas
- ERC20 API violation
- Compiler version not fixed
- Unchecked external call – Unchecked math
- Unsafe type inference
- Implicit visibility level

Executive Summary

Our team performed an analysis of code functionality, manual audit, and automated checks with Slither and remix IDE (see Appendix B pic 1-7). All issues found during automated analysis reviewed have been manually, and application vulnerabilities are presented in the Audit overview section. A general overview is shown in the AS-IS section, and you can find all found issues in the Audit overview section.

Findings relate mostly to the OC protections smart contract, while minor code style improvements are also recommended.

The auditors did not report critical findings which can lead to significant damage.

Severity Definitions

Risk Level	Description
Critical	Critical vulnerabilities are usually straightforward to exploit and can lead to tokens loss etc.
High	High-level vulnerabilities are difficult to exploit; however, they also significantly impact smart contract execution, e.g., public access to crucial functions.
Medium	Medium-level vulnerabilities are essential to fix; however, they can't lead to tokens loss.
Low	Low-level vulnerabilities are mostly related to outdated, unused, etc., code snippets that can't significantly impact execution.
Lowest / Code Style / Best Practice	Lowest-level vulnerabilities, code style violations, and info statements can't affect smart contract execution and can be ignored.

Code Style Comments

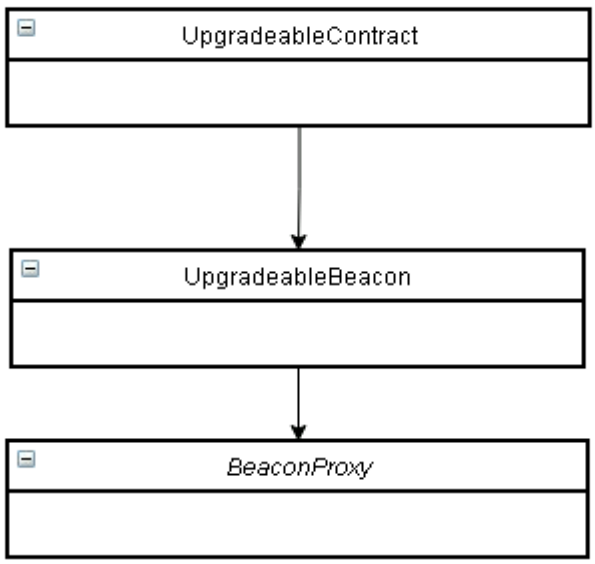
- Code style should be fixed according to https://docs.soliditylang.org/en/v0.8.2/style-guide.html
- In the code there are a lot of similar code style errors, such as missing space after ',' and 'if'.
- Missing spaces before '{'.

- It is also recommended to add linters library, 'solhint' or 'solium'.

AS-IS overview

Protection contract consists of the next smart contracts:

1. StructuredLinkedList.sol, Address.sol, Ownable.sol contracts – supporting libraries
2. Initializable.sol,PausableUpgradeable.so, SafeMathUpgradeable.sol,IERC20Upgradeable.sol, SafeERC20Upgradeable.sol,AccessControlUpgradeable.sol, ERC721PausableUpgradeable.sol contracts – openzeppelin
3. IERC20Token.sol, IERC1643.sol, IOCProtectionSeller.sol, IParamStorage.sol, IPool.sol, IProtection.sol, IUnionRouter.sol,IUUNNRegistry.sol contracts – interfaces
4. OCProtections contracts – OCProtections.sol, ParamStorage.sol, ProtectionSeller.sol, ProtectionUpgradable.sol, UnionERC20Pool.sol, UnionRouter.sol, uUNNToken.sol, BeaconProxy.sol, IBeacon.sol, Proxy.sol, ProxyAdmin.sol, TransparentUpgradeableProxy.sol, UpgradeableBeacon.sol, UpgradeableProxy.sol, UnionAssetPool.sol



AS-IS Upgradable overview

UpgradeableBeacon contract inherits the IBeacon interface and the class – Ownable

UpgradeableBeacon contract init function was called with the following parameters:

- address(implementation_)

Upgraded function was called with the following parameters:

- address(implementation_)

UpgradeableProxy contract inherits the class – Proxy.

UpgradeableProxy contract init function was called with the following parameters:

- address(_logic)
- bytes(_data)

Upgraded function was called with the following parameters:

- address(implementation)

Upgradable overview

Critical

No critical severity vulnerabilities were found.

High

No high severity vulnerabilities were found.

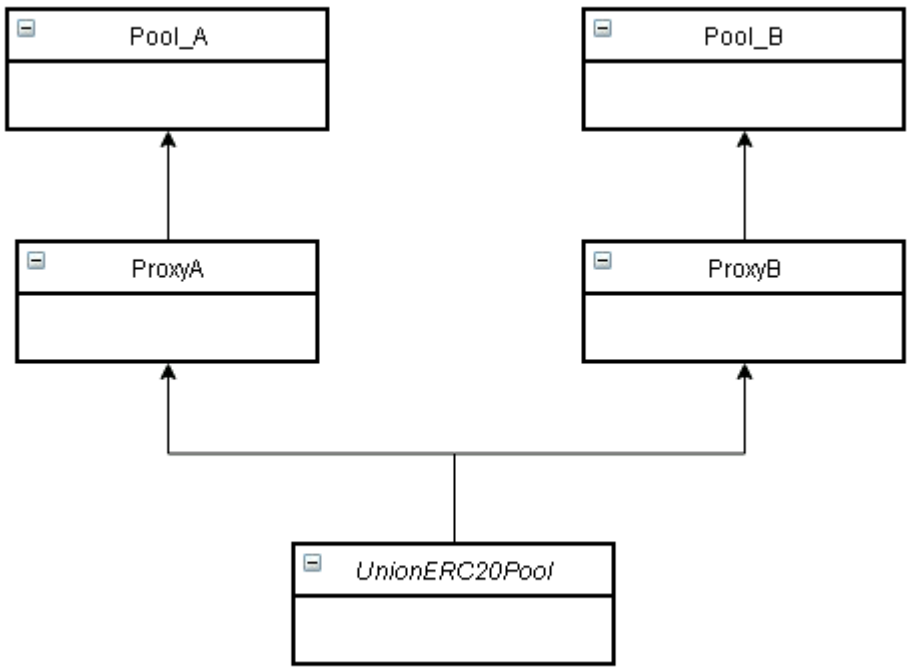
Medium

No medium severity vulnerabilities were found.

Low

1. Different versions of Solidity are used in Version used: ['>=0.6.0<0.8.0', '>=0.6.2<0.8.0']
2. Version=0.6.12 necessitates a version too recent to be trusted. Consider deploying with 0.6.11

Note: Standard OpenZeppelin smart contracts for updates known as UpgradeabilityProxy.The contract has several differences compared to the latest OpenZeppelin implementation.



AS-IS UnionERC20Pool overview

UnionERC20Pool contract inherits the IPool and the classes – contract inherits the classes – AccessControlUpgradeable, PausableUpgradeable, PoolUpgradable, SignLib

__UnionERC20Pool_init function was called with the following parameters:

- address admin
- address _basicToken
- string _description

__UnionERC20Pool_init_unchained function was called with the following parameters:

- address admin
- setPoolReserveAddress
- address _poolReserveAddress

setFoundationReserveAddress function was called with the following parameters:

- address _foundationReserveAddress

setPoolReservePremiumCommission function was called with the following parameters:

- uint8 _nom
- uint8 _denom

setFoundationReservePremiumCommission function was called with the following parameters:

- uint8 _nom
- uint8 _denom

setPoolReserveExcessLiquidityCommission function was called with the following parameters:

- uint8 _nom
- uint8 _denom

setFoundationReserveExcessLiquidityCommission function was called with the following parameters:

- uint8 _nom
- uint8 _denom

pause function was called with no parameters.

unpause function was called with no parameters.

withdrawPoolReserveCommission function was called with the following parameters:

- uint256 amount

withdrawFoundationReserveCommission function was called with the following parameters:

- uint256 amount

_distributeProfit function was called with the following parameters:

- uint256 totalPremiumMatured

withdraw function was called with the following parameters:

- uint256 _amount

flushMCRPendingQueue function was called with the following parameters:

- uint256 cycleAmount
- uint256[2] data, bytes signature

getBasicToken function was called with no parameters.

getBasicTokenDecimals function was called with no parameters.

getWriterData function was called with the following parameters:

- address _writer

getTotalValueLocked function was called with no parameters.

getPoolStat function was called with no parameters.

_updateMCR function was called with the following parameters:

- uint256 newMCR

- uint256 newMCRBlockNumber
- uint256 mcrIncrement

_unloadMCRPendingQueue function was called with the following parameters:

- uint cyclesAmount
- uint256 newMCRBlockNumber

_beforeDeposit function was called with the following parameters:

- uint256 amountTokenSent
- address sender
- address holder

_afterDeposit function was called with the following parameters:

- uint256 amountTokenSent
- uint256 amountLiquidityGot
- address sender
- address holder

_beforeWithdraw function was called with the following parameters:

- uint256 amountLiquidity
- address holder
- address receiver

_afterWithdraw function was called with the following parameters:

- uint256 amountTokenReceived
- address holder
- address receiver

UnionERC20Pool overview

Critical

No critical severity vulnerabilities were found.

High

No high severity vulnerabilities were found.

Medium

No medium severity vulnerabilities were found.

Low

1. Different versions of Solidity are used in Version used: ['>=0.4.24<0.8.0', '>=0.6.0', '>=0.6.0<0.8.0', '>=0.6.12', '>=0.6.2<0.8.0', '>=0.6.6'] (see Appendix A pic. 2 for evidence)

AS-IS PoolUpgradable overview

PoolUpgradable contract inherits the IERC20Token interface and the class – ERC20Upgradeable.

__Pool_init function was called with the following parameters:

- address(_basicToken)
- string(_description)

__Pool_init_unchained function was called with the following parameters:

- address(_basicToken)

depositTo function was called with the following parameters:

- uint256(_amount)
- address(_to)

deposit function was called with the following parameters:

- uint256(_amount)

withdraw function was called with the following parameters:

- uint256(_amount)

withdrawTo function was called with the following parameters:

- uint256(_amount)
- address(_to)

_deposit function was called with the following parameters:

- uint256(_amount)
- address(_to)

_withdraw function was called with the following parameters:

- uint256(amountLiquidity)
- address(to)

AS-IS ProtectionUpgradable overview

ProtectionUpgradable contract inherits the IProtection interface and the classes – Initializable, IERC1643.

version function was called with no parameters.

__Protection_init function was called with the following parameters:

- uint256(_id)
- address(_uunn)
- uint256(_amount)
- uint256(_validTo)
- uint256(_strike)
- address(_pools)

- uint16(_poolShares)

__Protection_init_unchained function was called with the following parameters:

- uint256(_id)
- address(_uunn)
- uint256(_amount)
- uint256(_validTo)
- uint256(_strike)
- address(_pools)
- uint16(_poolShares)

onlyTokenOwner function was called with no parameters.

_finalize function was called with no parameters.

setDocument function was called with the following parameters:

- bytes32(name)
- string(uri)
- bytes32(documentHash)

removeDocument function was called with the following parameters:

- bytes32(name)

getAllDocuments function was called with no parameters.

getPool function was called with the following parameters:

- uint256(_index)

getPoolsLength function was called with no parameters.

getPoolsTotalShare function was called with no parameters.

getBeneficiary function was called with the following parameters:

- uint256(_index)

getBeneficiariesLength function was called with no parameters.

exercise function was called with the following parameters:

- uint256(_amount)

AS-IS OCProtections overview

OCProtections contract inherits the ProtectionUpgradable Class.

version function was called with no parameters.

initialize function was called with the following parameters:

- address(_admin)
- address(_uunn)

pause function was called with no parameters.

unpause function was called with no parameters.

create function was called with the following parameters:

- address pool
- uint256 validTo
- uint256 amount
- uint256 strike
- uint256 deadline
- uint256 data
- bytes signature

createTo function was called with the following parameters:

- address pool
- uint256 validTo
- uint256 amount
- uint256 strike
- uint256 deadline
- uint256 data
- bytes signature
- address erc721Receiver

withdrawPremium function was called with the following parameters:

- uint256 _id
- uint256 _premium

exercise function was called with the following parameters:

- uint256 _id
- uint256 _amount

getProtectionData function was called with the following parameters:

- uint256 id

setDocument function was called with the following parameters:

- uint256 id
- bytes32 name
- string uri
- bytes32 documentHash

removeDocument function was called with the following parameters:

- uint256 id
- bytes32 name

getDocument function was called with the following parameters:

- uint256 id
- bytes32 _name

getAllDocuments function was called with the following parameters:

- uint256 id

OCProtections overview

Critical

No critical severity vulnerabilities were found.

High

1. Contract 'OCProtections', function 'removeDocument'. In the dev comments to the function it is said that the function can be only called by the contract owner but in reality it is available to call by anyone.
    a. Possible solution: restrict the access to the contract owner only.

Medium

1. Reentrancy vulnerabilities (see Appendix A pic. 4 for evidence).
2. Contract 'OCProtections', function 'setDocument'. The access to the function is not restricted. Probably that is expected behavior but looks like not. This way anybody can change an existing document to the other one.
    a. Possible solution: If document exists then resetting of it should be available for the document owner only, so it requires the following inspection:

        require(msg.sender == uunn.ownerOf(_id))

Low

1. Different versions of Solidity are used in Version used: ['>=0.4.24<0.8.0','>=0.6.0<0.8.0','>=0.6.12','>=0.6.2<0.8.0','>=0.6.6'] (see Appendix A pic. 5 for evidence).

2. Contract 'OCProtections', line 157. No need for a require statement. The condition under require will be checked on the next line with SafeMath library.

AS-IS ProtectionSeller overview

ProtectionSeller contract inherits the classes – Initializable AccessControlUpgradeable PausableUpgradeable

__ProtectionSeller_init function was called with the following parameters:

- address(_admin)
- address(_uunn)

__ProtectionSeller_init_unchained function was called with the following parameters:

- address(_admin)
- address(_uunn)

pause function was called with no parameters.

unpause function was called with no parameters.

_mint function was called with following parameters:

- uint256(_admin)
- address(protectionContract)
- address(to)

ProtectionSeller overview

Critical

No critical severity vulnerabilities were found.

