Woonkly - NFT protocol

Smart Contract Security Audit

Prepared by: **Halborn**

Date of Engagement: **February 17th, 2022 - March 4th, 2022**

Visit: **Halborn.com**

DOCUMENT REVISION HISTORY 3
CONTACTS 3
1 EXECUTIVE OVERVIEW 4
1.1 INTRODUCTION 5
1.2 AUDIT SUMMARY 5
1.3 TEST APPROACH & METHODOLOGY 5
RISK METHODOLOGY 6
1.4 SCOPE 8
2 ASSESSMENT SUMMARY & FINDINGS OVERVIEW 9
3 FINDINGS & TECH DETAILS 10
3.1 (HAL-01) FUNCTION ERC721WOONKLYNFTREVEALWAVE.CHANGEHIDDENBASEURI MODIFIES THE WRONG STATE VARIABLE - LOW 12
Description 12
Risk Level 12
Recommendation 12
Remediation Plan 13
3.2 (HAL-02) REVEALWAVE.REVEALDATE IS NOT USED - INFORMATIONAL 14
Description 14
Risk Level 14
Recommendation 15
Remediation Plan 15
3.3 (HAL-03) UNNEEDED INITIALIZATION OF UINT256 VARIABLES TO 0 - INFORMATIONAL 16
Description 16
Code Location 16

Risk Level 17
Recommendation 17
Remediation Plan 17
3.4 (HAL-04) USING ++I CONSUMES LESS GAS THAN I++ IN LOOPS - INFORMATIONAL 18
Description 18
Code Location 18
Risk Level 19
Proof of Concept 19
Risk Level 20
Recommendation 20
Remediation Plan 20
3.5 (HAL-05) POSSIBLE MISUSE OF PUBLIC FUNCTIONS - INFORMATIONAL 21
Description 21
Risk Level 22
Recommendation 22
Remediation Plan 23
4 AUTOMATED TESTING 24
4.1 STATIC ANALYSIS REPORT 25
Description 25
Slither results 25
4.2 AUTOMATED SECURITY SCAN 59
Description 59
MythX results 59

DOCUMENT REVISION HISTORY

VERSION	MODIFICATION	DATE	AUTHOR
0.1	Document Creation	02/17/2022	Roberto Reigada
0.2	Document Updates	03/04/2022	Roberto Reigada
0.3	Draft Review	03/11/2022	Gabi Urrutia
1.0	Remediation Plan	03/22/2022	Roberto Reigada
1.1	Remediation Plan Review	03/22/2022	Gabi Urrutia

CONTACTS

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

EXECUTIVE OVERVIEW

1.1 INTRODUCTION

Woonkly engaged Halborn to conduct a security audit on their staking smart contracts beginning on February 17th, 2022 and ending on March 4th, 2022. The security assessment was scoped to the smart contracts provided in the GitHub repository Woonkly/nft-protocol-contracts.

1.2 AUDIT SUMMARY

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

The purpose of this audit is to:

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

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

1.3 TEST APPROACH & METHODOLOGY

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

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

RISK METHODOLOGY:

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

RISK SCALE - LIKELIHOOD

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

RISK SCALE - IMPACT

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

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

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

CRITICAL	HIGH	MEDIUM	LOW	INFORMATIONAL

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

1.4 SCOPE

IN-SCOPE:

The security assessment was scoped to the following smart contracts:

- ProxyAdmin.sol
- ExchangeV2.sol
- RoyaltiesRegistry.sol
- ERC721WoonklyNFT.sol
- ERC721WoonklyNFTMinimal.sol
- ERC721WoonklyNFTRevealWave.sol
- ERC721WoonklyNFTMinimalRevealWave.sol
- ERC721WoonklyNFTFactoryC2.sol
- ERC721WoonklyNFTRevealWaveFactoryC2.sol
- ERC721WoonklyNFTMinimalBeacon.sol
- ERC721WoonklyNFTBeacon.sol
- ERC1155WoonklyNFT.sol
- ERC1155WoonklyNFTRevealWave.sol
- ERC1155WoonklyNFTFactoryC2.sol
- ERC1155WoonklyNFTRevealWaveFactoryC2.sol
- ERC1155WoonklyNFTBeacon.sol
- ERC721LazyMintTransferProxy.sol
- ERC1155LazyMintTransferProxy.sol
- TransferProxy.sol
- ERC20TransferProxy.sol
- AssetMatcherCollection.sol

Commit ID: 5418fe74eafa63d8211b1689031ba2c4652af285

Fixed Commit ID: 494370ffa011530f1db0a84993a0f62eee77da4e

2. ASSESSMENT SUMMARY & FINDINGS OVERVIEW

CRITICAL	HIGH	MEDIUM	LOW	INFORMATIONAL
0	0	0	1	4

LIKELIHOOD

IMPACT

	(HAL-01)			
(HAL-02) (HAL-03) (HAL-04) (HAL-05)				

SECURITY ANALYSIS	RISK LEVEL	REMEDIATION DATE
HAL01 - FUNCTION ERC721WOONKLYNFTREVEALWAVE.CHANGEHIDD MODIFIES THE WRONG STATE VARIABLE	Low	SOLVED - 03/22/2022
HAL02 - REVEALWAVE.REVEALDATE IS NOT USED	Informational	SOLVED - 03/22/2022
HAL03 - UNNEEDED INITIALIZATION OF UINT256 VARIABLES TO 0	Informational	SOLVED - 03/22/2022
HAL04 - USING ++I CONSUMES LESS GAS THAN I++ IN LOOPS	Informational	SOLVED - 03/22/2022
HAL05 - POSSIBLE MISUSE OF PUBLIC FUNCTIONS	Informational	ACKNOWLEDGED

FINDINGS & TECH DETAILS

3.1 (HAL-01) FUNCTION ERC721WOONKLYNFTREVEALWAVE.CHANGEHIDDENBAS MODIFIES THE WRONG STATE VARIABLE - LOW

Description:

The contracts ERC721WoonklyNFTRevealWave, ERC1155WoonklyNFTRevealWave and ERC721WoonklyNFTMinimalRevealWave contain the function changeHiddenBaseURI():

Listing 1: ERC721WoonklyNFTRevealWave.sol (Line 125)

```
119 function changeHiddenBaseURI(
120     uint _waveId,
121     string memory _baseURI
122 ) public onlyOwner {
123     require(bytes(_baseURI).length > 0,"Error: Input parameters
 ↳ can not be empty (string)");
124     RevealWave storage revealWave = waves[_waveId];
125     revealWave.revealBaseURI = _baseURI;
126 }
```

As we can see in the code above, the function is modifying the revealBaseURI variable instead of the hiddenBaseURI. As the name of the function indicates, this is not correct.

Risk Level:

Likelihood - 2
Impact - 3

Recommendation:

It is recommended to modify the changeHiddenBaseURI() function, so it updates the hiddenBaseURI variable instead of the revealBaseURI in the

contracts ERC721WoonklyNFTRevealWave, ERC1155WoonklyNFTRevealWave and ERC721WoonklyNFTMinimalRevealWave.

Remediation Plan:

SOLVED: The Woonkly team modified the changeHiddenBaseURI() function as suggested.

3.2 (HAL-02) REVEALWAVE.REVEALDATE IS NOT USED - INFORMATIONAL

Description:

The contracts ERC721WoonklyNFTRevealWave, ERC1155WoonklyNFTRevealWave and ‘ERC721WoonklyNFTMinimalRevealWave‘‘ contain the following struct:

Listing 2: ERC721WoonklyNFTRevealWave.sol (Line 21)

```
15     struct RevealWave {
16         bool isRevealed;
17         string name;
18         string hiddenBaseURI;
19         bool addTokenURIToHiddenBaseURI;
20         string revealBaseURI;
21         uint revealDate;
22     }
```

The revealDate parameter is not used anywhere in the code and provides no utility, for this reason, it can be removed from the struct. There is only a setter function that can also be removed:

Listing 3: ERC721WoonklyNFTRevealWave.sol (Line 152)

```
147 function changeRevealDate(
148     uint _waveId,
149     uint _revealDate
150 ) public onlyOwner {
151     RevealWave storage revealWave = waves[_waveId];
152     revealWave.revealDate = _revealDate;
153 }
```

Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

It is recommended to remove the revealDate variable from the RevealWave struct.
It is also recommended to remove the setter function changeRevealDate() in the contracts ERC721WoonklyNFTRevealWave, ERC1155WoonklyNFTRevealWave and ERC721WoonklyNFTMinimalRevealWave.

Remediation Plan:

SOLVED: The Woonkly team removed the revealDate variable from the RevealWave struct. The setter function changeRevealDate() was also removed.

3.3 (HAL-03) UNNEEDED INITIALIZATION OF UINT256 VARIABLES TO 0 - INFORMATIONAL

Description:

As i is an uint256, it is already initialized to 0. uint256 i = 0 reassigns the 0 to i which wastes gas.

Code Location:

ERC721LazyMinimal.sol
- Line 59: for (uint i = 0; i < data.creators.length; i++){`` - Line 85:for (uint i = 0; i < _creators.length; i++) {‘

WoonklyNFTTransferManager.sol
- Line 165: for (uint256 i = 0; i < fees.length; i++){`` - Line 184:for (uint256 i = 0; i < payouts.length - 1; i++) {- Line 206:for (uint256 i = 0; i < orderOriginFees.length; i++) {‘

ERC1155Lazy.sol
- Line 70: for (uint i = 0; i < data.creators.length; i++){
- Line 117: for (uint i = 0; i < _creators.length; i++){

ERC1155Base.sol
- Line 31: for (uint i = 0; i < ids.length; i++){

ERC1155Upgradeable.sol
- Line 119: for (uint256 i = 0; i < accounts.length; ++i){
- Line 200: for (uint256 i = 0; i < ids.length; ++i){
- Line 280: for (uint i = 0; i < ids.length; i++){
- Line 327: for (uint i = 0; i < ids.length; i++){

RoyaltiesRegistry.sol
- Line 99: for (uint i = 0; i < royalties.length; i++){
- Line 225: for (uint256 i = 0; i < values.length; i++){

ERC721Lazy.sol
- Line 63: `for (uint i = 0; i < data.creators.length; i++){`
- Line 99: `for (uint i = 0; i < _creators.length; i++){`

ERC721WoonklyNFTRevealWave.sol
- Line 47: `for (uint256 i = 0; i < operators.length; i++){`

Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

It is recommended to not initialize uint256 variables to 0 to save some gas. For example, use instead:
`for (uint256 i; i < operators.length; ++i){`.

Remediation Plan:

SOLVED: The Woonkly team removed the initialization of uint256 variable to 0 in the for loops mentioned reducing the gas costs.

3.4 (HAL-04) USING ++I CONSUMES LESS GAS THAN I++ IN LOOPS - INFORMATIONAL

Description:

In the loop below, the variable i is incremented using i++. It is known that, in loops, using ++i costs less gas per iteration than i++.

Code Location:

ERC721LazyMinimal.sol
- Line 59: for (uint i = 0; i < data.creators.length; i++){`` - Line 85:for (uint i = 0; i < _creators.length; i++) {‘

WoonklyNFTTransferManager.sol
- Line 165: for (uint256 i = 0; i < fees.length; i++){`` - Line 184:for (uint256 i = 0; i < payouts.length - 1; i++) {- Line 206:for (uint256 i = 0; i < orderOriginFees.length; i++) {‘

ERC1155Lazy.sol
- Line 70: for (uint i = 0; i < data.creators.length; i++){
- Line 117: for (uint i = 0; i < _creators.length; i++){

ERC1155Base.sol
- Line 31: for (uint i = 0; i < ids.length; i++){

ERC1155Upgradeable.sol
- Line 280: for (uint i = 0; i < ids.length; i++){
- Line 327: for (uint i = 0; i < ids.length; i++){

RoyaltiesRegistry.sol
- Line 99: for (uint i = 0; i < royalties.length; i++){
- Line 225: for (uint256 i = 0; i < values.length; i++){

ERC721Lazy.sol
- Line 63: `for (uint i = 0; i < data.creators.length; i++){`
- Line 99: `for (uint i = 0; i < _creators.length; i++){`

ERC721WoonklyNFTRevealWave.sol
- Line 47: `for (uint256 i = 0; i < operators.length; i++){`

Risk Level:

Likelihood - 1
Impact - 1

Proof of Concept:

For example, based in the following test contract:

Listing 4: Test.sol

```
//SPDX-License-Identifier: MIT
pragma solidity 0.8.9;

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

We can see the difference in the gas costs:

```
>>> test_contract.postiincrement(1)
Transaction sent: 0x1ecede6b109b707786d3685bd71dd9f22dc389957653036ca04c4cd2e72c5e0b
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 44
  test.postiincrement confirmed   Block: 13622335   Gas used: 21620 (0.32%)

<Transaction '0x1ecede6b109b707786d3685bd71dd9f22dc389957653036ca04c4cd2e72c5e0b'>
>>> test_contract.preiincrement(1)
Transaction sent: 0x205f09a4d2268de4c1a40f35bb2ec2847bf2ab8d584909b42c71a022b047614a
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 45
  test.preiincrement confirmed   Block: 13622336   Gas used: 21593 (0.32%)

<Transaction '0x205f09a4d2268de4c1a40f35bb2ec2847bf2ab8d584909b42c71a022b047614a'>
>>> test_contract.postiincrement(10)
Transaction sent: 0x98c04430526a59ba1cf947c114b62666a4417165947d31bf300cd6ae68328033
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 46
  test.postiincrement confirmed   Block: 13622337   Gas used: 22673 (0.34%)

<Transaction '0x98c04430526a59ba1cf947c114b62666a4417165947d31bf300cd6ae68328033'>
>>> test_contract.preiincrement(10)
Transaction sent: 0xf060d04714eff8482a828342414d5a20be9958c822d42860e7992aba20e1de05
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 47
  test.preiincrement confirmed   Block: 13622338   Gas used: 22601 (0.34%)

<Transaction '0xf060d04714eff8482a828342414d5a20be9958c822d42860e7992aba20e1de05'>
```

Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

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

Remediation Plan:

SOLVED: The Woonkly team uses now `++i` instead of `i++` to increment the iterator variable in for loops reducing the gas costs.

3.5 (HAL-05) POSSIBLE MISUSE OF PUBLIC FUNCTIONS - INFORMATIONAL

Description:

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

ProxyAdmin.sol
- getProxyImplementation() (ProxyAdmin.sol#19-31)
- getProxyAdmin() (ProxyAdmin.sol#37-49)
- changeProxyAdmin() (ProxyAdmin.sol#56-61)
- upgrade() (ProxyAdmin.sol#68-73)
- upgradeAndCall() (ProxyAdmin.sol#84-90)

ERC721WoonklyNFTRevealWave.sol
- setRevealWave() (ERC721WoonklyNFTRevealWave.sol#104-117)
- changeHiddenBaseURI() (ERC721WoonklyNFTRevealWave.sol#119-126)
- changeAddTokenURIToHiddenBaseURI() (ERC721WoonklyNFTRevealWave.sol#128-134)
- changeRevealBaseURI() (ERC721WoonklyNFTRevealWave.sol#136-145)
- changeRevealDate() (ERC721WoonklyNFTRevealWave.sol#147-153)
- resetIsRevealedAndRevealURI() (ERC721WoonklyNFTRevealWave.sol#155-161)
- changeName() (ERC721WoonklyNFTRevealWave.sol#164-171)
- mintAndTransferReveal() (ERC721WoonklyNFTRevealWave.sol#233-243)

ERC721WoonklyNFTMinimalRevealWave.sol
- setRevealWave() (ERC721WoonklyNFTMinimalRevealWave.sol#70-83)
- changeHiddenBaseURI() (ERC721WoonklyNFTMinimalRevealWave.sol#85-92)
- changeAddTokenURIToHiddenBaseURI() (ERC721WoonklyNFTMinimalRevealWave.sol#94-100)
- changeRevealBaseURI() (ERC721WoonklyNFTMinimalRevealWave.sol#102-111)
- changeRevealDate() (ERC721WoonklyNFTMinimalRevealWave.sol#113-119)
- resetIsRevealedAndRevealURI() (ERC721WoonklyNFTMinimalRevealWave.sol#121-127)
- changeName() (ERC721WoonklyNFTMinimalRevealWave.sol#130-137)
- mintAndTransferReveal() (ERC721WoonklyNFTMinimalRevealWave.sol#199-209)

ERC721WoonklyNFTFactoryC2.sol

- getAddress() (ERC721WoonklyNFTFactoryC2.sol#61-73)
- getAddress() (ERC721WoonklyNFTFactoryC2.sol#80-92)

ERC721WoonklyNFTRevealWaveFactoryC2.sol
- getAddress() (ERC721WoonklyNFTRevealWaveFactoryC2.sol#64-76)
- getAddress() (ERC721WoonklyNFTRevealWaveFactoryC2.sol#83-95)

ERC1155WoonklyNFTRevealWave.sol
- setRevealWave() (ERC1155WoonklyNFTRevealWave.sol#72-85)
- changeHiddenBaseURI() (ERC1155WoonklyNFTRevealWave.sol#87-94)
- changeAddTokenURIToHiddenBaseURI() (ERC1155WoonklyNFTRevealWave.sol#96-102)
- changeRevealBaseURI() (ERC1155WoonklyNFTRevealWave.sol#104-113)
- changeRevealDate() (ERC1155WoonklyNFTRevealWave.sol#115-121)
- resetIsRevealedAndRevealURI() (ERC1155WoonklyNFTRevealWave.sol#123-129)
- changeName() (ERC1155WoonklyNFTRevealWave.sol#132-139)
- assignRevealWaveIdToTokenId() (ERC1155WoonklyNFTRevealWave.sol#142-145)

ERC1155WoonklyNFTFactoryC2.sol
- getAddress() (ERC1155WoonklyNFTFactoryC2.sol#63-75)
- getAddress() (ERC1155WoonklyNFTFactoryC2.sol#82-94)

ERC1155WoonklyNFTRevealWaveFactoryC2.sol
- getAddress() (ERC1155WoonklyNFTRevealWaveFactoryC2.sol#63-75)
- getAddress() (ERC1155WoonklyNFTRevealWaveFactoryC2.sol#82-94)

Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

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

Remediation Plan:

ACKNOWLEDGED: The Woonkly team acknowledges this issue.

AUTOMATED TESTING

4.1 STATIC ANALYSIS REPORT

Description:

Halborn used automated testing techniques to enhance the coverage of certain areas of the scoped contracts. Among the tools used was Slither, a Solidity static analysis framework. After Halborn verified all the contracts in the repository and was able to compile them correctly into their ABI and binary formats, Slither was run on the all-scoped contracts. This tool can statically verify mathematical relationships between Solidity variables to detect invalid or inconsistent usage of the contracts' APIs across the entire code-base.

Slither results:

ProxyAdmin.sol

```
AdminUpgradeabilityProxy.constructor(address,address,bytes)._admin (contracts/deploy-proxy-admin/AdminUpgradeabilityProxy.sol#25) shadows:
        - AdminUpgradeabilityProxy._admin() (contracts/deploy-proxy-admin/AdminUpgradeabilityProxy.sol#110-115) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

UpgradeabilityProxy.constructor(address,bytes)._logic (contracts/deploy-proxy-admin/UpgradeabilityProxy.sol#23) lacks a zero-check on :
                - (success) = _logic.delegatecall(_data) (contracts/deploy-proxy-admin/UpgradeabilityProxy.sol#27)
AdminUpgradeabilityProxy.upgradeToAndCall(address,bytes).newImplementation (contracts/deploy-proxy-admin/AdminUpgradeabilityProxy.sol#101) lacks a zero-check on :
                - (success) = newImplementation.delegatecall(data) (contracts/deploy-proxy-admin/AdminUpgradeabilityProxy.sol#103)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation

Modifier AdminUpgradeabilityProxy.ifAdmin() (contracts/deploy-proxy-admin/AdminUpgradeabilityProxy.sol#50-56) does not always execute _; or revertReference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-modifier

Address.isContract(address) (contracts/deploy-proxy-admin/Address.sol#26-35) uses assembly
        - INLINE ASM (contracts/deploy-proxy-admin/Address.sol#33)
Address._verifyCallResult(bool,bytes,string) (contracts/deploy-proxy-admin/Address.sol#171-188) uses assembly
        - INLINE ASM (contracts/deploy-proxy-admin/Address.sol#180-183)
AdminUpgradeabilityProxy._admin() (contracts/deploy-proxy-admin/AdminUpgradeabilityProxy.sol#110-115) uses assembly
        - INLINE ASM (contracts/deploy-proxy-admin/AdminUpgradeabilityProxy.sol#112-114)
AdminUpgradeabilityProxy._setAdmin(address) (contracts/deploy-proxy-admin/AdminUpgradeabilityProxy.sol#121-127) uses assembly
        - INLINE ASM (contracts/deploy-proxy-admin/AdminUpgradeabilityProxy.sol#124-126)
Proxy._delegate(address) (contracts/deploy-proxy-admin/Proxy.sol#40-59) uses assembly
        - INLINE ASM (contracts/deploy-proxy-admin/Proxy.sol#41-58)
UpgradeabilityProxy._implementation() (contracts/deploy-proxy-admin/UpgradeabilityProxy.sol#49-54) uses assembly
        - INLINE ASM (contracts/deploy-proxy-admin/UpgradeabilityProxy.sol#51-53)
UpgradeabilityProxy._setImplementation(address) (contracts/deploy-proxy-admin/UpgradeabilityProxy.sol#69-77) uses assembly
        - INLINE ASM (contracts/deploy-proxy-admin/UpgradeabilityProxy.sol#74-76)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Different versions of Solidity is used:
        - Version used: ['>=0.6.0<0.8.0', '>=0.6.2<0.8.0', '^0.6.0']
        - >=0.6.2<0.8.0 (contracts/deploy-proxy-admin/Address.sol#3)
        - ^0.6.0 (contracts/deploy-proxy-admin/AdminUpgradeabilityProxy.sol#3)
        - >=0.6.0<0.8.0 (contracts/deploy-proxy-admin/Context.sol#3)
        - >=0.6.0<0.8.0 (contracts/deploy-proxy-admin/Ownable.sol#3)
        - ^0.6.0 (contracts/deploy-proxy-admin/Proxy.sol#3)
        - ^0.6.0 (contracts/deploy-proxy-admin/ProxyAdmin.sol#3)
        - ^0.6.0 (contracts/deploy-proxy-admin/UpgradeabilityProxy.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

Address._verifyCallResult(bool,bytes,string) (contracts/deploy-proxy-admin/Address.sol#171-188) is never used and should be removed
Address.functionCall(address,bytes) (contracts/deploy-proxy-admin/Address.sol#79-81) is never used and should be removed
Address.functionCall(address,bytes,string) (contracts/deploy-proxy-admin/Address.sol#89-91) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (contracts/deploy-proxy-admin/Address.sol#104-106) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (contracts/deploy-proxy-admin/Address.sol#114-121) is never used and should be removed
Address.functionDelegateCall(address,bytes) (contracts/deploy-proxy-admin/Address.sol#153-155) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (contracts/deploy-proxy-admin/Address.sol#163-169) is never used and should be removed
Address.functionStaticCall(address,bytes) (contracts/deploy-proxy-admin/Address.sol#129-131) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (contracts/deploy-proxy-admin/Address.sol#139-145) is never used and should be removed
Address.sendValue(address,uint256) (contracts/deploy-proxy-admin/Address.sol#53-59) is never used and should be removed
Context._msgData() (contracts/deploy-proxy-admin/Context.sol#20-23) is never used and should be removed
Proxy._implementation() (contracts/deploy-proxy-admin/Proxy.sol#32) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version>=0.6.2<0.8.0 (contracts/deploy-proxy-admin/Address.sol#3) is too complex
Pragma version^0.6.0 (contracts/deploy-proxy-admin/AdminUpgradeabilityProxy.sol#3) allows old versions
Pragma version>=0.6.0<0.8.0 (contracts/deploy-proxy-admin/Context.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (contracts/deploy-proxy-admin/Ownable.sol#3) is too complex
Pragma version^0.6.0 (contracts/deploy-proxy-admin/Proxy.sol#3) allows old versions
Pragma version^0.6.0 (contracts/deploy-proxy-admin/ProxyAdmin.sol#3) allows old versions
Pragma version^0.6.0 (contracts/deploy-proxy-admin/UpgradeabilityProxy.sol#3) allows old versions
solc-0.6.6 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (contracts/deploy-proxy-admin/Address.sol#53-59):
        - (success) = recipient.call{value: amount}() (contracts/deploy-proxy-admin/Address.sol#57)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (contracts/deploy-proxy-admin/Address.sol#114-121):
        - (success,returndata) = target.call{value: value}(data) (contracts/deploy-proxy-admin/Address.sol#119)
Low level call in Address.functionStaticCall(address,bytes,string) (contracts/deploy-proxy-admin/Address.sol#139-145):
        - (success,returndata) = target.staticcall(data) (contracts/deploy-proxy-admin/Address.sol#143)
Low level call in Address.functionDelegateCall(address,bytes,string) (contracts/deploy-proxy-admin/Address.sol#163-169):
        - (success,returndata) = target.delegatecall(data) (contracts/deploy-proxy-admin/Address.sol#167)
Low level call in AdminUpgradeabilityProxy.upgradeToAndCall(address,bytes) (contracts/deploy-proxy-admin/AdminUpgradeabilityProxy.sol#101-105):
        - (success) = newImplementation.delegatecall(data) (contracts/deploy-proxy-admin/AdminUpgradeabilityProxy.sol#103)
Low level call in ProxyAdmin.getProxyImplementation(AdminUpgradeabilityProxy) (contracts/deploy-proxy-admin/ProxyAdmin.sol#19-31):
        - (success,returndata) = address(proxy).staticcall(0x5c60da1b) (contracts/deploy-proxy-admin/ProxyAdmin.sol#26-28)
Low level call in ProxyAdmin.getProxyAdmin(AdminUpgradeabilityProxy) (contracts/deploy-proxy-admin/ProxyAdmin.sol#37-49):
        - (success,returndata) = address(proxy).staticcall(0xf851a440) (contracts/deploy-proxy-admin/ProxyAdmin.sol#44-46)
Low level call in UpgradeabilityProxy.constructor(address,bytes) (contracts/deploy-proxy-admin/UpgradeabilityProxy.sol#23-30):
        - (success) = _logic.delegatecall(_data) (contracts/deploy-proxy-admin/UpgradeabilityProxy.sol#27)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Redundant expression "this (contracts/deploy-proxy-admin/Context.sol#21)" inContext (contracts/deploy-proxy-admin/Context.sol#15-24)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

renounceOwnership() should be declared external:
        - Ownable.renounceOwnership() (contracts/deploy-proxy-admin/Ownable.sol#54-57)
transferOwnership(address) should be declared external:
        - Ownable.transferOwnership(address) (contracts/deploy-proxy-admin/Ownable.sol#63-67)
getProxyImplementation(AdminUpgradeabilityProxy) should be declared external:
        - ProxyAdmin.getProxyImplementation(AdminUpgradeabilityProxy) (contracts/deploy-proxy-admin/ProxyAdmin.sol#19-31)
getProxyAdmin(AdminUpgradeabilityProxy) should be declared external:
        - ProxyAdmin.getProxyAdmin(AdminUpgradeabilityProxy) (contracts/deploy-proxy-admin/ProxyAdmin.sol#37-49)
changeProxyAdmin(AdminUpgradeabilityProxy,address) should be declared external:
        - ProxyAdmin.changeProxyAdmin(AdminUpgradeabilityProxy,address) (contracts/deploy-proxy-admin/ProxyAdmin.sol#56-61)
upgrade(AdminUpgradeabilityProxy,address) should be declared external:
        - ProxyAdmin.upgrade(AdminUpgradeabilityProxy,address) (contracts/deploy-proxy-admin/ProxyAdmin.sol#68-73)
upgradeAndCall(AdminUpgradeabilityProxy,address,bytes) should be declared external:
        - ProxyAdmin.upgradeAndCall(AdminUpgradeabilityProxy,address,bytes) (contracts/deploy-proxy-admin/ProxyAdmin.sol#84-90)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