High

1. Uninitialized-state (see Appendix A pic. 1 for evidence).

Medium

No medium severity vulnerabilities were found.

Low

2. Different versions of Solidity are used in Version used: ['>=0.4.24<0.8.0','>=0.6.0<0.8.0','>=0.6.12', '>=0.6.2<0.8.0'] (see Appendix A pic. 2 for evidence)
3. version=0.6.12 necessitates a version too recent to be trusted. Consider deploying with 0.6.11

AS-IS uUNNToken overview

uUNNToken contract inherits the ProtectionUpgradable Class.

__uUNNToken_init function was called with the following parameters:

- address(_admin)

__uUNNToken_init_unchained function was called with the following parameters:

- address(_admin)

initialize function was called with the following parameters:

- address(_admin)

setBaseURI function was called with the following parameters:

- string _baseURI

pause function was called with no parameters.

unpause function was called with no parameters.

protectionContract function was called with the following parameters:

- uint256(tokenId)

mint function was called with the following parameters:

- uint256(tokenId)
- address(protectionContract)
- address(to)

burn function was called with the following parameters:

- uint256(tokenId)

Audit uUNNToken overview

Critical

No critical severity vulnerabilities were found.

High

No high severity vulnerabilities were found.

Medium

No medium severity vulnerabilities were found.

Low

1. Different versions of Solidity are used in Version used: ['>=0.4.24<0.8.0','>=0.6.0<0.8.0','>=0.6.12', '>=0.6.2<0.8.0'] (see Appendix A pic. 2 for evidence)
2. version=0.6.12 necessitates a version too recent to be trusted. Consider deploying with 0.6.11.

AS-IS UnionRouter overview

UnionRouter contract inherits the IUnionRouter interface and the class – AccessControlUpgradeable.

initialize function was called with the following parameters:

- address(admin)

addCollateralProtection function was called with the following parameters:

- address(token)
- address(pool)
- address(sellerContract)

removeCollateralProtection function was called with the following parameters:

- address(token)

setUUNNToken function was called with the following parameters:

- address(_address)

collateralProtection function was called with the following parameters:

- address(token)

uunnToken function was called with no parameters.

Audit UnionRouter overview

Critical

No critical severity vulnerabilities were found.

High

No high severity vulnerabilities were found.

Medium

No medium severity vulnerabilities were found.

Low

No low severity vulnerabilities were found.

AS-IS ParamStorage overview

ParamStorage contract inherits the IParamStorage interface and the class – AccessControlUpgradeable.

initialize function was called with the following parameters:

- address(admin)

setParamAddress function was called with the following parameters:

- uint16(_key)
- address(_value)

setParamUInt256 function was called with the following parameters:

- uint16(_key)
- address(_value)

getAddress function was called with the following parameters:

- uint16(key)

getUInt256 function was called with the following parameters:

- uint16(key)

ParamStorage overview

Critical

No critical severity vulnerabilities were found.

High

No high severity vulnerabilities were found.

Medium

No medium severity vulnerabilities were found.

Low

No low severity vulnerabilities were found.

AS-IS UnionAssetPool overview

UnionAssetPool contract inherits the IAssetPoolinterface and the class – UnionERC20Pool.

version initialize function was called with the following parameters:

- address admin
- address _basicToken
- address _ocProtectionStorage
- address _priceFeed
- bool _priceFeedReverse
- string _description

onProtectionPremium function was called with the following parameters:

- address buyer
- uint256 data

unlockPremium function was called with the following parameters:

- uint256[] _ids

onPayoutCoverage function was called with the following parameters:

- uint256 _id
- uint256 _premiumToUnlock
- uint256 _coverageToPay
- address _beneficiary

withdrawWithData function was called with the following parameters:

- uint256 _requestID
- uint256 _amount
- uint256[6] _data
- bytes _signature

getLatestPrice function was called with no parameters.

getPriceDecimals function was called with no parameters.

UnionAssetPool overview

Critical

No critical severity vulnerabilities were found.

High

No high severity vulnerabilities were found.

Medium

No medium severity vulnerabilities were found.

Low

No low severity vulnerabilities were found.

Disclaimers

Disclaimer

The smart contracts given for audit had been analyzed following the best industry practices at the date of this report, concerning: cybersecurity vulnerabilities and issues in smart contract source code, the details of which are disclosed in this report, (Source Code); the Source Code compilation, deployment, and functionality (performing the intended functions).

The audit makes no statements or warranties on the security of the code. It can also not be considered a sufficient assessment regarding the code's utility and safety, bug-free status, or any other contract statements. While we have done our best to conduct the analysis and produce this report, it is important to note that you should not rely on this report only – we recommend proceeding with several independent audits and a public bug bounty program to ensure the security of smart contracts.

Technical Disclaimer

Smart contracts are deployed and executed on the blockchain platform. The platform, programming language, and other software related to the smart contract can have their vulnerabilities leading to hacks. Thus, the audit can't guarantee the explicit security of the audited smart contracts.

Appendix A. Evidences

Pic 1. Never initialized:

```
ProtectionUpgradable.beneficiaries (ProtectionUpgradable/ProtectionUpgradable.sol#24) is never initialized. It is used in:
        - ProtectionUpgradable.getBeneficiary(uint256) (ProtectionUpgradable/ProtectionUpgradable.sol#160-163)
        - ProtectionUpgradable.getBeneficiariesLength() (ProtectionUpgradable/ProtectionUpgradable.sol#165-167)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-state-variables
```

Pic 2. Different pragma directives are used:

```
Different versions of Solidity is used in :
        - Version used: ['>=0.4.24<0.8.0', '>=0.6.0', '>=0.6.0<0.8.0', '>=0.6.12', '>=0.6.2<0.8.0', '>=0.6.6']
        - >=0.6.12 (UnionERC20Pool/UnionERC20Pool.sol#1)
        - >=0.6.0<0.8.0 (UnionERC20Pool/libraries.sol#5)
        - >=0.6.2<0.8.0 (UnionERC20Pool/libraries.sol#305)
        - >=0.4.24<0.8.0 (UnionERC20Pool/libraries.sol#474)
        - >=0.6.0<0.8.0 (UnionERC20Pool/libraries.sol#539)
        - >=0.6.0<0.8.0 (UnionERC20Pool/libraries.sol#574)
        - >=0.6.0<0.8.0 (UnionERC20Pool/libraries.sol#802)
        - >=0.6.0 (UnionERC20Pool/libraries.sol#899)
        - >=0.6.2<0.8.0 (UnionERC20Pool/libraries.sol#935)
        - >=0.6.0<0.8.0 (UnionERC20Pool/libraries.sol#956)
        - >=0.6.0<0.8.0 (UnionERC20Pool/libraries.sol#1118)
        - >=0.6.0<0.8.0 (UnionERC20Pool/libraries.sol#1198)
        - >=0.6.0<0.8.0 (UnionERC20Pool/libraries.sol#1513)
        - >=0.6.2<0.8.0 (UnionERC20Pool/libraries.sol#1590)
        - >=0.6.12 (UnionERC20Pool/libraries.sol#1601)
        - >=0.6.6 (UnionERC20Pool/libraries.sol#1712)
        - >=0.6.12 (UnionERC20Pool/libraries.sol#1756)
        - >=0.6.2<0.8.0 (UnionERC20Pool/libraries.sol#2039)
```

Pic 3. Array length assignment:

```
194     function setDocument(
195         uint256 id,
196         bytes32 name,
197         string calldata uri,
198         bytes32 documentHash
199     )
200         external
201     {
202         require(name != bytes32(0), "Bad name");
203         require(bytes(uri).length > 0, "Bad uri");
204
205         if (protectionDocuments[id].document[name].lastModified == uint256(0)) {
206             protectionDocuments[id].docNames.push(name);
207             protectionDocuments[id].docIndexes[name] = protectionDocuments[id].docNames.length;
208         }
209         protectionDocuments[id].document[name] = Document(documentHash, now, uri);
210         emit DocumentUpdated(id, name, uri, documentHash);
211     }
212
213      /**
214      * @notice Used to remove an existing document from the contract by giving the name of the document.
215      * @dev Can only be executed by the owner of the contract.
216      * @param name Name of the document. It should be unique always
217      */
```

Pic 4. Reentrancy vulnerabilities:

```
161     function exercise(uint256 _id, uint256 _amount) public {
162         require(msg.sender == uunn.ownerOf(_id), "Caller is not the Owner of Protection");
163         uint256 validTo = protections[_id].timelimits & 0x0000000000000000000000000000000000000000000000000FFFFFFFFFFFFFFF;
164         require (now <= validTo,"Protection expired");
165         require(_amount <= protections[_id].amount,"Amount too high");
166         uint priceDecimals = protections[_id].pool.getPriceDecimals();
167         uint currentPrice = protections[_id].pool.getLatestPrice();
168 
169         require(protections[_id].strike >= currentPrice, "Current price is too high");
170         uint256 profit = protections[_id].strike.sub(currentPrice).mul(_amount).div(priceDecimals);
171 
172 
173         uint256 premiumToUnlock = profit < protections[_id].premium ? profit : protections[_id].premium;
174         protections[_id].premium = protections[_id].premium.sub(premiumToUnlock);
175 
176         protections[_id].pool.onPayoutCoverage(_id, premiumToUnlock, profit, msg.sender);
177 
178         // uunn.triggerCoveragePayment(_id, profit, _amount, protections[_id].amount, msg.sender, protections[_id].pool);
179         protections[_id].amount = protections[_id].amount.sub(_amount);
180         emit Exercised(_id, _amount, profit);
181     }
```

Pic 5. Different pragma directives are used:

```
Different versions of Solidity is used in :
        - Version used: ['>=0.4.24<0.8.0', '>=0.6.0<0.8.0', '>=0.6.12', '>=0.6.2<0.8.0', '>=0.6.6']
        - >=0.6.12 (OCProtections/OCProtections.sol#1)
        - >=0.4.24<0.8.0 (OCProtections/libraries.sol#4)
        - >=0.6.0<0.8.0 (OCProtections/libraries.sol#69)
        - >=0.6.2<0.8.0 (OCProtections/libraries.sol#369)
        - >=0.6.0<0.8.0 (OCProtections/libraries.sol#537)
        - >=0.6.0<0.8.0 (OCProtections/libraries.sol#572)
        - >=0.6.0<0.8.0 (OCProtections/libraries.sol#800)
        - >=0.6.0<0.8.0 (OCProtections/libraries.sol#899)
        - >=0.6.0<0.8.0 (OCProtections/libraries.sol#1061)
        - >=0.6.0<0.8.0 (OCProtections/libraries.sol#1141)
        - >=0.6.0<0.8.0 (OCProtections/libraries.sol#1218)
        - >=0.6.2<0.8.0 (OCProtections/libraries.sol#1245)
        - >=0.6.2<0.8.0 (OCProtections/libraries.sol#1376)
        - >=0.6.2<0.8.0 (OCProtections/libraries.sol#1407)
        - >=0.6.2<0.8.0 (OCProtections/libraries.sol#1421)
        - >=0.6.2<0.8.0 (OCProtections/libraries.sol#1442)
        - >=0.6.2<0.8.0 (OCProtections/libraries.sol#1454)
        - >=0.6.6 (OCProtections/libraries.sol#1472)
```