ExchangeV2.sol

```
OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#74) shadows:
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
AssetMatcher.__gap (contracts/exchange-v2/AssetMatcher.sol#77) shadows:
        - OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#74)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
ExchangeV2Core.__gap (contracts/exchange-v2/ExchangeV2Core.sol#130) shadows:
        - OrderValidator.__gap (contracts/exchange-v2/OrderValidator.sol#50)
        - EIP712Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#120)
        - TransferExecutor.__gap (contracts/exchange-v2/TransferExecutor.sol#57)
        - AssetMatcher.__gap (contracts/exchange-v2/AssetMatcher.sol#77)
        - OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#74)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
OrderValidator.__gap (contracts/exchange-v2/OrderValidator.sol#50) shadows:
        - EIP712Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#120)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
RaribleTransferManager.__gap (contracts/exchange-v2/RaribleTransferManager.sol#226) shadows:
        - OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#74)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
TransferExecutor.__gap (contracts/exchange-v2/TransferExecutor.sol#57) shadows:
        - OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#74)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variable-shadowing

RaribleTransferManager.getRoyaltiesByAssetType(LibAsset.AssetType).data_scope_2 (contracts/exchange-v2/RaribleTransferManager.sol#147) is a storage variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-storage-variables

TransferExecutor.transfer(LibAsset.Asset,address,address,bytes4,bytes4).token_scope_1 (contracts/exchange-v2/TransferExecutor.sol#49) is a local variable never initialized
TransferExecutor.transfer(LibAsset.Asset,address,address,bytes4,bytes4).token_scope_0 (contracts/exchange-v2/TransferExecutor.sol#45) is a local variable never initialized
RaribleTransferManager.getRoyaltiesByAssetType(LibAsset.AssetType).empty (contracts/exchange-v2/RaribleTransferManager.sol#150) is a local variable never initialized
OrderValidator.validate(LibOrder.Order,bytes).signer (contracts/exchange-v2/OrderValidator.sol#32) is a local variable never initialized
TransferExecutor.transfer(LibAsset.Asset,address,address,bytes4,bytes4).tokenId_scope_2 (contracts/exchange-v2/TransferExecutor.sol#49) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

RaribleTransferManager.setDefaultFeeReceiver(address).newDefaultFeeReceiver (contracts/exchange-v2/RaribleTransferManager.sol#46) lacks a zero-check on :
            - defaultFeeReceiver = newDefaultFeeReceiver (contracts/exchange-v2/RaribleTransferManager.sol#47)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation

Variable 'RaribleTransferManager.getRoyaltiesByAssetType(LibAsset.AssetType).token (contracts/exchange-v2/RaribleTransferManager.sol#141)' in RaribleTransferManager.getRoyaltiesByAssetType(LibAsset.AssetType) (contracts/exchange-v2/Rarib
leTransferManager.sol#139-152) potentially used before declaration: (token,data) = abi.decode(matchNft.data,(address,LibERC1155LazyMint.Mint1155Data)) (contracts/exchange-v2/RaribleTransferManager.sol#144)
Variable 'RaribleTransferManager.getRoyaltiesByAssetType(LibAsset.AssetType).token (contracts/exchange-v2/RaribleTransferManager.sol#141)' in RaribleTransferManager.getRoyaltiesByAssetType(LibAsset.AssetType) (contracts/exchange-v2/Rarib
leTransferManager.sol#139-152) potentially used before declaration: (token,data) = abi.decode(matchNft.data,(address,LibERC721LazyMint.Mint721Data)) (contracts/exchange-v2/RaribleTransferManager.sol#147)
Variable 'RaribleTransferManager.getRoyaltiesByAssetType(LibAsset.AssetType).data (contracts/exchange-v2/RaribleTransferManager.sol#144)' in RaribleTransferManager.getRoyaltiesByAssetType(LibAsset.AssetType) (contracts/exchange-v2/Rarib
eTransferManager.sol#139-152) potentially used before declaration: (token,data) = abi.decode(matchNft.data,(address,LibERC721LazyMint.Mint721Data)) (contracts/exchange-v2/RaribleTransferManager.sol#147)
Variable 'TransferExecutor.transfer(LibAsset.Asset,address,address,bytes4,bytes4).token (contracts/exchange-v2/TransferExecutor.sol#42)' in TransferExecutor.transfer(LibAsset.Asset,address,address,bytes4,bytes4) (contracts/exchange-v2/Tr
ansferExecutor.sol#32-55) potentially used before declaration: (token,tokenId) = abi.decode(asset.assetType.data,(address,uint256)) (contracts/exchange-v2/TransferExecutor.sol#45)
Variable 'TransferExecutor.transfer(LibAsset.Asset,address,address,bytes4,bytes4).token (contracts/exchange-v2/TransferExecutor.sol#42)' in TransferExecutor.transfer(LibAsset.Asset,address,address,bytes4,bytes4) (contracts/exchange-v2/Tr
ansferExecutor.sol#32-55) potentially used before declaration: (token,tokenId) = abi.decode(asset.assetType.data,(address,uint256)) (contracts/exchange-v2/TransferExecutor.sol#49)
Variable 'TransferExecutor.transfer(LibAsset.Asset,address,address,bytes4,bytes4).tokenId (contracts/exchange-v2/TransferExecutor.sol#45)' in TransferExecutor.transfer(LibAsset.Asset,address,address,bytes4,bytes4) (contracts/exchange-v2/
TransferExecutor.sol#32-55) potentially used before declaration: (token,tokenId) = abi.decode(asset.assetType.data,(address,uint256)) (contracts/exchange-v2/TransferExecutor.sol#49)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables

Reentrancy in ExchangeV2Core.matchAndTransfer(LibOrder.Order,LibOrder.Order) (contracts/exchange-v2/ExchangeV2Core.sol#52-76):
        External calls:
        - address(msg.sender).transferEth(msg.value.sub(totalMakeValue)) (contracts/exchange-v2/ExchangeV2Core.sol#67)
        - address(msg.sender).transferEth(msg.value.sub(totalTakeValue)) (contracts/exchange-v2/ExchangeV2Core.sol#72)
        Event emitted after the call(s):
        - Match(leftOrderKeyHash,rightOrderKeyHash,orderLeft.maker,orderRight.maker,newFill.rightValue,newFill.leftValue,makeMatch,takeMatch) (contracts/exchange-v2/ExchangeV2Core.sol#75)
Reentrancy in ExchangeV2Core.matchOrders(LibOrder.Order,bytes,LibOrder.Order,bytes) (contracts/exchange-v2/ExchangeV2Core.sol#35-50):
        External calls:
        - validateFull(orderLeft,signatureLeft) (contracts/exchange-v2/ExchangeV2Core.sol#41)
            - order.maker.isContract() (contracts/exchange-v2/OrderValidator.sol#37)
        - validateFull(orderRight,signatureRight) (contracts/exchange-v2/ExchangeV2Core.sol#42)
            - order.maker.isContract() (contracts/exchange-v2/OrderValidator.sol#37)
        - matchAndTransfer(orderLeft,orderRight) (contracts/exchange-v2/ExchangeV2Core.sol#49)
            - (success) = to.call{value: value}() (contracts/exchange-v2/lib/LibTransfer.sol#7)
            - address(msg.sender).transferEth(msg.value.sub(totalMakeValue)) (contracts/exchange-v2/ExchangeV2Core.sol#67)
            - address(msg.sender).transferEth(msg.value.sub(totalTakeValue)) (contracts/exchange-v2/ExchangeV2Core.sol#72)
        External calls sending eth:
        - matchAndTransfer(orderLeft,orderRight) (contracts/exchange-v2/ExchangeV2Core.sol#49)
            - (success) = to.call{value: value}() (contracts/exchange-v2/lib/LibTransfer.sol#7)
        Event emitted after the call(s):
        - Match(leftOrderKeyHash,rightOrderKeyHash,orderLeft.maker,orderRight.maker,newFill.rightValue,newFill.leftValue,makeMatch,takeMatch) (contracts/exchange-v2/ExchangeV2Core.sol#75)
            - matchAndTransfer(orderLeft,orderRight) (contracts/exchange-v2/ExchangeV2Core.sol#49)
Reentrancy in TransferExecutor.transfer(LibAsset.Asset,address,address,bytes4,bytes4) (contracts/exchange-v2/TransferExecutor.sol#32-55):
        External calls:
        - to.transferEth(asset.value) (contracts/exchange-v2/TransferExecutor.sol#40)
        - IERC20TransferProxy(proxies[LibAsset.ERC20_ASSET_CLASS]).erc20safeTransferFrom(IERC20Upgradeable(token),from,to,asset.value) (contracts/exchange-v2/TransferExecutor.sol#43)
        - INftTransferProxy(proxies[LibAsset.ERC721_ASSET_CLASS]).erc721safeTransferFrom(IERC721Upgradeable(token_scope_0),from,to,tokenId) (contracts/exchange-v2/TransferExecutor.sol#47)
        - INftTransferProxy(proxies[LibAsset.ERC1155_ASSET_CLASS]).erc1155safeTransferFrom(IERC1155Upgradeable(token_scope_1),from,to,tokenId_scope_2,asset.value,) (contracts/exchange-v2/TransferExecutor.sol#50)
        - ITransferProxy(proxies[asset.assetType.assetClass]).transfer(asset,from,to) (contracts/exchange-v2/TransferExecutor.sol#52)
        Event emitted after the call(s):
        - Transfer(asset,from,to,transferDirection,transferType) (contracts/exchange-v2/TransferExecutor.sol#54)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

LibOrder.validate(LibOrder.Order) (contracts/exchange-v2/LibOrder.sol#75-78) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(order.start == 0 || order.start < block.timestamp,Order start validation failed) (contracts/exchange-v2/LibOrder.sol#76)
        - require(bool,string)(order.end == 0 || order.end > block.timestamp,Order end validation failed) (contracts/exchange-v2/LibOrder.sol#77)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

EIP712Upgradeable._getChainId() (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#93-99) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#96-98)
AddressUpgradeable.isContract(address) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#26-35) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#33)
AddressUpgradeable._verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#147-164) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#156-159)
LibSignature.recover(bytes32,bytes) (node_modules/@rarible/libraries/contracts/LibSignature.sol#20-45) uses assembly
        - INLINE ASM (node_modules/@rarible/libraries/contracts/LibSignature.sol#38-42)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Different versions of Solidity is used:
        - Version used: ['0.7.6', '>=0.4.24<0.8.0', '>=0.6.0<0.8.0', '>=0.6.2<0.8.0', '>=0.6.9<0.8.0', '^0.7.0']
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/introspection/IERC165Upgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/math/MathUpgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#3)
        - >=0.4.24<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/proxy/Initializable.sol#4)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/IERC1155Upgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/IERC20Upgradeable.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC721/IERC721Upgradeable.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#3)
        - 0.7.6 (node_modules/@rarible/exchange-interfaces/contracts/IAssetMatcher.sol#3)
        - v2 (node_modules/@rarible/exchange-interfaces/contracts/IAssetMatcher.sol#4)
        - >=0.6.9<0.8.0 (node_modules/@rarible/exchange-interfaces/contracts/IERC20TransferProxy.sol#3)
        - v2 (node_modules/@rarible/exchange-interfaces/contracts/IERC20TransferProxy.sol#4)
        - >=0.6.9<0.8.0 (node_modules/@rarible/exchange-interfaces/contracts/INftTransferProxy.sol#3)
        - v2 (node_modules/@rarible/exchange-interfaces/contracts/INftTransferProxy.sol#4)
        - >=0.6.9<0.8.0 (node_modules/@rarible/exchange-interfaces/contracts/ITransferProxy.sol#3)
        - v2 (node_modules/@rarible/exchange-interfaces/contracts/ITransferProxy.sol#4)
        - >=0.6.2<0.8.0 (node_modules/@rarible/lazy-mint/contracts/erc-1155/LibERC1155LazyMint.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@rarible/lazy-mint/contracts/erc-721/LibERC721LazyMint.sol#3)
        - 0.7.6 (node_modules/@rarible/lib-asset/contracts/LibAsset.sol#3)
        - ^0.7.0 (node_modules/@rarible/libraries/contracts/LibSignature.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/IRoyaltiesProvider.sol#3)
        - v2 (node_modules/@rarible/royalties/contracts/IRoyaltiesProvider.sol#4)
        - >=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/LibPart.sol#3)
        - 0.7.6 (contracts/exchange-v2/AssetMatcher.sol#3)
        - v2 (contracts/exchange-v2/AssetMatcher.sol#4)
        - 0.7.6 (contracts/exchange-v2/ExchangeV2.sol#3)
        - v2 (contracts/exchange-v2/ExchangeV2.sol#4)
        - 0.7.6 (contracts/exchange-v2/ExchangeV2Core.sol#3)
        - v2 (contracts/exchange-v2/ExchangeV2Core.sol#4)
        - 0.7.6 (contracts/exchange-v2/ITransferExecutor.sol#3)
        - v2 (contracts/exchange-v2/ITransferExecutor.sol#4)
        - 0.7.6 (contracts/exchange-v2/ITransferManager.sol#3)
        - v2 (contracts/exchange-v2/ITransferManager.sol#4)
        - 0.7.6 (contracts/exchange-v2/LibFeeSide.sol#3)
        - 0.7.6 (contracts/exchange-v2/LibFill.sol#3)
        - 0.7.6 (contracts/exchange-v2/LibOrder.sol#3)
        - 0.7.6 (contracts/exchange-v2/LibOrderData.sol#3)
        - 0.7.6 (contracts/exchange-v2/LibOrderDataV1.sol#3)
        - v2 (contracts/exchange-v2/LibOrderDataV1.sol#4)
        - 0.7.6 (contracts/exchange-v2/LibOrderDataV2.sol#3)
        - v2 (contracts/exchange-v2/LibOrderDataV2.sol#4)
        - 0.7.6 (contracts/exchange-v2/OrderValidator.sol#3)
        - 0.7.6 (contracts/exchange-v2/RaribleTransferManager.sol#3)
        - v2 (contracts/exchange-v2/RaribleTransferManager.sol#4)
        - 0.7.6 (contracts/exchange-v2/TransferExecutor.sol#3)
        - v2 (contracts/exchange-v2/TransferExecutor.sol#4)
        - 0.7.6 (contracts/exchange-v2/interfaces/IERC1271.sol#3)
        - 0.7.6 (contracts/exchange-v2/lib/BpLibrary.sol#3)
        - 0.7.6 (contracts/exchange-v2/lib/LibMath.sol#3)
        - 0.7.6 (contracts/exchange-v2/lib/LibTransfer.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

AddressUpgradeable._verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#147-164) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#79-81) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#89-91) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#104-106) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#114-121) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#129-131) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#139-145) is never used and should be removed
AddressUpgradeable.sendValue(address,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#53-59) is never used and should be removed
BpLibrary.bp(uint256,uint256) (contracts/exchange-v2/lib/BpLibrary.sol#10-13) is never used and should be removed
ContextUpgradeable.__Context_init() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#17-19) is never used and should be removed
ContextUpgradeable._msgData() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#27-30) is never used and should be removed
EIP712Upgradeable.__EIP712_init(string,string) (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#44-46) is never used and should be removed
```