Appendix B. Automated tools reports

Pic 1. OCProtections Slither automated report:

```
INFO:Detectors:
OCProtections.withdrawPremium(uint256,uint256) (OCProtections/OCProtections.sol#155-159) uses a dangerous strict equality:
        - require(bool,string)(msg.sender == address(protections[_id].pool) && msg.sender != address(0),Premium can be withdrawn by backed pool only) (OCProtections/OCProtections.sol
#156)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dangerous-strict-equalities
INFO:Detectors:
Reentrancy in OCProtections.exercise(uint256,uint256) (OCProtections/OCProtections.sol#161-181):
        External calls:
        - protections[_id].pool.onPayoutCoverage(_id,premiumToUnlock,profit,msg.sender) (OCProtections/OCProtections.sol#176)
        State variables written after the call(s):
        - protections[_id].amount = protections[_id].amount.sub(_amount) (OCProtections/OCProtections.sol#179)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-1
INFO:Detectors:
OCProtections.createTo(address,uint256,uint256,uint256,uint256,uint256[11],bytes,address) (OCProtections/OCProtections.sol#104-153) ignores return value by IERC20Upgradeable(IPool(po
ol).getBasicToken()).approve(pool,ocdata[1]) (OCProtections/OCProtections.sol#133)
OCProtections.exercise(uint256,uint256) (OCProtections/OCProtections.sol#161-181) ignores return value by protections[_id].pool.onPayoutCoverage(_id,premiumToUnlock,profit,msg.sender
) (OCProtections/OCProtections.sol#176)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return
INFO:Detectors:
IUUNNRegistry.mint(uint256,address,address).protectionContract (OCProtections/libraries.sol#1412) shadows:
        - IUUNNRegistry.protectionContract(uint256) (OCProtections/libraries.sol#1414) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing
INFO:Detectors:
Reentrancy in OCProtections.createTo(address,uint256,uint256,uint256,uint256,uint256[11],bytes,address) (OCProtections/OCProtections.sol#104-153):
        External calls:
        - IERC20Upgradeable(IPool(pool).getBasicToken()).safeTransferFrom(msg.sender,address(this),ocdata[1]) (OCProtections/OCProtections.sol#132)
        - IERC20Upgradeable(IPool(pool).getBasicToken()).approve(pool,ocdata[1]) (OCProtections/OCProtections.sol#133)
        - IAssetPool(pool).onProtectionPremium(address(this),(data[0],data[1],coverage,data[3],data[7],data[8],data[9])) (OCProtections/OCProtections.sol#137)
        State variables written after the call(s):
        - protections[ocdata[0]] = OCProtectionData(IAssetPool(pool),timelimits,ocdata[3],ocdata[4],ocdata[1],0) (OCProtections/OCProtections.sol#144)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2
INFO:Detectors:
Reentrancy in OCProtections.createTo(address,uint256,uint256,uint256,uint256,uint256[11],bytes,address) (OCProtections/OCProtections.sol#104-153):
        External calls:
        - IERC20Upgradeable(IPool(pool).getBasicToken()).safeTransferFrom(msg.sender,address(this),ocdata[1]) (OCProtections/OCProtections.sol#132)
        - IERC20Upgradeable(IPool(pool).getBasicToken()).approve(pool,ocdata[1]) (OCProtections/OCProtections.sol#133)
        - IAssetPool(pool).onProtectionPremium(address(this),(data[0],data[1],coverage,data[3],data[7],data[8],data[9])) (OCProtections/OCProtections.sol#137)
        - uunn.mint(ocdata[0],address(this),erc721Receiver) (OCProtections/OCProtections.sol#145)
        Event emitted after the call(s):
        - OCProtectionCreated(erc721Receiver,ocdata[0],pool,ocdata[3],ocdata[4],now,ocdata[2],ocdata[1],ocdata[5]) (OCProtections/OCProtections.sol#149)
Reentrancy in OCProtections.exercise(uint256,uint256) (OCProtections/OCProtections.sol#161-181):
```

```
INFO:Detectors:
Reentrancy in OCProtections.createTo(address,uint256,uint256,uint256,uint256,uint256[11],bytes,address) (OCProtections/OCProtections.sol#104-153):
        External calls:
        - IERC20Upgradeable(IPool(pool).getBasicToken()).safeTransferFrom(msg.sender,address(this),ocdata[1]) (OCProtections/OCProtections.sol#132)
        - IERC20Upgradeable(IPool(pool).getBasicToken()).approve(pool,ocdata[1]) (OCProtections/OCProtections.sol#133)
        - IAssetPool(pool).onProtectionPremium(address(this),(data[0],data[1],coverage,data[3],data[7],data[8],data[9])) (OCProtections/OCProtections.sol#137)
        State variables written after the call(s):
        - protections[ocdata[0]] = OCProtectionData(IAssetPool(pool),timelimits,ocdata[3],ocdata[4],ocdata[1],0) (OCProtections/OCProtections.sol#144)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2
INFO:Detectors:
Reentrancy in OCProtections.createTo(address,uint256,uint256,uint256,uint256,uint256[11],bytes,address) (OCProtections/OCProtections.sol#104-153):
        External calls:
        - IERC20Upgradeable(IPool(pool).getBasicToken()).safeTransferFrom(msg.sender,address(this),ocdata[1]) (OCProtections/OCProtections.sol#132)
        - IERC20Upgradeable(IPool(pool).getBasicToken()).approve(pool,ocdata[1]) (OCProtections/OCProtections.sol#133)
        - IAssetPool(pool).onProtectionPremium(address(this),(data[0],data[1],coverage,data[3],data[7],data[8],data[9])) (OCProtections/OCProtections.sol#137)
        - uunn.mint(ocdata[0],address(this),erc721Receiver) (OCProtections/OCProtections.sol#145)
        Event emitted after the call(s):
        - OCProtectionCreated(erc721Receiver,ocdata[0],pool,ocdata[3],ocdata[4],now,ocdata[2],ocdata[1],ocdata[5]) (OCProtections/OCProtections.sol#149)
Reentrancy in OCProtections.exercise(uint256,uint256) (OCProtections/OCProtections.sol#161-181):
        External calls:
        - protections[_id].pool.onPayoutCoverage(_id,premiumToUnlock,profit,msg.sender) (OCProtections/OCProtections.sol#176)
        Event emitted after the call(s):
        - Exercised(_id,_amount,profit) (OCProtections/OCProtections.sol#180)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3
INFO:Detectors:
OCProtections.createTo(address,uint256,uint256,uint256,uint256,uint256[11],bytes,address) (OCProtections/OCProtections.sol#104-153) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(block.timestamp <= data[5],quotation expired) (OCProtections/OCProtections.sol#129)
OCProtections.withdrawPremium(uint256,uint256) (OCProtections/OCProtections.sol#155-159) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(msg.sender == address(protections[_id].pool) && msg.sender != address(0),Premium can be withdrawn by backed pool only) (OCProtections/OCProtections.sol#156)
        - require(bool,string)(protections[_id].premium <= _premium,Not enough premium left) (OCProtections/OCProtections.sol#157)
OCProtections.exercise(uint256,uint256) (OCProtections/OCProtections.sol#161-181) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(now <= validTo,Protection expired) (OCProtections/OCProtections.sol#164)
        - require(bool,string)(_amount <= protections[_id].amount,Amount too high) (OCProtections/OCProtections.sol#165)
        - require(bool,string)(protections[_id].strike >= currentPrice,Current price is too high) (OCProtections/OCProtections.sol#169)
        - profit < protections[_id].premium (OCProtections/OCProtections.sol#173)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp
```

```
INFO:Detectors:
Initializable._isConstructor() (OCProtections/libraries.sol#51-62) uses assembly
        - INLINE ASM (OCProtections/libraries.sol#60)
AddressUpgradeable.isContract(address) (OCProtections/libraries.sol#392-401) uses assembly
        - INLINE ASM (OCProtections/libraries.sol#399)
AddressUpgradeable._verifyCallResult(bool,bytes,string) (OCProtections/libraries.sol#513-530) uses assembly
        - INLINE ASM (OCProtections/libraries.sol#522-525)
SignLib.splitSignature(bytes) (OCProtections/libraries.sol#1490-1511) uses assembly
        - INLINE ASM (OCProtections/libraries.sol#1501-1508)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
INFO:Detectors:
Different versions of Solidity is used in :
        - Version used: ['>=0.4.24<0.8.0', '>=0.6.0<0.8.0', '>=0.6.12', '>=0.6.2<0.8.0', '>=0.6.6']
        - >=0.6.12 (OCProtections/OCProtections.sol#1)
        - >=0.4.24<0.8.0 (OCProtections/libraries.sol#4)
        - >=0.6.0<0.8.0 (OCProtections/libraries.sol#69)
        - >=0.6.2<0.8.0 (OCProtections/libraries.sol#369)
        - >=0.6.0<0.8.0 (OCProtections/libraries.sol#537)
        - >=0.6.0<0.8.0 (OCProtections/libraries.sol#572)
        - >=0.6.0<0.8.0 (OCProtections/libraries.sol#800)
        - >=0.6.0<0.8.0 (OCProtections/libraries.sol#899)
        - >=0.6.0<0.8.0 (OCProtections/libraries.sol#1061)
        - >=0.6.0<0.8.0 (OCProtections/libraries.sol#1141)
        - >=0.6.0<0.8.0 (OCProtections/libraries.sol#1218)
        - >=0.6.2<0.8.0 (OCProtections/libraries.sol#1245)
        - >=0.6.2<0.8.0 (OCProtections/libraries.sol#1376)
        - >=0.6.2<0.8.0 (OCProtections/libraries.sol#1407)
        - >=0.6.2<0.8.0 (OCProtections/libraries.sol#1421)
        - >=0.6.2<0.8.0 (OCProtections/libraries.sol#1442)
        - >=0.6.2<0.8.0 (OCProtections/libraries.sol#1454)
        - >=0.6.6 (OCProtections/libraries.sol#1472)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used
INFO:Detectors:
Pragma version>=0.6.12 (OCProtections/OCProtections.sol#1) necessitates a version too recent to be trusted. Consider deploying with 0.6.11
Pragma version>=0.4.24<0.8.0 (OCProtections/libraries.sol#4) is too complex
Pragma version>=0.6.0<0.8.0 (OCProtections/libraries.sol#69) is too complex
Pragma version>=0.6.2<0.8.0 (OCProtections/libraries.sol#369) is too complex
Pragma version>=0.6.0<0.8.0 (OCProtections/libraries.sol#537) is too complex
Pragma version>=0.6.0<0.8.0 (OCProtections/libraries.sol#572) is too complex
Pragma version>=0.6.0<0.8.0 (OCProtections/libraries.sol#800) is too complex
Pragma version>=0.6.0<0.8.0 (OCProtections/libraries.sol#899) is too complex
Pragma version>=0.6.0<0.8.0 (OCProtections/libraries.sol#1061) is too complex
```

```
INFO:Detectors:
ProtectionUpgradable.__Protection_init_unchained(uint256,address,uint256,uint256,address[],uint16[]) (OCProtection/libraries.sol#564-578) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(_validTo > now,Protection validTo is in the past) (OCProtection/libraries.sol#568)
ProtectionUpgradable._finalize() (OCProtection/libraries.sol#585-589) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)((block.timestamp >= validTo || amount == 0),Cannot finalize protection) (OCProtection/libraries.sol#586)
ProtectionUpgradable.setDocument(bytes32,string,bytes32) (OCProtection/libraries.sol#591-606) uses timestamp for comparisons
        Dangerous comparisons:
        - document[name].lastModified == uint256(0) (OCProtection/libraries.sol#600)
ProtectionUpgradable.removeDocument(bytes32) (OCProtection/libraries.sol#613-627) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(document[name].lastModified != uint256(0),Not existed) (OCProtection/libraries.sol#618)
ProtectionUpgradable.exercise(uint256) (OCProtection/libraries.sol#674-677) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(now <= validTo,Protection expired) (OCProtection/libraries.sol#675)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp
INFO:Detectors:
Initializable._isConstructor() (OCProtection/libraries.sol#53-64) uses assembly
        - INLINE ASM (OCProtection/libraries.sol#62)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
INFO:Detectors:
Different versions of Solidity is used in :
        - Version used: ['>=0.4.24<0.8.0', '>=0.6.0<0.8.0', '>=0.6.12', '>=0.6.2<0.8.0']
        - >=0.6.12 (OCProtection/OCProtection.sol#1)
        - >=0.4.24<0.8.0 (OCProtection/libraries.sol#6)
        - >=0.6.0<0.8.0 (OCProtection/libraries.sol#71)
        - >=0.6.0<0.8.0 (OCProtection/libraries.sol#233)
        - >=0.6.2<0.8.0 (OCProtection/libraries.sol#260)
        - >=0.6.2<0.8.0 (OCProtection/libraries.sol#391)
        - >=0.6.2<0.8.0 (OCProtection/libraries.sol#422)
        - >=0.6.2<0.8.0 (OCProtection/libraries.sol#439)
        - >=0.6.2<0.8.0 (OCProtection/libraries.sol#456)
        - >=0.6.2<0.8.0 (OCProtection/libraries.sol#498)
        - >=0.6.12 (OCProtection/libraries.sol#509)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used
```

```
INFO:Detectors:
Pragma version>=0.6.12 (OCProtections/OCProtections.sol#1) necessitates a version too recent to be trusted. Consider deploying with 0.6.11
Pragma version>=0.4.24<0.8.0 (OCProtections/libraries.sol#4) is too complex
Pragma version>=0.6.0<0.8.0 (OCProtections/libraries.sol#69) is too complex
Pragma version>=0.6.2<0.8.0 (OCProtections/libraries.sol#369) is too complex
Pragma version>=0.6.0<0.8.0 (OCProtections/libraries.sol#537) is too complex
Pragma version>=0.6.0<0.8.0 (OCProtections/libraries.sol#572) is too complex
Pragma version>=0.6.0<0.8.0 (OCProtections/libraries.sol#800) is too complex
Pragma version>=0.6.0<0.8.0 (OCProtections/libraries.sol#899) is too complex
Pragma version>=0.6.0<0.8.0 (OCProtections/libraries.sol#1061) is too complex
Pragma version>=0.6.0<0.8.0 (OCProtections/libraries.sol#1141) is too complex
Pragma version>=0.6.0<0.8.0 (OCProtections/libraries.sol#1218) is too complex
Pragma version>=0.6.2<0.8.0 (OCProtections/libraries.sol#1245) is too complex
Pragma version>=0.6.2<0.8.0 (OCProtections/libraries.sol#1376) is too complex
Pragma version>=0.6.2<0.8.0 (OCProtections/libraries.sol#1407) is too complex
Pragma version>=0.6.2<0.8.0 (OCProtections/libraries.sol#1421) is too complex
Pragma version>=0.6.2<0.8.0 (OCProtections/libraries.sol#1442) is too complex
Pragma version>=0.6.2<0.8.0 (OCProtections/libraries.sol#1454) is too complex
Pragma version>=0.6.6 (OCProtections/libraries.sol#1472) allows old versions
solc-0.6.12 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Low level call in AddressUpgradeable.sendValue(address,uint256) (OCProtections/libraries.sol#419-425):
        - (success) = recipient.call{value: amount}() (OCProtections/libraries.sol#423)
Low level call in AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (OCProtections/libraries.sol#480-487):
        - (success,returndata) = target.call{value: value}(data) (OCProtections/libraries.sol#485)
Low level call in AddressUpgradeable.functionStaticCall(address,bytes,string) (OCProtections/libraries.sol#505-511):
        - (success,returndata) = target.staticcall(data) (OCProtections/libraries.sol#509)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
INFO:Detectors:
OCProtections (OCProtections/OCProtections.sol#23-259) should inherit from IOCProtections (OCProtections/libraries.sol#1461-1468)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-inheritance
INFO:Detectors:
Parameter OCProtections.initialize(address,address)._admin (OCProtections/OCProtections.sol#70) is not in mixedCase
Parameter OCProtections.initialize(address,address)._uunn (OCProtections/OCProtections.sol#70) is not in mixedCase
Parameter OCProtections.withdrawPremium(uint256,uint256)._id (OCProtections/OCProtections.sol#155) is not in mixedCase
Parameter OCProtections.withdrawPremium(uint256,uint256)._premium (OCProtections/OCProtections.sol#155) is not in mixedCase
Parameter OCProtections.exercise(uint256,uint256)._id (OCProtections/OCProtections.sol#161) is not in mixedCase
Parameter OCProtections.exercise(uint256,uint256)._amount (OCProtections/OCProtections.sol#161) is not in mixedCase
Parameter OCProtections.getDocument(uint256,bytes32)._name (OCProtections/OCProtections.sol#242) is not in mixedCase
Variable OCProtections.PROTECTION_PREMIUM_DATA_PROVIDER (OCProtections/OCProtections.sol#30) is not in mixedCase
```

```
INFO:Detectors:
Redundant expression "this (OCProtections/libraries.sol#562)" inContextUpgradeable (OCProtections/libraries.sol#550-566)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements
INFO:Detectors:
OCProtections.exercise(uint256,uint256) (OCProtections/OCProtections.sol#161-181) uses literals with too many digits:
        - validTo = protections[_id].timelimits & 0x0000000000000000000000000000000000000000FFFFFFFFFFFFFFFF (OCProtections/OCProtections.sol#163)
OCProtections.getProtectionData(uint256) (OCProtections/OCProtections.sol#183-192) uses literals with too many digits:
        - (address(protections[id].pool),protections[id].amount,protections[id].strike,protections[id].premium,(protections[id].timelimits >> 64) & 0x0000000000000000000000000000000000000000FFFFFFFFFFFFFFFF,protections
[id].timelimits & 0x0000000000000000000000000000000000000000FFFFFFFFFFFFFFFF) (OCProtections/OCProtections.sol#184-191)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits
INFO:Detectors:
PausableUpgradeable.__gap (OCProtections/libraries.sol#892) is never used in OCProtections (OCProtections/OCProtections.sol#23-259)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variables
INFO:Detectors:
initialize(address,address) should be declared external:
        - OCProtections.initialize(address,address) (OCProtections/OCProtections.sol#70-77)
pause() should be declared external:
        - OCProtections.pause() (OCProtections/OCProtections.sol#90-92)
unpause() should be declared external:
        - OCProtections.unpause() (OCProtections/OCProtections.sol#96-98)
create(address,uint256,uint256,uint256,uint256[11],bytes) should be declared external:
        - OCProtections.create(address,uint256,uint256,uint256,uint256[11],bytes) (OCProtections/OCProtections.sol#100-102)
exercise(uint256,uint256) should be declared external:
        - OCProtections.exercise(uint256,uint256) (OCProtections/OCProtections.sol#161-181)
getProtectionData(uint256) should be declared external:
        - OCProtections.getProtectionData(uint256) (OCProtections/OCProtections.sol#183-192)
getRoleMemberCount(bytes32) should be declared external:
        - AccessControlUpgradeable.getRoleMemberCount(bytes32) (OCProtections/libraries.sol#671-673)
getRoleMember(bytes32,uint256) should be declared external:
        - AccessControlUpgradeable.getRoleMember(bytes32,uint256) (OCProtections/libraries.sol#687-689)
getRoleAdmin(bytes32) should be declared external:
        - AccessControlUpgradeable.getRoleAdmin(bytes32) (OCProtections/libraries.sol#697-699)
grantRole(bytes32,address) should be declared external:
        - AccessControlUpgradeable.grantRole(bytes32,address) (OCProtections/libraries.sol#711-715)
revokeRole(bytes32,address) should be declared external:
        - AccessControlUpgradeable.revokeRole(bytes32,address) (OCProtections/libraries.sol#726-730)
renounceRole(bytes32,address) should be declared external:
        - AccessControlUpgradeable.renounceRole(bytes32,address) (OCProtections/libraries.sol#746-750)
paused() should be declared external:
        - PausableUpgradeable.paused() (OCProtections/libraries.sol#841-843)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

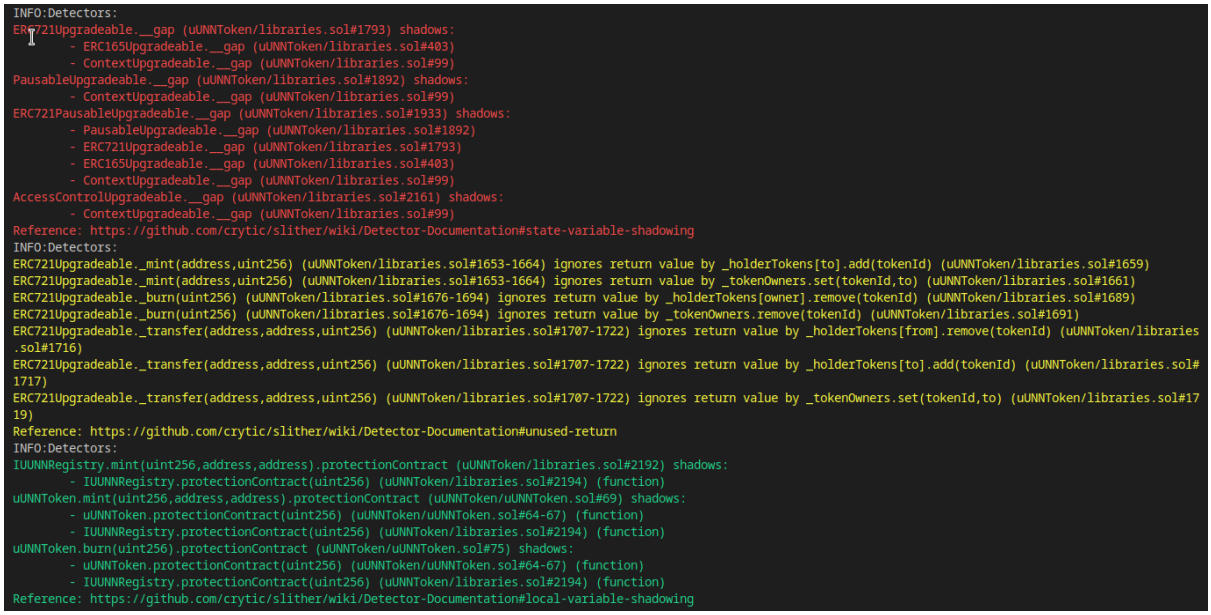
Pic 2. ProtectionUpgradable Slither automated report:

```
INFO:Detectors:
ProtectionUpgradable.beneficiaries (ProtectionUpgradable/ProtectionUpgradable.sol#24) is never initialized. It is used in:
        - ProtectionUpgradable.getBeneficiary(uint256) (ProtectionUpgradable/ProtectionUpgradable.sol#160-163)
        - ProtectionUpgradable.getBeneficiariesLength() (ProtectionUpgradable/ProtectionUpgradable.sol#165-167)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-state-variables
INFO:Detectors:
ProtectionUpgradable.setDocument(bytes32,string,bytes32) (ProtectionUpgradable/ProtectionUpgradable.sol#86-101) uses a dangerous strict equality:
        - document[name].lastModified == uint256(0) (ProtectionUpgradable/ProtectionUpgradable.sol#95)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dangerous-strict-equalities
INFO:Detectors:
IUUNNRegistry.mint(uint256,address,address).protectionContract (ProtectionUpgradable/libraries.sol#446) shadows:
        - IUUNNRegistry.protectionContract(uint256) (ProtectionUpgradable/libraries.sol#448) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing
INFO:Detectors:
ProtectionUpgradable.__Protection_init_unchained(uint256,address,uint256,uint256,address[],uint16[]) (ProtectionUpgradable/ProtectionUpgradable.sol#59-73) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(_validTo > now,Protection validTo is in the past) (ProtectionUpgradable/ProtectionUpgradable.sol#63)
ProtectionUpgradable._finalize() (ProtectionUpgradable/ProtectionUpgradable.sol#80-84) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)((block.timestamp >= validTo || amount == 0),Cannot finalize protection) (ProtectionUpgradable/ProtectionUpgradable.sol#81)
ProtectionUpgradable.setDocument(bytes32,string,bytes32) (ProtectionUpgradable/ProtectionUpgradable.sol#86-101) uses timestamp for comparisons
        Dangerous comparisons:
        - document[name].lastModified == uint256(0) (ProtectionUpgradable/ProtectionUpgradable.sol#95)
ProtectionUpgradable.removeDocument(bytes32) (ProtectionUpgradable/ProtectionUpgradable.sol#108-122) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(document[name].lastModified != uint256(0),Not existed) (ProtectionUpgradable/ProtectionUpgradable.sol#113)
ProtectionUpgradable.exercise(uint256) (ProtectionUpgradable/ProtectionUpgradable.sol#169-172) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(now <= validTo,Protection expired) (ProtectionUpgradable/ProtectionUpgradable.sol#170)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp
```

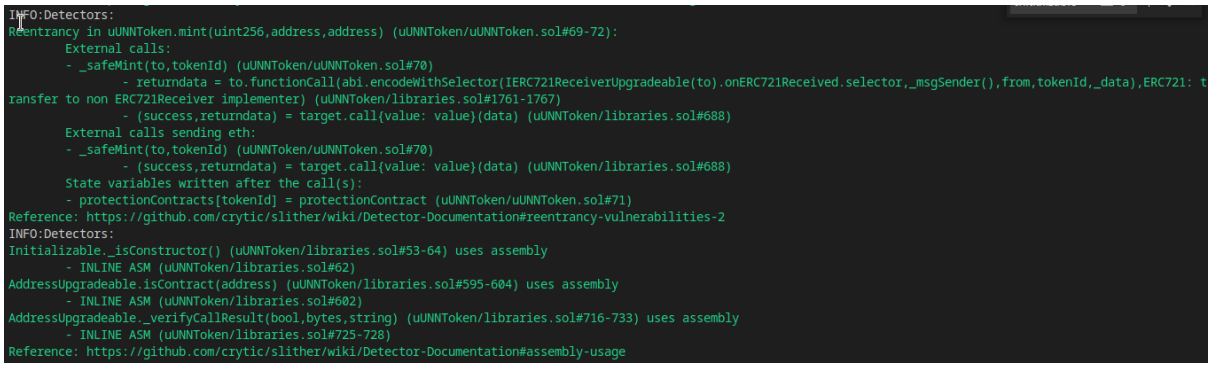
```
INFO:Detectors:
Initializable._isConstructor() (ProtectionUpgradable/libraries.sol#53-64) uses assembly
        - INLINE ASM (ProtectionUpgradable/libraries.sol#62)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
INFO:Detectors:
Different versions of Solidity is used in :
        - Version used: ['>=0.4.24<0.8.0', '>=0.6.0<0.8.0', '>=0.6.12', '>=0.6.2<0.8.0']
        - >=0.6.12 (ProtectionUpgradable/ProtectionUpgradable.sol#1)
        - >=0.4.24<0.8.0 (ProtectionUpgradable/libraries.sol#6)
        - >=0.6.0<0.8.0 (ProtectionUpgradable/libraries.sol#71)
        - >=0.6.0<0.8.0 (ProtectionUpgradable/libraries.sol#233)
        - >=0.6.2<0.8.0 (ProtectionUpgradable/libraries.sol#260)
        - >=0.6.2<0.8.0 (ProtectionUpgradable/libraries.sol#391)
        - >=0.6.2<0.8.0 (ProtectionUpgradable/libraries.sol#422)
        - >=0.6.2<0.8.0 (ProtectionUpgradable/libraries.sol#439)
        - >=0.6.2<0.8.0 (ProtectionUpgradable/libraries.sol#456)
        - >=0.6.2<0.8.0 (ProtectionUpgradable/libraries.sol#498)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used
INFO:Detectors:
Pragma version>=0.6.12 (ProtectionUpgradable/ProtectionUpgradable.sol#1) necessitates a version too recent to be trusted. Consider deploying with 0.6.11
Pragma version>=0.4.24<0.8.0 (ProtectionUpgradable/libraries.sol#6) is too complex
Pragma version>=0.6.0<0.8.0 (ProtectionUpgradable/libraries.sol#71) is too complex
Pragma version>=0.6.0<0.8.0 (ProtectionUpgradable/libraries.sol#233) is too complex
Pragma version>=0.6.2<0.8.0 (ProtectionUpgradable/libraries.sol#260) is too complex
Pragma version>=0.6.2<0.8.0 (ProtectionUpgradable/libraries.sol#391) is too complex
Pragma version>=0.6.2<0.8.0 (ProtectionUpgradable/libraries.sol#422) is too complex
Pragma version>=0.6.2<0.8.0 (ProtectionUpgradable/libraries.sol#439) is too complex
Pragma version>=0.6.2<0.8.0 (ProtectionUpgradable/libraries.sol#456) is too complex
Pragma version>=0.6.2<0.8.0 (ProtectionUpgradable/libraries.sol#498) is too complex
solc-0.6.12 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
```

```
INFO:Detectors:
Function ProtectionUpgradable.__Protection_init(uint256,address,uint256,uint256,address[],uint16[]) (ProtectionUpgradable/ProtectionUpgradable.sol#55-57) is not in mixedCase
Parameter ProtectionUpgradable.__Protection_init(uint256,address,uint256,uint256,address[],uint16[])._id (ProtectionUpgradable/ProtectionUpgradable.sol#55) is not in mixedCase
Parameter ProtectionUpgradable.__Protection_init(uint256,address,uint256,uint256,address[],uint16[])._uunn (ProtectionUpgradable/ProtectionUpgradable.sol#55) is not in mixedCase
Parameter ProtectionUpgradable.__Protection_init(uint256,address,uint256,uint256,address[],uint16[])._amount (ProtectionUpgradable/ProtectionUpgradable.sol#55) is not in mixedCase
Parameter ProtectionUpgradable.__Protection_init(uint256,address,uint256,uint256,address[],uint16[])._validTo (ProtectionUpgradable/ProtectionUpgradable.sol#55) is not in mixedCase
Parameter ProtectionUpgradable.__Protection_init(uint256,address,uint256,uint256,address[],uint16[])._pools (ProtectionUpgradable/ProtectionUpgradable.sol#55) is not in mixedCase
Parameter ProtectionUpgradable.__Protection_init(uint256,address,uint256,uint256,address[],uint16[])._poolShares (ProtectionUpgradable/ProtectionUpgradable.sol#55) is not in mixedCase
Function ProtectionUpgradable.__Protection_init_unchained(uint256,address,uint256,uint256,address[],uint16[]) (ProtectionUpgradable/ProtectionUpgradable.sol#59-73) is not in mixedCase
Parameter ProtectionUpgradable.__Protection_init_unchained(uint256,address,uint256,uint256,address[],uint16[])._id (ProtectionUpgradable/ProtectionUpgradable.sol#59) is not in mixedCase
Parameter ProtectionUpgradable.__Protection_init_unchained(uint256,address,uint256,uint256,address[],uint16[])._uunn (ProtectionUpgradable/ProtectionUpgradable.sol#59) is not in mixedCase
Parameter ProtectionUpgradable.__Protection_init_unchained(uint256,address,uint256,uint256,address[],uint16[])._amount (ProtectionUpgradable/ProtectionUpgradable.sol#59) is not in mixedCase
Parameter ProtectionUpgradable.__Protection_init_unchained(uint256,address,uint256,uint256,address[],uint16[])._validTo (ProtectionUpgradable/ProtectionUpgradable.sol#59) is not in mixedCase
Parameter ProtectionUpgradable.__Protection_init_unchained(uint256,address,uint256,uint256,address[],uint16[])._pools (ProtectionUpgradable/ProtectionUpgradable.sol#59) is not in mixedCase
Parameter ProtectionUpgradable.__Protection_init_unchained(uint256,address,uint256,uint256,address[],uint16[])._poolShares (ProtectionUpgradable/ProtectionUpgradable.sol#59) is not in mixedCase
Parameter ProtectionUpgradable.getDocument(bytes32)._name (ProtectionUpgradable/ProtectionUpgradable.sol#131) is not in mixedCase
Parameter ProtectionUpgradable.getPool(uint256)._index (ProtectionUpgradable/ProtectionUpgradable.sol#147) is not in mixedCase
Parameter ProtectionUpgradable.getBeneficiary(uint256)._index (ProtectionUpgradable/ProtectionUpgradable.sol#160) is not in mixedCase
Parameter ProtectionUpgradable.exercise(uint256)._amount (ProtectionUpgradable/ProtectionUpgradable.sol#169) is not in mixedCase
Variable ProtectionUpgradable.__gap (ProtectionUpgradable/ProtectionUpgradable.sol#174) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
ProtectionUpgradable.__gap (ProtectionUpgradable/ProtectionUpgradable.sol#174) is never used in ProtectionUpgradable (ProtectionUpgradable/ProtectionUpgradable.sol#16-175)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variables
INFO:Detectors:
version() should be declared external:
        - ProtectionUpgradable.version() (ProtectionUpgradable/ProtectionUpgradable.sol#50-53)
exercise(uint256) should be declared external:
        - ProtectionUpgradable.exercise(uint256) (ProtectionUpgradable/ProtectionUpgradable.sol#169-172)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:ProtectionUpgradable/ProtectionUpgradable.sol analyzed (10 contracts with 46 detectors), 43 result(s) found
```

Pic 3. uUNNToken Slither automated report:



```
INFO:Detectors:
ERC721Upgradeable.__gap (uUNNToken/libraries.sol#1793) shadows:
        - ERC165Upgradeable.__gap (uUNNToken/libraries.sol#403)
        - ContextUpgradeable.__gap (uUNNToken/libraries.sol#99)
PausableUpgradeable.__gap (uUNNToken/libraries.sol#1892) shadows:
        - ContextUpgradeable.__gap (uUNNToken/libraries.sol#99)
ERC721PausableUpgradeable.__gap (uUNNToken/libraries.sol#1933) shadows:
        - PausableUpgradeable.__gap (uUNNToken/libraries.sol#1892)
        - ERC721Upgradeable.__gap (uUNNToken/libraries.sol#1793)
        - ERC165Upgradeable.__gap (uUNNToken/libraries.sol#403)
        - ContextUpgradeable.__gap (uUNNToken/libraries.sol#99)
AccessControlUpgradeable.__gap (uUNNToken/libraries.sol#2161) shadows:
        - ContextUpgradeable.__gap (uUNNToken/libraries.sol#99)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variable-shadowing
INFO:Detectors:
ERC721Upgradeable._mint(address,uint256) (uUNNToken/libraries.sol#1653-1664) ignores return value by _holderTokens[to].add(tokenId) (uUNNToken/libraries.sol#1659)
ERC721Upgradeable._mint(address,uint256) (uUNNToken/libraries.sol#1653-1664) ignores return value by _tokenOwners.set(tokenId,to) (uUNNToken/libraries.sol#1661)
ERC721Upgradeable._burn(uint256) (uUNNToken/libraries.sol#1676-1694) ignores return value by _holderTokens[owner].remove(tokenId) (uUNNToken/libraries.sol#1689)
ERC721Upgradeable._burn(uint256) (uUNNToken/libraries.sol#1676-1694) ignores return value by _tokenOwners.remove(tokenId) (uUNNToken/libraries.sol#1691)
ERC721Upgradeable._transfer(address,address,uint256) (uUNNToken/libraries.sol#1707-1722) ignores return value by _holderTokens[from].remove(tokenId) (uUNNToken/libraries
.sol#1716)
ERC721Upgradeable._transfer(address,address,uint256) (uUNNToken/libraries.sol#1707-1722) ignores return value by _holderTokens[to].add(tokenId) (uUNNToken/libraries.sol#
1717)
ERC721Upgradeable._transfer(address,address,uint256) (uUNNToken/libraries.sol#1707-1722) ignores return value by _tokenOwners.set(tokenId,to) (uUNNToken/libraries.sol#17
19)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return
INFO:Detectors:
IUUNNRegistry.mint(uint256,address,address).protectionContract (uUNNToken/libraries.sol#2192) shadows:
        - IUUNNRegistry.protectionContract(uint256) (uUNNToken/libraries.sol#2194) (function)
uUNNToken.mint(uint256,address,address).protectionContract (uUNNToken/uUNNToken.sol#69) shadows:
        - uUNNToken.protectionContract(uint256) (uUNNToken/uUNNToken.sol#64-67) (function)
        - IUUNNRegistry.protectionContract(uint256) (uUNNToken/libraries.sol#2194) (function)
uUNNToken.burn(uint256).protectionContract (uUNNToken/uUNNToken.sol#75) shadows:
        - uUNNToken.protectionContract(uint256) (uUNNToken/uUNNToken.sol#64-67) (function)
        - IUUNNRegistry.protectionContract(uint256) (uUNNToken/libraries.sol#2194) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing
```



```
INFO:Detectors:
Reentrancy in uUNNToken.mint(uint256,address,address) (uUNNToken/uUNNToken.sol#69-72):
        External calls:
        - _safeMint(to,tokenId) (uUNNToken/uUNNToken.sol#70)
                - returndata = to.functionCall(abi.encodeWithSelector(IERC721ReceiverUpgradeable(to).onERC721Received.selector,_msgSender(),from,tokenId,_data),ERC721: t
ransfer to non ERC721Receiver implementer) (uUNNToken/libraries.sol#1761-1767)
                - (success,returndata) = target.call{value: value}(data) (uUNNToken/libraries.sol#688)
        External calls sending eth:
        - _safeMint(to,tokenId) (uUNNToken/uUNNToken.sol#70)
                - (success,returndata) = target.call{value: value}(data) (uUNNToken/libraries.sol#688)
        State variables written after the call(s):
        - protectionContracts[tokenId] = protectionContract (uUNNToken/uUNNToken.sol#71)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2
INFO:Detectors:
Initializable._isConstructor() (uUNNToken/libraries.sol#53-64) uses assembly
        - INLINE ASM (uUNNToken/libraries.sol#62)
AddressUpgradeable.isContract(address) (uUNNToken/libraries.sol#595-604) uses assembly
        - INLINE ASM (uUNNToken/libraries.sol#602)
AddressUpgradeable._verifyCallResult(bool,bytes,string) (uUNNToken/libraries.sol#716-733) uses assembly
        - INLINE ASM (uUNNToken/libraries.sol#725-728)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
```

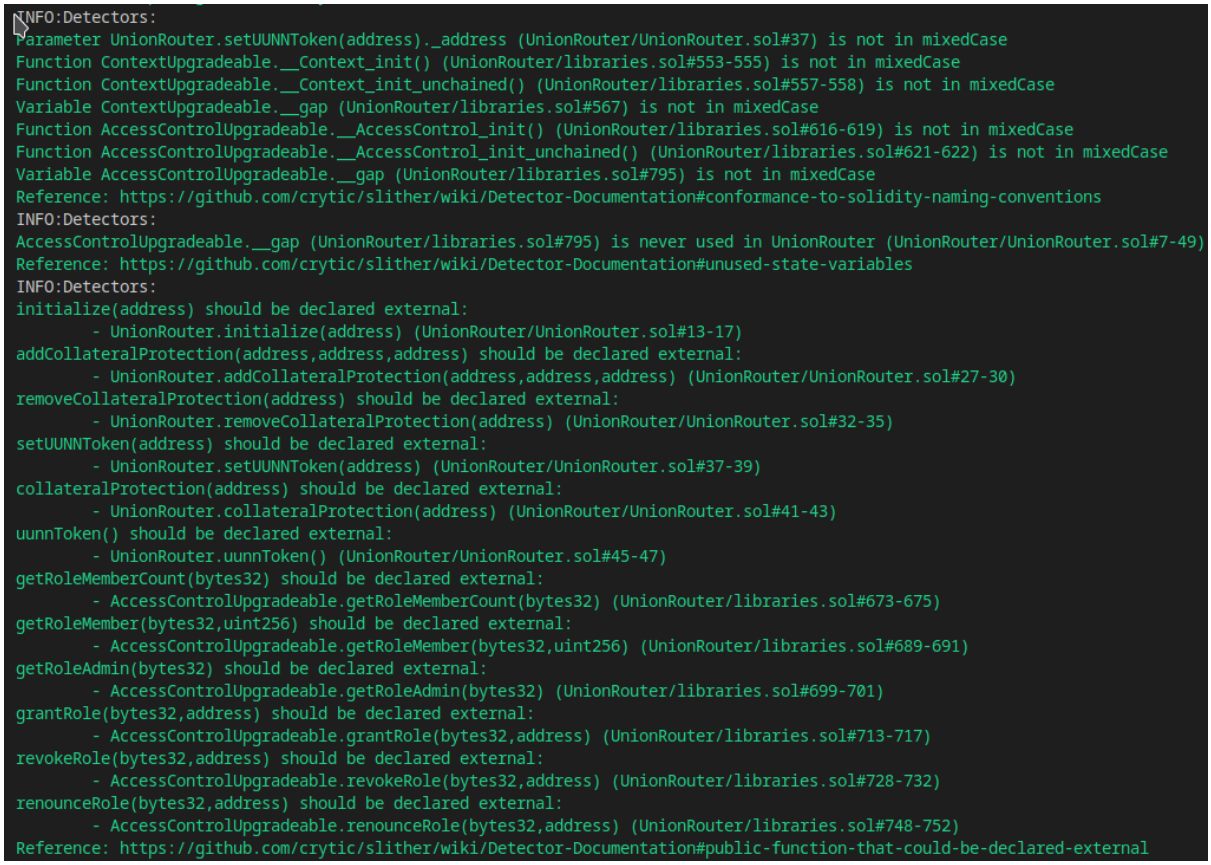
```
INFO:Detectors:
Different versions of Solidity is used in :
        - Version used: ['>=0.4.24<0.8.0', '>=0.6.0<0.8.0', '>=0.6.12', '>=0.6.2<0.8.0']
        - >=0.4.24<0.8.0 (uUNNToken/libraries.sol#6)
        - >=0.6.0<0.8.0 (uUNNToken/libraries.sol#71)
        - >=0.6.0<0.8.0 (uUNNToken/libraries.sol#106)
        - >=0.6.2<0.8.0 (uUNNToken/libraries.sol#133)
        - >=0.6.2<0.8.0 (uUNNToken/libraries.sol#264)
        - >=0.6.2<0.8.0 (uUNNToken/libraries.sol#293)
        - >=0.6.0<0.8.0 (uUNNToken/libraries.sol#324)
        - >=0.6.0<0.8.0 (uUNNToken/libraries.sol#348)
        - >=0.6.0<0.8.0 (uUNNToken/libraries.sol#410)
        - >=0.6.2<0.8.0 (uUNNToken/libraries.sol#572)
        - >=0.6.0<0.8.0 (uUNNToken/libraries.sol#740)
        - >=0.6.0<0.8.0 (uUNNToken/libraries.sol#1040)
        - >=0.6.0<0.8.0 (uUNNToken/libraries.sol#1280)
        - >=0.6.0<0.8.0 (uUNNToken/libraries.sol#1317)
        - >=0.6.0<0.8.0 (uUNNToken/libraries.sol#1800)
        - >=0.6.0<0.8.0 (uUNNToken/libraries.sol#1899)
        - >=0.6.0<0.8.0 (uUNNToken/libraries.sol#1940)
        - >=0.6.2<0.8.0 (uUNNToken/libraries.sol#2168)
        - >=0.6.2<0.8.0 (uUNNToken/libraries.sol#2185)
        - >=0.6.2<0.8.0 (uUNNToken/libraries.sol#2202)
        - >=0.6.12 (uUNNToken/uUNNToken.sol#1)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used
INFO:Detectors:
Pragma version>=0.4.24<0.8.0 (uUNNToken/libraries.sol#6) is too complex
Pragma version>=0.6.0<0.8.0 (uUNNToken/libraries.sol#71) is too complex
Pragma version>=0.6.0<0.8.0 (uUNNToken/libraries.sol#106) is too complex
Pragma version>=0.6.2<0.8.0 (uUNNToken/libraries.sol#133) is too complex
Pragma version>=0.6.2<0.8.0 (uUNNToken/libraries.sol#264) is too complex
Pragma version>=0.6.2<0.8.0 (uUNNToken/libraries.sol#293) is too complex
Pragma version>=0.6.0<0.8.0 (uUNNToken/libraries.sol#324) is too complex
Pragma version>=0.6.0<0.8.0 (uUNNToken/libraries.sol#348) is too complex
Pragma version>=0.6.0<0.8.0 (uUNNToken/libraries.sol#410) is too complex
Pragma version>=0.6.2<0.8.0 (uUNNToken/libraries.sol#572) is too complex
Pragma version>=0.6.0<0.8.0 (uUNNToken/libraries.sol#740) is too complex
Pragma version>=0.6.0<0.8.0 (uUNNToken/libraries.sol#1040) is too complex
Pragma version>=0.6.0<0.8.0 (uUNNToken/libraries.sol#1280) is too complex
Pragma version>=0.6.0<0.8.0 (uUNNToken/libraries.sol#1317) is too complex
Pragma version>=0.6.0<0.8.0 (uUNNToken/libraries.sol#1800) is too complex
Pragma version>=0.6.0<0.8.0 (uUNNToken/libraries.sol#1899) is too complex
```

```
INFO:Detectors:
Low level call in AddressUpgradeable.sendValue(address,uint256) (uUNNToken/libraries.sol#622-628):
        - (success) = recipient.call{value: amount}() (uUNNToken/libraries.sol#626)
Low level call in AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (uUNNToken/libraries.sol#683-690):
        - (success,returndata) = target.call{value: value}(data) (uUNNToken/libraries.sol#688)
Low level call in AddressUpgradeable.functionStaticCall(address,bytes,string) (uUNNToken/libraries.sol#708-714):
        - (success,returndata) = target.staticcall(data) (uUNNToken/libraries.sol#712)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
INFO:Detectors:
Function ContextUpgradeable.__Context_init() (uUNNToken/libraries.sol#85-87) is not in mixedCase
Function ContextUpgradeable.__Context_init_unchained() (uUNNToken/libraries.sol#89-90) is not in mixedCase
Variable ContextUpgradeable.__gap (uUNNToken/libraries.sol#99) is not in mixedCase
Function ERC165Upgradeable.__ERC165_init() (uUNNToken/libraries.sol#369-371) is not in mixedCase
Function ERC165Upgradeable.__ERC165_init_unchained() (uUNNToken/libraries.sol#373-377) is not in mixedCase
Variable ERC165Upgradeable.__gap (uUNNToken/libraries.sol#403) is not in mixedCase
Function ERC721Upgradeable.__ERC721_init(string,string) (uUNNToken/libraries.sol#1407-1411) is not in mixedCase
Function ERC721Upgradeable.__ERC721_init_unchained(string,string) (uUNNToken/libraries.sol#1413-1421) is not in mixedCase
Parameter ERC721Upgradeable.safeTransferFrom(address,address,uint256,bytes)._data (uUNNToken/libraries.sol#1565) is not in mixedCase
Variable ERC721Upgradeable.__gap (uUNNToken/libraries.sol#1793) is not in mixedCase
Function PausableUpgradeable.__Pausable_init() (uUNNToken/libraries.sol#1829-1832) is not in mixedCase
Function PausableUpgradeable.__Pausable_init_unchained() (uUNNToken/libraries.sol#1834-1836) is not in mixedCase
Variable PausableUpgradeable.__gap (uUNNToken/libraries.sol#1892) is not in mixedCase
Function ERC721PausableUpgradeable.__ERC721Pausable_init() (uUNNToken/libraries.sol#1912-1917) is not in mixedCase
Function ERC721PausableUpgradeable.__ERC721Pausable_init_unchained() (uUNNToken/libraries.sol#1919-1920) is not in mixedCase
Variable ERC721PausableUpgradeable.__gap (uUNNToken/libraries.sol#1933) is not in mixedCase
Function AccessControlUpgradeable.__AccessControl_init() (uUNNToken/libraries.sol#1982-1985) is not in mixedCase
Function AccessControlUpgradeable.__AccessControl_init_unchained() (uUNNToken/libraries.sol#1987-1988) is not in mixedCase
Variable AccessControlUpgradeable.__gap (uUNNToken/libraries.sol#2161) is not in mixedCase
Contract uUNNToken (uUNNToken/uUNNToken.sol#9-86) is not in CapWords
Function uUNNToken.__uUNNToken_init(address) (uUNNToken/uUNNToken.sol#16-21) is not in mixedCase
Parameter uUNNToken.__uUNNToken_init(address)._admin (uUNNToken/uUNNToken.sol#16) is not in mixedCase
Function uUNNToken.__uUNNToken_init_unchained(address) (uUNNToken/uUNNToken.sol#23-27) is not in mixedCase
Parameter uUNNToken.__uUNNToken_init_unchained(address)._admin (uUNNToken/uUNNToken.sol#23) is not in mixedCase
Parameter uUNNToken.initialize(address)._admin (uUNNToken/uUNNToken.sol#29) is not in mixedCase
Parameter uUNNToken.triggerCoveragePayment(uint256,uint256,uint256,uint256,address,IPool)._tokenId (uUNNToken/uUNNToken.sol#81) is not in mixedCase
Parameter uUNNToken.triggerCoveragePayment(uint256,uint256,uint256,uint256,address,IPool)._coverageToPay (uUNNToken/uUNNToken.sol#81) is not in mixedCase
Parameter uUNNToken.triggerCoveragePayment(uint256,uint256,uint256,uint256,address,IPool)._paidPartNom (uUNNToken/uUNNToken.sol#81) is not in mixedCase
Parameter uUNNToken.triggerCoveragePayment(uint256,uint256,uint256,uint256,address,IPool)._paidPartDenom (uUNNToken/uUNNToken.sol#81) is not in mixedCase
Parameter uUNNToken.triggerCoveragePayment(uint256,uint256,uint256,uint256,address,IPool)._beneficiary (uUNNToken/uUNNToken.sol#81) is not in mixedCase
Parameter uUNNToken.triggerCoveragePayment(uint256,uint256,uint256,uint256,address,IPool)._pool (uUNNToken/uUNNToken.sol#81) is not in mixedCase
Variable uUNNToken.PROTECTION_FACTORY_ROLE (uUNNToken/uUNNToken.sol#11) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
```

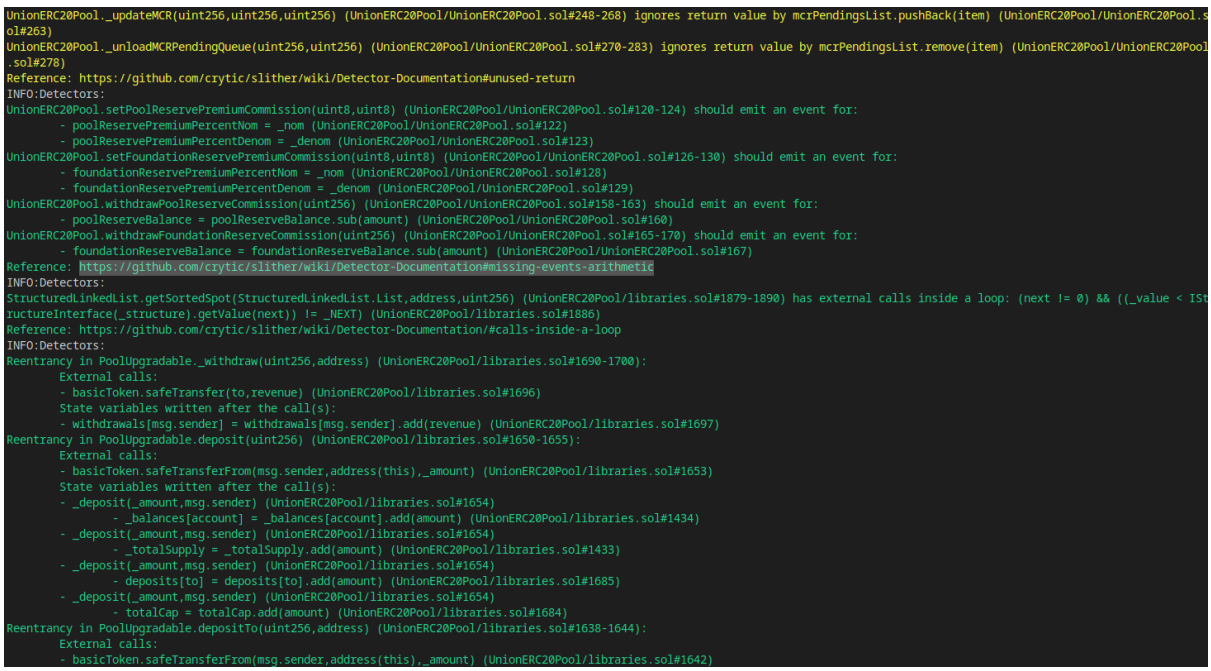
```
INFO:Detectors:
supportsInterface(bytes4) should be declared external:
        - ERC165Upgradeable.supportsInterface(bytes4) (uUNNToken/libraries.sol#384-386)
balanceOf(address) should be declared external:
        - ERC721Upgradeable.balanceOf(address) (uUNNToken/libraries.sol#1426-1430)
name() should be declared external:
        - ERC721Upgradeable.name() (uUNNToken/libraries.sol#1442-1444)
symbol() should be declared external:
        - ERC721Upgradeable.symbol() (uUNNToken/libraries.sol#1449-1451)
tokenURI(uint256) should be declared external:
        - ERC721Upgradeable.tokenURI(uint256) (uUNNToken/libraries.sol#1456-1471)
baseURI() should be declared external:
        - ERC721Upgradeable.baseURI() (uUNNToken/libraries.sol#1478-1480)
tokenOfOwnerByIndex(address,uint256) should be declared external:
        - ERC721Upgradeable.tokenOfOwnerByIndex(address,uint256) (uUNNToken/libraries.sol#1485-1487)
totalSupply() should be declared external:
        - ERC721Upgradeable.totalSupply() (uUNNToken/libraries.sol#1492-1495)
tokenByIndex(uint256) should be declared external:
        - ERC721Upgradeable.tokenByIndex(uint256) (uUNNToken/libraries.sol#1500-1503)
approve(address,uint256) should be declared external:
        - ERC721Upgradeable.approve(address,uint256) (uUNNToken/libraries.sol#1508-1517)
setApprovalForAll(address,bool) should be declared external:
        - ERC721Upgradeable.setApprovalForAll(address,bool) (uUNNToken/libraries.sol#1531-1536)
transferFrom(address,address,uint256) should be declared external:
        - ERC721Upgradeable.transferFrom(address,address,uint256) (uUNNToken/libraries.sol#1548-1553)
safeTransferFrom(address,address,uint256) should be declared external:
        - ERC721Upgradeable.safeTransferFrom(address,address,uint256) (uUNNToken/libraries.sol#1558-1560)
getRoleMemberCount(bytes32) should be declared external:
        - AccessControlUpgradeable.getRoleMemberCount(bytes32) (uUNNToken/libraries.sol#2039-2041)
getRoleMember(bytes32,uint256) should be declared external:
        - AccessControlUpgradeable.getRoleMember(bytes32,uint256) (uUNNToken/libraries.sol#2055-2057)
getRoleAdmin(bytes32) should be declared external:
        - AccessControlUpgradeable.getRoleAdmin(bytes32) (uUNNToken/libraries.sol#2065-2067)
grantRole(bytes32,address) should be declared external:
        - AccessControlUpgradeable.grantRole(bytes32,address) (uUNNToken/libraries.sol#2079-2083)
revokeRole(bytes32,address) should be declared external:
        - AccessControlUpgradeable.revokeRole(bytes32,address) (uUNNToken/libraries.sol#2094-2098)
renounceRole(bytes32,address) should be declared external:
        - AccessControlUpgradeable.renounceRole(bytes32,address) (uUNNToken/libraries.sol#2114-2118)
initialize(address) should be declared external:
        - uUNNToken.initialize(address) (uUNNToken/uUNNToken.sol#29-31)
version() should be declared external:
```

Pic 4. UnionRouter Slither automated report:

```
INFO:Detectors:
AccessControlUpgradeable.__gap (UnionRouter/libraries.sol#795) shadows:
        - ContextUpgradeable.__gap (UnionRouter/libraries.sol#567)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variable-shadowing
INFO:Detectors:
AddressUpgradeable.isContract(address) (UnionRouter/libraries.sol#328-337) uses assembly
        - INLINE ASM (UnionRouter/libraries.sol#335)
AddressUpgradeable._verifyCallResult(bool,bytes,string) (UnionRouter/libraries.sol#449-466) uses assembly
        - INLINE ASM (UnionRouter/libraries.sol#458-461)
Initializable._isConstructor() (UnionRouter/libraries.sol#521-532) uses assembly
        - INLINE ASM (UnionRouter/libraries.sol#530)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
INFO:Detectors:
Different versions of Solidity is used in :
        - Version used: ['>=0.4.24<0.8.0', '>=0.6.0<0.8.0', '>=0.6.12', '>=0.6.2<0.8.0']
        - >=0.6.12 (UnionRouter/UnionRouter.sol#2)
        - >=0.6.0<0.8.0 (UnionRouter/libraries.sol#5)
        - >=0.6.2<0.8.0 (UnionRouter/libraries.sol#305)
        - >=0.4.24<0.8.0 (UnionRouter/libraries.sol#474)
        - >=0.6.0<0.8.0 (UnionRouter/libraries.sol#539)
        - >=0.6.0<0.8.0 (UnionRouter/libraries.sol#574)
        - >=0.6.2<0.8.0 (UnionRouter/libraries.sol#802)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used
INFO:Detectors:
Pragma version>=0.6.12 (UnionRouter/UnionRouter.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.11
Pragma version>=0.6.0<0.8.0 (UnionRouter/libraries.sol#5) is too complex
Pragma version>=0.6.2<0.8.0 (UnionRouter/libraries.sol#305) is too complex
Pragma version>=0.4.24<0.8.0 (UnionRouter/libraries.sol#474) is too complex
Pragma version>=0.6.0<0.8.0 (UnionRouter/libraries.sol#539) is too complex
Pragma version>=0.6.0<0.8.0 (UnionRouter/libraries.sol#574) is too complex
Pragma version>=0.6.2<0.8.0 (UnionRouter/libraries.sol#802) is too complex
solc-0.6.12 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Low level call in AddressUpgradeable.sendValue(address,uint256) (UnionRouter/libraries.sol#355-361):
        - (success) = recipient.call{value: amount}() (UnionRouter/libraries.sol#359)
Low level call in AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (UnionRouter/libraries.sol#416-423):
        - (success,returndata) = target.call{value: value}(data) (UnionRouter/libraries.sol#421)
Low level call in AddressUpgradeable.functionStaticCall(address,bytes,string) (UnionRouter/libraries.sol#441-447):
        - (success,returndata) = target.staticcall(data) (UnionRouter/libraries.sol#445)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
```



Pic 5. UnionERC20Pool Slither automated report:



```
        - _deposit(_amount,_to) (UnionERC20Pool/libraries.sol#1643)
                - _balances[account] = _balances[account].add(amount) (UnionERC20Pool/libraries.sol#1434)
        - _deposit(_amount,_to) (UnionERC20Pool/libraries.sol#1643)
                - _totalSupply = _totalSupply.add(amount) (UnionERC20Pool/libraries.sol#1433)
        - _deposit(_amount,_to) (UnionERC20Pool/libraries.sol#1643)
                - deposits[to] = deposits[to].add(amount) (UnionERC20Pool/libraries.sol#1685)
        - _deposit(_amount,_to) (UnionERC20Pool/libraries.sol#1643)
                - totalCap = totalCap.add(amount) (UnionERC20Pool/libraries.sol#1684)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2
INFO:Detectors:
Reentrancy in PoolUpgradable._withdraw(uint256,address) (UnionERC20Pool/libraries.sol#1690-1700):
        External calls:
        - basicToken.safeTransfer(to,revenue) (UnionERC20Pool/libraries.sol#1696)
        Event emitted after the call(s):
        - Withdraw(msg.sender,revenue) (UnionERC20Pool/libraries.sol#1698)
Reentrancy in PoolUpgradable.deposit(uint256) (UnionERC20Pool/libraries.sol#1650-1655):
        External calls:
        - basicToken.safeTransferFrom(msg.sender,address(this),_amount) (UnionERC20Pool/libraries.sol#1653)
        Event emitted after the call(s):
        - Deposit(to,amount) (UnionERC20Pool/libraries.sol#1686)
                - _deposit(_amount,msg.sender) (UnionERC20Pool/libraries.sol#1654)
        - Transfer(address(0),account,amount) (UnionERC20Pool/libraries.sol#1435)
                - _deposit(_amount,msg.sender) (UnionERC20Pool/libraries.sol#1654)
Reentrancy in PoolUpgradable.depositTo(uint256,address) (UnionERC20Pool/libraries.sol#1638-1644):
        External calls:
        - basicToken.safeTransferFrom(msg.sender,address(this),_amount) (UnionERC20Pool/libraries.sol#1642)
        Event emitted after the call(s):
        - Deposit(to,amount) (UnionERC20Pool/libraries.sol#1686)
                - _deposit(_amount,_to) (UnionERC20Pool/libraries.sol#1643)
        - Transfer(address(0),account,amount) (UnionERC20Pool/libraries.sol#1435)
                - _deposit(_amount,_to) (UnionERC20Pool/libraries.sol#1643)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3
INFO:Detectors:
AddressUpgradeable.isContract(address) (UnionERC20Pool/libraries.sol#328-337) uses assembly
        - INLINE ASM (UnionERC20Pool/libraries.sol#335)
AddressUpgradeable._verifyCallResult(bool,bytes,string) (UnionERC20Pool/libraries.sol#449-466) uses assembly
        - INLINE ASM (UnionERC20Pool/libraries.sol#458-461)
Initializable._isConstructor() (UnionERC20Pool/libraries.sol#521-532) uses assembly
        - INLINE ASM (UnionERC20Pool/libraries.sol#530)
SignLib.splitSignature(bytes) (UnionERC20Pool/libraries.sol#1730-1751) uses assembly
        - INLINE ASM (UnionERC20Pool/libraries.sol#1741-1748)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
INFO:Detectors:
Different versions of Solidity is used in :
```

```
        - Version used: ['>=0.4.24<0.8.0', '>=0.6.0', '>=0.6.0<0.8.0', '>=0.6.12', '>=0.6.2<0.8.0', '>=0.6.6']
        - >=0.6.12 (UnionERC20Pool/UnionERC20Pool.sol#1)
        - >=0.6.0<0.8.0 (UnionERC20Pool/libraries.sol#5)
        - >=0.6.2<0.8.0 (UnionERC20Pool/libraries.sol#305)
        - >=0.4.24<0.8.0 (UnionERC20Pool/libraries.sol#474)
        - >=0.6.0<0.8.0 (UnionERC20Pool/libraries.sol#539)
        - >=0.6.0<0.8.0 (UnionERC20Pool/libraries.sol#574)
        - >=0.6.0<0.8.0 (UnionERC20Pool/libraries.sol#802)
        - >=0.6.0 (UnionERC20Pool/libraries.sol#899)
        - >=0.6.2<0.8.0 (UnionERC20Pool/libraries.sol#935)
        - >=0.6.0<0.8.0 (UnionERC20Pool/libraries.sol#956)
        - >=0.6.0<0.8.0 (UnionERC20Pool/libraries.sol#1118)
        - >=0.6.0<0.8.0 (UnionERC20Pool/libraries.sol#1198)
        - >=0.6.0<0.8.0 (UnionERC20Pool/libraries.sol#1513)
        - >=0.6.2<0.8.0 (UnionERC20Pool/libraries.sol#1590)
        - >=0.6.12 (UnionERC20Pool/libraries.sol#1601)
        - >=0.6.6 (UnionERC20Pool/libraries.sol#1712)
        - >=0.6.12 (UnionERC20Pool/libraries.sol#1756)
        - >=0.6.2<0.8.0 (UnionERC20Pool/libraries.sol#2039)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used
INFO:Detectors:
UnionERC20Pool._unloadMCRPendingQueue(uint256,uint256) (UnionERC20Pool/UnionERC20Pool.sol#270-283) has costly operations inside a loop:
        - totalMcrPending = totalMcrPending.sub(mcrPending) (UnionERC20Pool/UnionERC20Pool.sol#277)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#costly-operations-inside-a-loop
INFO:Detectors:
Pragma version>=0.6.12 (UnionERC20Pool/UnionERC20Pool.sol#1) necessitates a version too recent to be trusted. Consider deploying with 0.6.11
Pragma version>=0.6.0<0.8.0 (UnionERC20Pool/libraries.sol#5) is too complex
Pragma version>=0.6.2<0.8.0 (UnionERC20Pool/libraries.sol#305) is too complex
Pragma version>=0.4.24<0.8.0 (UnionERC20Pool/libraries.sol#474) is too complex
Pragma version>=0.6.0<0.8.0 (UnionERC20Pool/libraries.sol#539) is too complex
Pragma version>=0.6.0<0.8.0 (UnionERC20Pool/libraries.sol#574) is too complex
Pragma version>=0.6.0<0.8.0 (UnionERC20Pool/libraries.sol#802) is too complex
Pragma version>=0.6.0 (UnionERC20Pool/libraries.sol#899) allows old versions
Pragma version>=0.6.2<0.8.0 (UnionERC20Pool/libraries.sol#935) is too complex
Pragma version>=0.6.0<0.8.0 (UnionERC20Pool/libraries.sol#956) is too complex
Pragma version>=0.6.0<0.8.0 (UnionERC20Pool/libraries.sol#1118) is too complex
Pragma version>=0.6.0<0.8.0 (UnionERC20Pool/libraries.sol#1198) is too complex
Pragma version>=0.6.0<0.8.0 (UnionERC20Pool/libraries.sol#1513) is too complex
Pragma version>=0.6.2<0.8.0 (UnionERC20Pool/libraries.sol#1590) is too complex
Pragma version>=0.6.12 (UnionERC20Pool/libraries.sol#1601) necessitates a version too recent to be trusted. Consider deploying with 0.6.11
Pragma version>=0.6.6 (UnionERC20Pool/libraries.sol#1712) allows old versions
Pragma version>=0.6.12 (UnionERC20Pool/libraries.sol#1756) necessitates a version too recent to be trusted. Consider deploying with 0.6.11
Pragma version>=0.6.2<0.8.0 (UnionERC20Pool/libraries.sol#2039) is too complex
solc-0.6.12 is not recommended for deployment
```

```
INFO:Detectors:
Low level call in AddressUpgradeable.sendValue(address,uint256) (UnionERC20Pool/libraries.sol#355-361):
        - (success) = recipient.call{value: amount}() (UnionERC20Pool/libraries.sol#359)
Low level call in AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (UnionERC20Pool/libraries.sol#416-423):
        - (success,returndata) = target.call{value: value}(data) (UnionERC20Pool/libraries.sol#421)
Low level call in AddressUpgradeable.functionStaticCall(address,bytes,string) (UnionERC20Pool/libraries.sol#441-447):
        - (success,returndata) = target.staticcall(data) (UnionERC20Pool/libraries.sol#445)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
INFO:Detectors:
Function UnionERC20Pool.__UnionERC20Pool_init(address,address,string) (UnionERC20Pool/UnionERC20Pool.sol#60-65) is not in mixedCase
Parameter UnionERC20Pool.__UnionERC20Pool_init(address,address,string)._basicToken (UnionERC20Pool/UnionERC20Pool.sol#60) is not in mixedCase
Parameter UnionERC20Pool.__UnionERC20Pool_init(address,address,string)._description (UnionERC20Pool/UnionERC20Pool.sol#60) is not in mixedCase
Function UnionERC20Pool.__UnionERC20Pool_init_unchained(address) (UnionERC20Pool/UnionERC20Pool.sol#67-91) is not in mixedCase
Parameter UnionERC20Pool.setPoolReserveAddress(address)._poolReserveAddress (UnionERC20Pool/UnionERC20Pool.sol#110) is not in mixedCase
Parameter UnionERC20Pool.setFoundationReserveAddress(address)._foundationReserveAddress (UnionERC20Pool/UnionERC20Pool.sol#115) is not in mixedCase
Parameter UnionERC20Pool.setPoolReservePremiumCommission(uint8,uint8)._nom (UnionERC20Pool/UnionERC20Pool.sol#120) is not in mixedCase
Parameter UnionERC20Pool.setPoolReservePremiumCommission(uint8,uint8)._denom (UnionERC20Pool/UnionERC20Pool.sol#120) is not in mixedCase
Parameter UnionERC20Pool.setFoundationReservePremiumCommission(uint8,uint8)._nom (UnionERC20Pool/UnionERC20Pool.sol#126) is not in mixedCase
Parameter UnionERC20Pool.setFoundationReservePremiumCommission(uint8,uint8)._denom (UnionERC20Pool/UnionERC20Pool.sol#126) is not in mixedCase
Parameter UnionERC20Pool.setPoolReserveExcessLiquidityCommission(uint8,uint8)._nom (UnionERC20Pool/UnionERC20Pool.sol#132) is not in mixedCase
Parameter UnionERC20Pool.setPoolReserveExcessLiquidityCommission(uint8,uint8)._denom (UnionERC20Pool/UnionERC20Pool.sol#132) is not in mixedCase
Parameter UnionERC20Pool.setFoundationReserveExcessLiquidityCommission(uint8,uint8)._nom (UnionERC20Pool/UnionERC20Pool.sol#138) is not in mixedCase
Parameter UnionERC20Pool.setFoundationReserveExcessLiquidityCommission(uint8,uint8)._denom (UnionERC20Pool/UnionERC20Pool.sol#138) is not in mixedCase
Parameter UnionERC20Pool.getWriterData(address)._writer (UnionERC20Pool/UnionERC20Pool.sol#224) is not in mixedCase
Variable UnionERC20Pool.__gap (UnionERC20Pool/UnionERC20Pool.sol#300) is not in mixedCase
Variable UnionERC20Pool.OPERATOR_ROLE (UnionERC20Pool/UnionERC20Pool.sol#26) is not in mixedCase
Variable UnionERC20Pool.MCR_PROVIDER (UnionERC20Pool/UnionERC20Pool.sol#27) is not in mixedCase
Function ContextUpgradeable.__Context_init() (UnionERC20Pool/libraries.sol#553-555) is not in mixedCase
Function ContextUpgradeable.__Context_init_unchained() (UnionERC20Pool/libraries.sol#557-558) is not in mixedCase
Variable ContextUpgradeable.__gap (UnionERC20Pool/libraries.sol#567) is not in mixedCase
Function AccessControlUpgradeable.__AccessControl_init() (UnionERC20Pool/libraries.sol#616-619) is not in mixedCase
Function AccessControlUpgradeable.__AccessControl_init_unchained() (UnionERC20Pool/libraries.sol#621-622) is not in mixedCase
Variable AccessControlUpgradeable.__gap (UnionERC20Pool/libraries.sol#795) is not in mixedCase
Function PausableUpgradeable.__Pausable_init() (UnionERC20Pool/libraries.sol#831-834) is not in mixedCase
Function PausableUpgradeable.__Pausable_init_unchained() (UnionERC20Pool/libraries.sol#836-838) is not in mixedCase
Variable PausableUpgradeable.__gap (UnionERC20Pool/libraries.sol#894) is not in mixedCase
Function ERC20Upgradeable.__ERC20_init(string,string) (UnionERC20Pool/libraries.sol#1250-1253) is not in mixedCase
Function ERC20Upgradeable.__ERC20_init_unchained(string,string) (UnionERC20Pool/libraries.sol#1255-1259) is not in mixedCase
Variable ERC20Upgradeable.__gap (UnionERC20Pool/libraries.sol#1506) is not in mixedCase
Function PoolUpgradable.__Pool_init(address,string) (UnionERC20Pool/libraries.sol#1623-1626) is not in mixedCase
Parameter PoolUpgradable.__Pool_init(address,string)._basicToken (UnionERC20Pool/libraries.sol#1623) is not in mixedCase
Parameter PoolUpgradable.__Pool_init(address,string)._description (UnionERC20Pool/libraries.sol#1623) is not in mixedCase
Function PoolUpgradable.__Pool_init_unchained(address) (UnionERC20Pool/libraries.sol#1628-1630) is not in mixedCase
Parameter PoolUpgradable.__Pool_init_unchained(address)._basicToken (UnionERC20Pool/libraries.sol#1628) is not in mixedCase
```