AUTOMATED TESTING

```
ITransferManager.doTransfers(LibAsset.AssetType,LibAsset.AssetType,LibFill.FillResult,LibOrder.Order,LibOrder.Order,LibOrderDataV2.DataV2,LibOrderDataV2.DataV2) (contracts/exchange-v2/ITransferManager.sol#19-27) is never used and should
be removed
LibERC1155LazyMint.hash(LibERC1155LazyMint.Mint1155Data) (node_modules/@rarible/lazy-mint/contracts/erc-1155/LibERC1155LazyMint.sol#22-39) is never used and should be removed
LibERC721LazyMint.hash(LibERC721LazyMint.Mint721Data) (node_modules/@rarible/lazy-mint/contracts/erc-721/LibERC721LazyMint.sol#21-37) is never used and should be removed
LibMath.isRoundingErrorCeil(uint256,uint256,uint256) (contracts/exchange-v2/lib/LibMath.sol#91-126) is never used and should be removed
LibMath.safeGetPartialAmountCeil(uint256,uint256,uint256) (contracts/exchange-v2/lib/LibMath.sol#75-84) is never used and should be removed
LibPart.hash(LibPart.Part) (node_modules/@rarible/royalties/contracts/LibPart.sol#13-15) is never used and should be removed
MathUpgradeable.average(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/MathUpgradeable.sol#27-30) is never used and should be removed
MathUpgradeable.max(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/MathUpgradeable.sol#12-14) is never used and should be removed
MathUpgradeable.min(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/MathUpgradeable.sol#19-21) is never used and should be removed
OwnableUpgradeable.__Ownable_init() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#27-30) is never used and should be removed
SafeMathUpgradeable.div(uint256,uint256,string) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#190-193) is never used and should be removed
SafeMathUpgradeable.mod(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#152-155) is never used and should be removed
SafeMathUpgradeable.mod(uint256,uint256,string) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#210-213) is never used and should be removed
SafeMathUpgradeable.sub(uint256,uint256,string) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#170-173) is never used and should be removed
SafeMathUpgradeable.tryAdd(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#24-28) is never used and should be removed
SafeMathUpgradeable.tryDiv(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#60-63) is never used and should be removed
SafeMathUpgradeable.tryMod(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#70-73) is never used and should be removed
SafeMathUpgradeable.tryMul(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#45-53) is never used and should be removed
SafeMathUpgradeable.trySub(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#35-38) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/introspection/IERC165Upgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/math/MathUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#3) is too complex
Pragma version>=0.4.24<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/proxy/Initializable.sol#4) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/IERC1155Upgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/IERC20Upgradeable.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC721/IERC721Upgradeable.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#3) is too complex
Pragma version>=0.6.9<0.8.0 (node_modules/@rarible/exchange-interfaces/contracts/IERC20TransferProxy.sol#3) is too complex
Pragma version>=0.6.9<0.8.0 (node_modules/@rarible/exchange-interfaces/contracts/INftTransferProxy.sol#3) is too complex
Pragma version>=0.6.9<0.8.0 (node_modules/@rarible/exchange-interfaces/contracts/ITransferProxy.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/lazy-mint/contracts/erc-1155/LibERC1155LazyMint.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/lazy-mint/contracts/erc-721/LibERC721LazyMint.sol#3) is too complex
Pragma version^0.7.0 (node_modules/@rarible/libraries/contracts/LibSignature.sol#3) allows old versions
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/IRoyaltiesProvider.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/LibPart.sol#3) is too complex
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in AddressUpgradeable.sendValue(address,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#53-59):
        - (success) = recipient.call{value: amount}() (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#57)
Low level call in AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#114-121):
        - (success,returndata) = target.call{value: value}(data) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#119)
Low level call in AddressUpgradeable.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#139-145):
        - (success,returndata) = target.staticcall(data) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#143)
Low level call in LibTransfer.transferEth(address,uint256) (contracts/exchange-v2/lib/LibTransfer.sol#6-9):
        - (success) = to.call{value: value}() (contracts/exchange-v2/lib/LibTransfer.sol#7)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function OwnableUpgradeable.__Ownable_init() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#27-30) is not in mixedCase
Function OwnableUpgradeable.__Ownable_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#32-36) is not in mixedCase
Variable OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#76) is not in mixedCase
Function EIP712Upgradeable.__EIP712_init(string,string) (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#44-46) is not in mixedCase
Function EIP712Upgradeable.__EIP712_init_unchained(string,string) (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#48-53) is not in mixedCase
Function EIP712Upgradeable._EIP712NameHash() (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#107-109) is not in mixedCase
Function EIP712Upgradeable._EIP712VersionHash() (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#117-119) is not in mixedCase
Variable EIP712Upgradeable._HASHED_NAME (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#27) is not in mixedCase
Variable EIP712Upgradeable._HASHED_VERSION (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#28) is not in mixedCase
Variable EIP712Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#120) is not in mixedCase

Function ContextUpgradeable.__Context_init() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#17-19) is not in mixedCase
Function ContextUpgradeable.__Context_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#21-22) is not in mixedCase
Variable ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31) is not in mixedCase
Variable AssetMatcher.__gap (contracts/exchange-v2/AssetMatcher.sol#77) is not in mixedCase
Function ExchangeV2.__ExchangeV2_init(INftTransferProxy,IERC20TransferProxy,uint256,address,IRoyaltiesProvider) (contracts/exchange-v2/ExchangeV2.sol#11-23) is not in mixedCase
Parameter ExchangeV2.__ExchangeV2_init(INftTransferProxy,IERC20TransferProxy,uint256,address,IRoyaltiesProvider)._transferProxy (contracts/exchange-v2/ExchangeV2.sol#12) is not in mixedCase
Parameter ExchangeV2.__ExchangeV2_init(INftTransferProxy,IERC20TransferProxy,uint256,address,IRoyaltiesProvider)._erc20TransferProxy (contracts/exchange-v2/ExchangeV2.sol#13) is not in mixedCase
Variable ExchangeV2Core.__gap (contracts/exchange-v2/ExchangeV2Core.sol#130) is not in mixedCase
Function OrderValidator.__OrderValidator_init_unchained() (contracts/exchange-v2/OrderValidator.sol#18-20) is not in mixedCase
Variable OrderValidator.__gap (contracts/exchange-v2/OrderValidator.sol#50) is not in mixedCase
Function RaribleTransferManager.__RaribleTransferManager_init_unchained(uint256,address,IRoyaltiesProvider) (contracts/exchange-v2/RaribleTransferManager.sol#28-36) is not in mixedCase
Variable RaribleTransferManager.__gap (contracts/exchange-v2/RaribleTransferManager.sol#226) is not in mixedCase
Function TransferExecutor.__TransferExecutor_init_unchained(INftTransferProxy,IERC20TransferProxy) (contracts/exchange-v2/TransferExecutor.sol#21-25) is not in mixedCase
Variable TransferExecutor.__gap (contracts/exchange-v2/TransferExecutor.sol#57) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#94)" inEIP712Upgradeable (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#25-121)
Redundant expression "this (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#28)" inContextUpgradeable (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#16-32)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

Variable TransferExecutor.transfer(LibAsset.Asset,address,address,bytes4,bytes4).token_scope_0 (contracts/exchange-v2/TransferExecutor.sol#45) is too similar to TransferExecutor.transfer(LibAsset.Asset,address,address,bytes4,bytes4).toke
n_scope_1 (contracts/exchange-v2/TransferExecutor.sol#49)
Variable TransferExecutor.transfer(LibAsset.Asset,address,address,bytes4,bytes4).token_scope_0 (contracts/exchange-v2/TransferExecutor.sol#45) is too similar to RaribleTransferManager.getRoyaltiesByAssetType(LibAsset.AssetType).token_sco
pe_1 (contracts/exchange-v2/RaribleTransferManager.sol#147)
Variable RaribleTransferManager.getRoyaltiesByAssetType(LibAsset.AssetType).token_scope_0 (contracts/exchange-v2/RaribleTransferManager.sol#144) is too similar to RaribleTransferManager.getRoyaltiesByAssetType(LibAsset.AssetType).token_s
cope_1 (contracts/exchange-v2/RaribleTransferManager.sol#147)
Variable RaribleTransferManager.getRoyaltiesByAssetType(LibAsset.AssetType).token_scope_0 (contracts/exchange-v2/RaribleTransferManager.sol#144) is too similar to TransferExecutor.transfer(LibAsset.Asset,address,address,bytes4,bytes4).to
ken_scope_1 (contracts/exchange-v2/TransferExecutor.sol#49)
Variable ITransferManager.doTransfers(LibAsset.AssetType,LibAsset.AssetType,LibFill.FillResult,LibOrder.Order,LibOrder.Order,LibOrderDataV2.DataV2,LibOrderDataV2.DataV2).totalMakeValue (contracts/exchange-v2/ITransferManager.sol#27) is t
oo similar to ExchangeV2Core.matchAndTransfer(LibOrder.Order,LibOrder.Order).totalTakeValue (contracts/exchange-v2/ExchangeV2Core.sol#62)
Variable ITransferManager.doTransfers(LibAsset.AssetType,LibAsset.AssetType,LibFill.FillResult,LibOrder.Order,LibOrder.Order,LibOrderDataV2.DataV2,LibOrderDataV2.DataV2).totalMakeValue (contracts/exchange-v2/ITransferManager.sol#27) is t
oo similar to ITransferManager.doTransfers(LibAsset.AssetType,LibAsset.AssetType,LibFill.FillResult,LibOrder.Order,LibOrder.Order,LibOrderDataV2.DataV2,LibOrderDataV2.DataV2).totalTakeValue (contracts/exchange-v2/ITransferManager.sol#27)
Variable RaribleTransferManager.doTransfers(LibAsset.AssetType,LibAsset.AssetType,LibFill.FillResult,LibOrder.Order,LibOrder.Order,LibOrderDataV2.DataV2,LibOrderDataV2.DataV2).totalMakeValue (contracts/exchange-v2/RaribleTransferMan
ager.sol#70) is too similar to ExchangeV2Core.matchAndTransfer(LibOrder.Order,LibOrder.Order).totalTakeValue (contracts/exchange-v2/ExchangeV2Core.sol#62)
Variable RaribleTransferManager.doTransfers(LibAsset.AssetType,LibAsset.AssetType,LibFill.FillResult,LibOrder.Order,LibOrder.Order,LibOrderDataV2.DataV2,LibOrderDataV2.DataV2).totalTakeValue (contracts/exchange-v2/RaribleTransferMan
ager.sol#70) is too similar to RaribleTransferManager.doTransfers(LibAsset.AssetType,LibAsset.AssetType,LibFill.FillResult,LibOrder.Order,LibOrder.Order,LibOrderDataV2.DataV2,LibOrderDataV2.DataV2).totalTakeValue (contracts/exchange-v2/Ribbl
eTransferManager.sol#70)
Variable ExchangeV2Core.matchAndTransfer(LibOrder.Order,LibOrder.Order).totalMakeValue (contracts/exchange-v2/ExchangeV2Core.sol#62) is too similar to ExchangeV2Core.matchAndTransfer(LibOrder.Order,LibOrder.Order).totalTakeValue (contrac
ts/exchange-v2/ExchangeV2Core.sol#62)
Variable ITransferManager.doTransfers(LibAsset.AssetType,LibAsset.AssetType,LibFill.FillResult,LibOrder.Order,LibOrder.Order,LibOrderDataV2.DataV2,LibOrderDataV2.DataV2).totalMakeValue (contracts/exchange-v2/ITransferManager.sol#27) is t
oo similar to RaribleTransferManager.doTransfers(LibAsset.AssetType,LibAsset.AssetType,LibFill.FillResult,LibOrder.Order,LibOrder.Order,LibOrderDataV2.DataV2,LibOrderDataV2.DataV2).totalTakeValue (contracts/exchange-v2/RaribleTransferMan
ager.sol#70)
Variable ExchangeV2Core.matchAndTransfer(LibOrder.Order,LibOrder.Order).totalMakeValue (contracts/exchange-v2/ExchangeV2Core.sol#62) is too similar to RaribleTransferManager.doTransfers(LibAsset.AssetType,LibAsset.AssetType,LibFill.FillR
esult,LibOrder.Order,LibOrder.Order,LibOrderDataV2.DataV2,LibOrderDataV2.DataV2).totalTakeValue (contracts/exchange-v2/RaribleTransferManager.sol#70)
Variable ExchangeV2Core.matchAndTransfer(LibOrder.Order,LibOrder.Order).totalMakeValue (contracts/exchange-v2/ExchangeV2Core.sol#62) is too similar to ITransferManager.doTransfers(LibAsset.AssetType,LibAsset.AssetType,LibFill.FillResult,
LibOrder.Order,LibOrder.Order,LibOrderDataV2.DataV2,LibOrderDataV2.DataV2).totalTakeValue (contracts/exchange-v2/ITransferManager.sol#27)
Variable RaribleTransferManager.doTransfers(LibAsset.AssetType,LibAsset.AssetType,LibFill.FillResult,LibOrder.Order,LibOrder.Order,LibOrderDataV2.DataV2,LibOrderDataV2.DataV2).totalMakeValue (contracts/exchange-v2/RaribleTransferManager.
sol#70) is too similar to ITransferManager.doTransfers(LibAsset.AssetType,LibAsset.AssetType,LibFill.FillResult,LibOrder.Order,LibOrder.Order,LibOrderDataV2.DataV2,LibOrderDataV2.DataV2).totalTakeValue (contracts/exchange-v2/ITransferMan
ager.sol#27)
Variable LibFill.fillOrder(LibOrder.Order,LibOrder.Order,uint256,uint256,bool,bool).leftMakeValue (contracts/exchange-v2/LibFill.sol#27) is too similar to LibFill.fillRight(uint256,uint256,uint256,uint256).leftTakeValue (contracts/exchan
ge-v2/LibFill.sol#37)
Variable LibFill.fillLeft(uint256,uint256,uint256,uint256).leftMakeValue (contracts/exchange-v2/LibFill.sol#43) is too similar to LibFill.fillRight(uint256,uint256,uint256,uint256).leftTakeValue (contracts/exchange-v2/LibFill.sol#37)
Variable LibFill.fillRight(uint256,uint256,uint256,uint256).leftMakeValue (contracts/exchange-v2/LibFill.sol#37) is too similar to LibFill.fillRight(uint256,uint256,uint256,uint256).leftTakeValue (contracts/exchange-v2/LibFill.sol#37)
Variable LibFill.fillRight(uint256,uint256,uint256,uint256).leftMakeValue (contracts/exchange-v2/LibFill.sol#37) is too similar to LibFill.fillOrder(LibOrder.Order,LibOrder.Order,uint256,uint256,bool,bool).leftTakeValue (contracts/exchan
ge-v2/LibFill.sol#27)
Variable LibFill.fillOrder(LibOrder.Order,LibOrder.Order,uint256,uint256,bool,bool).leftMakeValue (contracts/exchange-v2/LibFill.sol#27) is too similar to LibFill.fillOrder(LibOrder.Order,LibOrder.Order,uint256,uint256,bool,bool).leftTak
eValue (contracts/exchange-v2/LibFill.sol#27)
Variable LibFill.fillLeft(uint256,uint256,uint256,uint256).leftMakeValue (contracts/exchange-v2/LibFill.sol#43) is too similar to LibFill.fillLeft(uint256,uint256,uint256,uint256).leftTakeValue (contracts/exchange-v2/LibFill.sol#43)
Variable LibFill.fillLeft(uint256,uint256,uint256,uint256).leftMakeValue (contracts/exchange-v2/LibFill.sol#43) is too similar to LibFill.fillOrder(LibOrder.Order,LibOrder.Order,uint256,uint256,bool,bool).leftTakeValue (contracts/exchang
e-v2/LibFill.sol#27)
Variable LibFill.fillOrder(LibOrder.Order,LibOrder.Order,uint256,uint256,bool,bool).leftMakeValue (contracts/exchange-v2/LibFill.sol#27) is too similar to LibFill.fillLeft(uint256,uint256,uint256,uint256).leftTakeValue (contracts/exchang
e-v2/LibFill.sol#43)
Variable LibFill.fillRight(uint256,uint256,uint256,uint256).leftMakeValue (contracts/exchange-v2/LibFill.sol#37) is too similar to LibFill.fillLeft(uint256,uint256,uint256,uint256).leftTakeValue (contracts/exchange-v2/LibFill.sol#43)
Variable LibFill.fillOrder(LibOrder.Order,LibOrder.Order,uint256,uint256,bool,bool).rightMakeValue (contracts/exchange-v2/LibFill.sol#28) is too similar to LibFill.fillLeft(uint256,uint256,uint256,uint256).rightTakeValue (contracts/excha
nge-v2/LibFill.sol#43)
Variable LibFill.fillOrder(LibOrder.Order,LibOrder.Order,uint256,uint256,bool,bool).rightMakeValue (contracts/exchange-v2/LibFill.sol#28) is too similar to LibFill.fillOrder(LibOrder.Order,LibOrder.Order,uint256,uint256,bool,bool).rightT
akeValue (contracts/exchange-v2/LibFill.sol#28)
Variable LibFill.fillLeft(uint256,uint256,uint256,uint256).rightMakeValue (contracts/exchange-v2/LibFill.sol#43) is too similar to LibFill.fillRight(uint256,uint256,uint256,uint256).rightTakeValue (contracts/exchange-v2/LibFill.sol#37)
Variable LibFill.fillLeft(uint256,uint256,uint256,uint256).rightMakeValue (contracts/exchange-v2/LibFill.sol#43) is too similar to LibFill.fillOrder(LibOrder.Order,LibOrder.Order,uint256,uint256,bool,bool).rightTakeValue (contracts/excha
nge-v2/LibFill.sol#28)
Variable LibFill.fillOrder(LibOrder.Order,LibOrder.Order,uint256,uint256,bool,bool).rightMakeValue (contracts/exchange-v2/LibFill.sol#28) is too similar to LibFill.fillRight(uint256,uint256,uint256,uint256).rightTakeValue (contracts/exch
ange-v2/LibFill.sol#37)
Variable LibFill.fillRight(uint256,uint256,uint256,uint256).rightMakeValue (contracts/exchange-v2/LibFill.sol#37) is too similar to LibFill.fillRight(uint256,uint256,uint256,uint256).rightTakeValue (contracts/exchange-v2/LibFill.sol#37)
Variable LibFill.fillRight(uint256,uint256,uint256,uint256).rightMakeValue (contracts/exchange-v2/LibFill.sol#37) is too similar to LibFill.fillOrder(LibOrder.Order,LibOrder.Order,uint256,uint256,bool,bool).rightTakeValue (contracts/exch
ange-v2/LibFill.sol#28)
Variable LibFill.fillRight(uint256,uint256,uint256,uint256).rightMakeValue (contracts/exchange-v2/LibFill.sol#37) is too similar to LibFill.fillLeft(uint256,uint256,uint256,uint256).rightTakeValue (contracts/exchange-v2/LibFill.sol#43)
Variable LibFill.fillLeft(uint256,uint256,uint256,uint256).rightMakeValue (contracts/exchange-v2/LibFill.sol#43) is too similar to LibFill.fillLeft(uint256,uint256,uint256,uint256).rightTakeValue (contracts/exchange-v2/LibFill.sol#43)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-are-too-similar

LibERC1155LazyMint._INTERFACE_ID_MINT_AND_TRANSFER (node_modules/@rarible/lazy-mint/contracts/erc-1155/LibERC1155LazyMint.sol#9) is never used in LibERC1155LazyMint (node_modules/@rarible/lazy-mint/contracts/erc-1155/LibERC1155LazyMint.s
ol#7-40)
LibERC721LazyMint._INTERFACE_ID_MINT_AND_TRANSFER (node_modules/@rarible/lazy-mint/contracts/erc-721/LibERC721LazyMint.sol#9) is never used in LibERC721LazyMint (node_modules/@rarible/lazy-mint/contracts/erc-721/LibERC721LazyMint.sol#7-3
9)
RaribleTransferManager.__gap (contracts/exchange-v2/RaribleTransferManager.sol#226) is never used in ExchangeV2 (contracts/exchange-v2/ExchangeV2.sol#10-25)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

renounceOwnership() should be declared external:
        - OwnableUpgradeable.renounceOwnership() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#60-63)
transferOwnership(address) should be declared external:
        - OwnableUpgradeable.transferOwnership(address) (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#69-73)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

RoyaltiesRegistry.sol