```
Parameter StructuredLinkedList.getAdjacent(StructuredLinkedList.List,uint256,bool)._node (UnionERC20Pool/libraries.sol#1842) is not in mixedCase
Parameter StructuredLinkedList.getAdjacent(StructuredLinkedList.List,uint256,bool)._direction (UnionERC20Pool/libraries.sol#1842) is not in mixedCase
Parameter StructuredLinkedList.getNextNode(StructuredLinkedList.List,uint256)._node (UnionERC20Pool/libraries.sol#1856) is not in mixedCase
Parameter StructuredLinkedList.getPreviousNode(StructuredLinkedList.List,uint256)._node (UnionERC20Pool/libraries.sol#1866) is not in mixedCase
Parameter StructuredLinkedList.getSortedSpot(StructuredLinkedList.List,address,uint256)._structure (UnionERC20Pool/libraries.sol#1879) is not in mixedCase
Parameter StructuredLinkedList.getSortedSpot(StructuredLinkedList.List,address,uint256)._value (UnionERC20Pool/libraries.sol#1879) is not in mixedCase
Parameter StructuredLinkedList.insertAfter(StructuredLinkedList.List,uint256,uint256)._node (UnionERC20Pool/libraries.sol#1905) is not in mixedCase
Parameter StructuredLinkedList.insertAfter(StructuredLinkedList.List,uint256,uint256)._new (UnionERC20Pool/libraries.sol#1905) is not in mixedCase
Parameter StructuredLinkedList.insertBefore(StructuredLinkedList.List,uint256,uint256)._node (UnionERC20Pool/libraries.sol#1916) is not in mixedCase
Parameter StructuredLinkedList.insertBefore(StructuredLinkedList.List,uint256,uint256)._new (UnionERC20Pool/libraries.sol#1916) is not in mixedCase
Parameter StructuredLinkedList.remove(StructuredLinkedList.List,uint256)._node (UnionERC20Pool/libraries.sol#1926) is not in mixedCase
Parameter StructuredLinkedList.pushFront(StructuredLinkedList.List,uint256)._node (UnionERC20Pool/libraries.sol#1945) is not in mixedCase
Parameter StructuredLinkedList.pushBack(StructuredLinkedList.List,uint256)._node (UnionERC20Pool/libraries.sol#1955) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
Redundant expression "this (UnionERC20Pool/libraries.sol#564)" inContextUpgradeable (UnionERC20Pool/libraries.sol#552-568)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements
INFO:Detectors:
UnionERC20Pool._updateMCR(uint256,uint256,uint256) (UnionERC20Pool/UnionERC20Pool.sol#248-268) uses literals with too many digits:
        - item = (mcrIncrement << 64).add(block.number & 0x0000000000000000000000000000000000000000FFFFFFFFFFFFFFFF) (UnionERC20Pool/UnionERC20Pool.sol#262)
UnionERC20Pool._unloadMCRPendingQueue(uint256,uint256) (UnionERC20Pool/UnionERC20Pool.sol#270-283) uses literals with too many digits:
        - blockNumber = uint64(item & 0x0000000000000000000000000000000000000000FFFFFFFFFFFFFFFF) (UnionERC20Pool/UnionERC20Pool.sol#273)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits
INFO:Detectors:
UnionERC20Pool (UnionERC20Pool/UnionERC20Pool.sol#13-302) does not implement functions:
        - IPool.unlockPremium(uint256[]) (UnionERC20Pool/libraries.sol#938)
        - IPool.version() (UnionERC20Pool/libraries.sol#942)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unimplemented-functions
INFO:Detectors:
UnionERC20Pool.__gap (UnionERC20Pool/UnionERC20Pool.sol#300) is never used in UnionERC20Pool (UnionERC20Pool/UnionERC20Pool.sol#13-302)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variables
INFO:Detectors:
UnionERC20Pool.excessLiquidityManagerAddress (UnionERC20Pool/UnionERC20Pool.sol#41) should be constant
UnionERC20Pool.lockedPremium (UnionERC20Pool/UnionERC20Pool.sol#49) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant
INFO:Detectors:
setPoolReserveAddress(address) should be declared external:
        - UnionERC20Pool.setPoolReserveAddress(address) (UnionERC20Pool/UnionERC20Pool.sol#110-113)
setFoundationReserveAddress(address) should be declared external:
        - UnionERC20Pool.setFoundationReserveAddress(address) (UnionERC20Pool/UnionERC20Pool.sol#115-118)
setPoolReservePremiumCommission(uint8,uint8) should be declared external:
        - UnionERC20Pool.setPoolReservePremiumCommission(uint8,uint8) (UnionERC20Pool/UnionERC20Pool.sol#120-124)
setFoundationReservePremiumCommission(uint8,uint8) should be declared external:
        - UnionERC20Pool.setFoundationReservePremiumCommission(uint8,uint8) (UnionERC20Pool/UnionERC20Pool.sol#126-130)
```

```
setPoolReserveAddress(address) should be declared external:
        - UnionERC20Pool.setPoolReserveAddress(address) (UnionERC20Pool/UnionERC20Pool.sol#110-113)
setFoundationReserveAddress(address) should be declared external:
        - UnionERC20Pool.setFoundationReserveAddress(address) (UnionERC20Pool/UnionERC20Pool.sol#115-118)
setPoolReservePremiumCommission(uint8,uint8) should be declared external:
        - UnionERC20Pool.setPoolReservePremiumCommission(uint8,uint8) (UnionERC20Pool/UnionERC20Pool.sol#120-124)
setFoundationReservePremiumCommission(uint8,uint8) should be declared external:
        - UnionERC20Pool.setFoundationReservePremiumCommission(uint8,uint8) (UnionERC20Pool/UnionERC20Pool.sol#126-130)
setPoolReserveExcessLiquidityCommission(uint8,uint8) should be declared external:
        - UnionERC20Pool.setPoolReserveExcessLiquidityCommission(uint8,uint8) (UnionERC20Pool/UnionERC20Pool.sol#132-136)
setFoundationReserveExcessLiquidityCommission(uint8,uint8) should be declared external:
        - UnionERC20Pool.setFoundationReserveExcessLiquidityCommission(uint8,uint8) (UnionERC20Pool/UnionERC20Pool.sol#138-142)
pause() should be declared external:
        - UnionERC20Pool.pause() (UnionERC20Pool/UnionERC20Pool.sol#147-149)
unpause() should be declared external:
        - UnionERC20Pool.unpause() (UnionERC20Pool/UnionERC20Pool.sol#153-155)
withdrawPoolReserveCommission(uint256) should be declared external:
        - UnionERC20Pool.withdrawPoolReserveCommission(uint256) (UnionERC20Pool/UnionERC20Pool.sol#158-163)
withdrawFoundationReserveCommission(uint256) should be declared external:
        - UnionERC20Pool.withdrawFoundationReserveCommission(uint256) (UnionERC20Pool/UnionERC20Pool.sol#165-170)
flushMCRPendingQueue(uint256,uint256[2],bytes) should be declared external:
        - UnionERC20Pool.flushMCRPendingQueue(uint256,uint256[2],bytes) (UnionERC20Pool/UnionERC20Pool.sol#190-205)
getBasicToken() should be declared external:
        - UnionERC20Pool.getBasicToken() (UnionERC20Pool/UnionERC20Pool.sol#207-209)
getBasicTokenDecimals() should be declared external:
        - UnionERC20Pool.getBasicTokenDecimals() (UnionERC20Pool/UnionERC20Pool.sol#212-214)
getWriterData(address) should be declared external:
        - UnionERC20Pool.getWriterData(address) (UnionERC20Pool/UnionERC20Pool.sol#224-226)
getTotalValueLocked() should be declared external:
        - UnionERC20Pool.getTotalValueLocked() (UnionERC20Pool/UnionERC20Pool.sol#234-236)
getPoolStat() should be declared external:
        - UnionERC20Pool.getPoolStat() (UnionERC20Pool/UnionERC20Pool.sol#238-246)
getRoleMemberCount(bytes32) should be declared external:
        - AccessControlUpgradeable.getRoleMemberCount(bytes32) (UnionERC20Pool/libraries.sol#673-675)
getRoleMember(bytes32,uint256) should be declared external:
        - AccessControlUpgradeable.getRoleMember(bytes32,uint256) (UnionERC20Pool/libraries.sol#689-691)
getRoleAdmin(bytes32) should be declared external:
        - AccessControlUpgradeable.getRoleAdmin(bytes32) (UnionERC20Pool/libraries.sol#699-701)
grantRole(bytes32,address) should be declared external:
        - AccessControlUpgradeable.grantRole(bytes32,address) (UnionERC20Pool/libraries.sol#713-717)
revokeRole(bytes32,address) should be declared external:
        - AccessControlUpgradeable.revokeRole(bytes32,address) (UnionERC20Pool/libraries.sol#728-732)
renounceRole(bytes32,address) should be declared external:
        - AccessControlUpgradeable.renounceRole(bytes32,address) (UnionERC20Pool/libraries.sol#748-752)
```

```
name() should be declared external:
        - ERC20Upgradeable.name() (UnionERC20Pool/libraries.sol#1264-1266)
symbol() should be declared external:
        - ERC20Upgradeable.symbol() (UnionERC20Pool/libraries.sol#1272-1274)
decimals() should be declared external:
        - ERC20Upgradeable.decimals() (UnionERC20Pool/libraries.sol#1289-1291)
transfer(address,uint256) should be declared external:
        - ERC20Upgradeable.transfer(address,uint256) (UnionERC20Pool/libraries.sol#1315-1318)
allowance(address,address) should be declared external:
        - ERC20Upgradeable.allowance(address,address) (UnionERC20Pool/libraries.sol#1323-1325)
approve(address,uint256) should be declared external:
        - ERC20Upgradeable.approve(address,uint256) (UnionERC20Pool/libraries.sol#1334-1337)
transferFrom(address,address,uint256) should be declared external:
        - ERC20Upgradeable.transferFrom(address,address,uint256) (UnionERC20Pool/libraries.sol#1352-1356)
increaseAllowance(address,uint256) should be declared external:
        - ERC20Upgradeable.increaseAllowance(address,uint256) (UnionERC20Pool/libraries.sol#1370-1373)
decreaseAllowance(address,uint256) should be declared external:
        - ERC20Upgradeable.decreaseAllowance(address,uint256) (UnionERC20Pool/libraries.sol#1389-1392)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

Pic 6. UpgradeableProxy Slither automated report:

```
INFO:Detectors:
UpgradeableProxy.constructor(address,bytes) (upgradable/UpgradeableProxy.sol#24-32) uses delegatecall to a input-controlled function id
        - (success) = _logic.delegatecall(_data) (upgradable/UpgradeableProxy.sol#29)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#controlled-delegatecall
INFO:Detectors:
Proxy._delegate(address) (upgradable/Proxy.sol#21-41) uses assembly
        - INLINE ASM (upgradable/Proxy.sol#23-40)
UpgradeableProxy._implementation() (upgradable/UpgradeableProxy.sol#49-55) uses assembly
        - INLINE ASM (upgradable/UpgradeableProxy.sol#52-54)
UpgradeableProxy._setImplementation(address) (upgradable/UpgradeableProxy.sol#70-79) uses assembly
        - INLINE ASM (upgradable/UpgradeableProxy.sol#76-78)
Address.isContract(address) (upgradable/util/Address.sol#26-35) uses assembly
        - INLINE ASM (upgradable/util/Address.sol#33)
Address._verifyCallResult(bool,bytes,string) (upgradable/util/Address.sol#171-188) uses assembly
        - INLINE ASM (upgradable/util/Address.sol#180-183)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
INFO:Detectors:
Different versions of Solidity is used in :
        - Version used: ['>=0.6.0<0.8.0', '>=0.6.2<0.8.0']
        - >=0.6.0<0.8.0 (upgradable/Proxy.sol#3)
        - >=0.6.0<0.8.0 (upgradable/UpgradeableProxy.sol#3)
        - >=0.6.2<0.8.0 (upgradable/util/Address.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used
INFO:Detectors:
Pragma version>=0.6.0<0.8.0 (upgradable/Proxy.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (upgradable/UpgradeableProxy.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (upgradable/util/Address.sol#3) is too complex
solc-0.6.12 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Low level call in UpgradeableProxy.constructor(address,bytes) (upgradable/UpgradeableProxy.sol#24-32):
        - (success) = _logic.delegatecall(_data) (upgradable/UpgradeableProxy.sol#29)
Low level call in Address.sendValue(address,uint256) (upgradable/util/Address.sol#53-59):
        - (success) = recipient.call{value: amount}() (upgradable/util/Address.sol#57)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (upgradable/util/Address.sol#114-121):
        - (success,returndata) = target.call{value: value}(data) (upgradable/util/Address.sol#119)
Low level call in Address.functionStaticCall(address,bytes,string) (upgradable/util/Address.sol#139-145):
        - (success,returndata) = target.staticcall(data) (upgradable/util/Address.sol#143)
Low level call in Address.functionDelegateCall(address,bytes,string) (upgradable/util/Address.sol#163-169):
        - (success,returndata) = target.delegatecall(data) (upgradable/util/Address.sol#167)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
INFO:Slither:upgradable/UpgradeableProxy.sol analyzed (3 contracts with 46 detectors), 16 result(s) found
```

Pic 7. UpgradeableBeacon Slither automated report:

```
INFO:Detectors:
Different versions of Solidity is used in :
        - Version used: ['>=0.6.0<0.8.0', '>=0.6.2<0.8.0', '^0.6.0']
        - >=0.6.0<0.8.0 (upgradable/IBeacon.sol#3)
        - >=0.6.0<0.8.0 (upgradable/UpgradeableBeacon.sol#3)
        - >=0.6.2<0.8.0 (upgradable/util/Address.sol#3)
        - ^0.6.0 (upgradable/util/Context.sol#3)
        - >=0.6.0<0.8.0 (upgradable/util/Ownable.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used
INFO:Detectors:
Pragma version>=0.6.0<0.8.0 (upgradable/IBeacon.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (upgradable/UpgradeableBeacon.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (upgradable/util/Address.sol#3) is too complex
Pragma version^0.6.0 (upgradable/util/Context.sol#3) allows old versions
Pragma version>=0.6.0<0.8.0 (upgradable/util/Ownable.sol#3) is too complex
solc-0.6.12 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Low level call in Address.sendValue(address,uint256) (upgradable/util/Address.sol#53-59):
        - (success) = recipient.call{value: amount}() (upgradable/util/Address.sol#57)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (upgradable/util/Address.sol#114-121):
        - (success,returndata) = target.call{value: value}(data) (upgradable/util/Address.sol#119)
Low level call in Address.functionStaticCall(address,bytes,string) (upgradable/util/Address.sol#139-145):
        - (success,returndata) = target.staticcall(data) (upgradable/util/Address.sol#143)
Low level call in Address.functionDelegateCall(address,bytes,string) (upgradable/util/Address.sol#163-169):
        - (success,returndata) = target.delegatecall(data) (upgradable/util/Address.sol#167)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
INFO:Detectors:
implementation() should be declared external:
        - UpgradeableBeacon.implementation() (upgradable/UpgradeableBeacon.sol#34-36)
upgradeTo(address) should be declared external:
        - UpgradeableBeacon.upgradeTo(address) (upgradable/UpgradeableBeacon.sol#48-51)
owner() should be declared external:
        - Ownable.owner() (upgradable/util/Ownable.sol#35-37)
renounceOwnership() should be declared external:
        - Ownable.renounceOwnership() (upgradable/util/Ownable.sol#54-57)
transferOwnership(address) should be declared external:
        - Ownable.transferOwnership(address) (upgradable/util/Ownable.sol#63-67)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:upgradable/UpgradeableBeacon.sol analyzed (5 contracts with 46 detectors), 18 result(s) found
```