```
OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#74) shadows:
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
RoyaltiesRegistry.__gap (contracts/royalties-registry/RoyaltiesRegistry.sol#250) shadows:
        - OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#74)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variable-shadowing

LibRoyalties2981.calculateRoyalties(address,uint256) (node_modules/@rarible/royalties/contracts/LibRoyalties2981.sol#15-26) performs a multiplication on the result of a division:
        -percent = (amount * 100 / _WEIGHT_VALUE) * 100 (node_modules/@rarible/royalties/contracts/LibRoyalties2981.sol#20)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#divide-before-multiply

RoyaltiesRegistry.getRoyaltiesRaribleV2(address,uint256).result (contracts/royalties-registry/RoyaltiesRegistry.sol#199) is a local variable never initialized
RoyaltiesRegistry.calculateRoyaltiesType(address,address).result (contracts/royalties-registry/RoyaltiesRegistry.sol#119) is a local variable never initialized
RoyaltiesRegistry.calculateRoyaltiesType(address,address).result_scope_0 (contracts/royalties-registry/RoyaltiesRegistry.sol#125) is a local variable never initialized
RoyaltiesRegistry.getRoyaltiesRaribleV1(address,uint256).values (contracts/royalties-registry/RoyaltiesRegistry.sol#215) is a local variable never initialized
LibRoyalties2981.calculateRoyalties(address,uint256).result (node_modules/@rarible/royalties/contracts/LibRoyalties2981.sol#16) is a local variable never initialized
RoyaltiesRegistry.getRoyaltiesEIP2981(address,uint256).receiver (contracts/royalties-registry/RoyaltiesRegistry.sol#234) is a local variable never initialized
RoyaltiesRegistry.getRoyaltiesRaribleV1(address,uint256).recipients (contracts/royalties-registry/RoyaltiesRegistry.sol#209) is a local variable never initialized
RoyaltiesRegistry.calculateRoyaltiesType(address,address).result_scope_1 (contracts/royalties-registry/RoyaltiesRegistry.sol#131) is a local variable never initialized
RoyaltiesRegistry.providerExtractor(address,uint256,address).result (contracts/royalties-registry/RoyaltiesRegistry.sol#243) is a local variable never initialized
RoyaltiesRegistry.getRoyaltiesEIP2981(address,uint256).royaltyAmount (contracts/royalties-registry/RoyaltiesRegistry.sol#234) is a local variable never initialized
RoyaltiesRegistry.getRoyaltiesRaribleV1(address,uint256).resultRecipients (contracts/royalties-registry/RoyaltiesRegistry.sol#210) is a local variable never initialized
RoyaltiesRegistry.getRoyaltiesRaribleV1(address,uint256).resultValues (contracts/royalties-registry/RoyaltiesRegistry.sol#216) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

RoyaltiesRegistry.calculateRoyaltiesType(address,address) (contracts/royalties-registry/RoyaltiesRegistry.sol#118-146) ignores return value by IERC165Upgradeable(token).supportsInterface(LibRoyaltiesV2._INTERFACE_ID_ROYALTIES) (contracts
/royalties-registry/RoyaltiesRegistry.sol#118-123)
RoyaltiesRegistry.calculateRoyaltiesType(address,address) (contracts/royalties-registry/RoyaltiesRegistry.sol#118-146) ignores return value by IERC165Upgradeable(token).supportsInterface(LibRoyaltiesV1._INTERFACE_ID_FEES) (contracts/roya
lties-registry/RoyaltiesRegistry.sol#125-129)
RoyaltiesRegistry.calculateRoyaltiesType(address,address) (contracts/royalties-registry/RoyaltiesRegistry.sol#118-146) ignores return value by IERC165Upgradeable(token).supportsInterface(LibRoyalties2981._INTERFACE_ID_ROYALTIES) (contrac
ts/royalties-registry/RoyaltiesRegistry.sol#131-135)
RoyaltiesRegistry.getRoyaltiesRaribleV2(address,uint256) (contracts/royalties-registry/RoyaltiesRegistry.sol#198-204) ignores return value by RoyaltiesV2(token).getRaribleV2Royalties(tokenId) (contracts/royalties-registry/RoyaltiesRegist
ry.sol#199-203)
RoyaltiesRegistry.getRoyaltiesRaribleV1(address,uint256) (contracts/royalties-registry/RoyaltiesRegistry.sol#207-230) ignores return value by v1.getFeeRecipients(tokenId) (contracts/royalties-registry/RoyaltiesRegistry.sol#210-214)
RoyaltiesRegistry.getRoyaltiesRaribleV1(address,uint256) (contracts/royalties-registry/RoyaltiesRegistry.sol#207-230) ignores return value by v1.getFeeBps(tokenId) (contracts/royalties-registry/RoyaltiesRegistry.sol#216-220)
RoyaltiesRegistry.getRoyaltiesEIP2981(address,uint256) (contracts/royalties-registry/RoyaltiesRegistry.sol#233-239) ignores return value by IERC2981(token).royaltyInfo(tokenId,LibRoyalties2981._WEIGHT_VALUE) (contracts/royalties-registry
/RoyaltiesRegistry.sol#234-238)
RoyaltiesRegistry.providerExtractor(address,uint256,address) (contracts/royalties-registry/RoyaltiesRegistry.sol#242-248) ignores return value by IRoyaltiesProvider(providerAddress).getRoyalties(token,tokenId) (contracts/royalties-regist
ry/RoyaltiesRegistry.sol#243-247)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

Variable 'RoyaltiesRegistry.calculateRoyaltiesType(address,address).result (contracts/royalties-registry/RoyaltiesRegistry.sol#119)' in RoyaltiesRegistry.calculateRoyaltiesType(address,address) (contracts/royalties-registry/RoyaltiesRegi
stry.sol#118-146) potentially used before declaration: result (contracts/royalties-registry/RoyaltiesRegistry.sol#120)
Variable 'RoyaltiesRegistry.calculateRoyaltiesType(address,address).result_scope_0 (contracts/royalties-registry/RoyaltiesRegistry.sol#125)' in RoyaltiesRegistry.calculateRoyaltiesType(address,address) (contracts/royalties-registry/Royal
tiesRegistry.sol#118-146) potentially used before declaration: result_scope_0 (contracts/royalties-registry/RoyaltiesRegistry.sol#126)
Variable 'RoyaltiesRegistry.calculateRoyaltiesType(address,address).result_scope_1 (contracts/royalties-registry/RoyaltiesRegistry.sol#131)' in RoyaltiesRegistry.calculateRoyaltiesType(address,address) (contracts/royalties-registry/Royal
tiesRegistry.sol#118-146) potentially used before declaration: result_scope_1 (contracts/royalties-registry/RoyaltiesRegistry.sol#132)
Variable 'RoyaltiesRegistry.getRoyaltiesRaribleV2(address,uint256).result (contracts/royalties-registry/RoyaltiesRegistry.sol#199)' in RoyaltiesRegistry.getRoyaltiesRaribleV2(address,uint256) (contracts/royalties-registry/RoyaltiesRegist
ry.sol#198-204) potentially used before declaration: result (contracts/royalties-registry/RoyaltiesRegistry.sol#200)
Variable 'RoyaltiesRegistry.getRoyaltiesRaribleV1(address,uint256).resultRecipients (contracts/royalties-registry/RoyaltiesRegistry.sol#210)' in RoyaltiesRegistry.getRoyaltiesRaribleV1(address,uint256) (contracts/royalties-registry/Royal
tiesRegistry.sol#207-230) potentially used before declaration: recipients = resultRecipients (contracts/royalties-registry/RoyaltiesRegistry.sol#211)
Variable 'RoyaltiesRegistry.getRoyaltiesRaribleV1(address,uint256).resultValues (contracts/royalties-registry/RoyaltiesRegistry.sol#216)' in RoyaltiesRegistry.getRoyaltiesRaribleV1(address,uint256) (contracts/royalties-registry/Royalties
Registry.sol#207-230) potentially used before declaration: values = resultValues (contracts/royalties-registry/RoyaltiesRegistry.sol#217)
Variable 'RoyaltiesRegistry.getRoyaltiesEIP2981(address,uint256).royaltyAmount (contracts/royalties-registry/RoyaltiesRegistry.sol#234)' in RoyaltiesRegistry.getRoyaltiesEIP2981(address,uint256) (contracts/royalties-registry/RoyaltiesReg
istry.sol#233-239) potentially used before declaration: LibRoyalties2981.calculateRoyalties(receiver,royaltyAmount) (contracts/royalties-registry/RoyaltiesRegistry.sol#235)
Variable 'RoyaltiesRegistry.getRoyaltiesEIP2981(address,uint256).receiver (contracts/royalties-registry/RoyaltiesRegistry.sol#234)' in RoyaltiesRegistry.getRoyaltiesEIP2981(address,uint256) (contracts/royalties-registry/RoyaltiesRegistry
.sol#233-239) potentially used before declaration: LibRoyalties2981.calculateRoyalties(receiver,royaltyAmount) (contracts/royalties-registry/RoyaltiesRegistry.sol#235)
Variable 'RoyaltiesRegistry.providerExtractor(address,uint256,address).result (contracts/royalties-registry/RoyaltiesRegistry.sol#243)' in RoyaltiesRegistry.providerExtractor(address,uint256,address) (contracts/royalties-registry/Royalti
esRegistry.sol#242-248) potentially used before declaration: result (contracts/royalties-registry/RoyaltiesRegistry.sol#244)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables

AddressUpgradeable.isContract(address) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#26-35) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#33)
AddressUpgradeable._verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#147-164) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#156-159)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Different versions of Solidity is used:
        - Version used: ['>=0.4.24<0.8.0', '>=0.6.0<0.8.0', '>=0.6.2<0.8.0']
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/introspection/IERC165Upgradeable.sol#3)
        - >=0.4.24<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/proxy/Initializable.sol#4)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC721/IERC721Upgradeable.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/IERC2981.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/IRoyaltiesProvider.sol#3)
        - v2 (node_modules/@rarible/royalties/contracts/IRoyaltiesProvider.sol#4)
        - >=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/LibPart.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/LibRoyalties2981.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/LibRoyaltiesV1.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/LibRoyaltiesV2.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/RoyaltiesV1.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/RoyaltiesV2.sol#3)
        - v2 (node_modules/@rarible/royalties/contracts/RoyaltiesV2.sol#4)
        - >=0.6.2<0.8.0 (contracts/royalties-registry/RoyaltiesRegistry.sol#3)
        - v2 (contracts/royalties-registry/RoyaltiesRegistry.sol#4)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

AddressUpgradeable._verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#147-164) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#79-81) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#88-91) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#104-106) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#114-121) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#129-131) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#139-145) is never used and should be removed
AddressUpgradeable.sendValue(address,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#53-59) is never used and should be removed
ContextUpgradeable.__Context_init() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#17-19) is never used and should be removed
ContextUpgradeable.__Context_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#21-22) is never used and should be removed
ContextUpgradeable._msgData() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#27-30) is never used and should be removed
LibPart.hash(LibPart.Part) (node_modules/@rarible/royalties/contracts/LibPart.sol#13-15) is never used and should be removed
OwnableUpgradeable.__Ownable_init() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#27-30) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/introspection/IERC165Upgradeable.sol#3) is too complex
Pragma version>=0.4.24<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/proxy/Initializable.sol#4) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC721/IERC721Upgradeable.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/IERC2981.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/IRoyaltiesProvider.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/LibPart.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/LibRoyalties2981.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/LibRoyaltiesV1.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/LibRoyaltiesV2.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/RoyaltiesV1.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/RoyaltiesV2.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (contracts/royalties-registry/RoyaltiesRegistry.sol#3) is too complex
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

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

Function OwnableUpgradeable.__Ownable_init() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#27-30) is not in mixedCase
Function OwnableUpgradeable.__Ownable_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#32-36) is not in mixedCase
Variable OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#74) is not in mixedCase
Function ContextUpgradeable.__Context_init() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#17-18) is not in mixedCase
Function ContextUpgradeable.__Context_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#21-22) is not in mixedCase
Variable ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31) is not in mixedCase
Function RoyaltiesRegistry.__RoyaltiesRegistry_init() (contracts/royalties-registry/RoyaltiesRegistry.sol#42-44) is not in mixedCase
Variable RoyaltiesRegistry.__gap (contracts/royalties-registry/RoyaltiesRegistry.sol#250) is not in mixedCase
Constant RoyaltiesRegistry.royaltiesTypesAmount (contracts/royalties-registry/RoyaltiesRegistry.sol#40) is not in UPPER_CASE_WITH_UNDERSCORES
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#28)" inContextUpgradeable (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#16-32)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

Variable RoyaltiesRegistry.calculateRoyaltiesType(address,address).result_scope_0 (contracts/royalties-registry/RoyaltiesRegistry.sol#125) is too similar to RoyaltiesRegistry.calculateRoyaltiesType(address,address).result_scope_1 (contra
cts/royalties-registry/RoyaltiesRegistry.sol#131)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-are-too-similar

LibRoyalties2981.slitherConstructorConstantVariables() (node_modules/@rarible/royalties/contracts/LibRoyalties2981.sol#7-27) uses literals with too many digits:
        - _WEIGHT_VALUE = 1000000 (node_modules/@rarible/royalties/contracts/LibRoyalties2981.sol#12)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits

LibRoyalties2981._INTERFACE_ID_ROYALTIES (node_modules/@rarible/royalties/contracts/LibRoyalties2981.sol#11) is never used in LibRoyalties2981 (node_modules/@rarible/royalties/contracts/LibRoyalties2981.sol#7-27)
RoyaltiesRegistry.__gap (contracts/royalties-registry/RoyaltiesRegistry.sol#250) is never used in RoyaltiesRegistry (contracts/royalties-registry/RoyaltiesRegistry.sol#16-251)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

renounceOwnership() should be declared external:
        - OwnableUpgradeable.renounceOwnership() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#60-63)
transferOwnership(address) should be declared external:
        - OwnableUpgradeable.transferOwnership(address) (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#69-73)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

ERC721WoonklyNFT.sol

```
OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#76) shadows:
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
HasContractURI.__gap (contracts/HasContractURI.sol#29) shadows:
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
Mint721Validator.__gap (contracts/Mint721Validator.sol#16) shadows:
        - ERC1271Validator.__gap (contracts/erc-1271/ERC1271Validator.sol#34)
        - EIP712Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#120)
ERC1271Validator.__gap (contracts/erc-1271/ERC1271Validator.sol#34) shadows:
        - EIP712Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#120)
ERC721Base.__gap (contracts/erc-721/ERC721Base.sol#30) shadows:
        - HasContractURI.__gap (contracts/HasContractURI.sol#29)
        - ERC721Lazy.__gap (contracts/erc-721/ERC721Lazy.sol#114)
        - Mint721Validator.__gap (contracts/Mint721Validator.sol#16)
        - ERC1271Validator.__gap (contracts/erc-1271/ERC1271Validator.sol#34)
        - EIP712Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#120)
        - ERC721BurnableUpgradeable.__gap (contracts/erc-721/ERC721BurnableUpgradeable.sol#40)
        - ERC721DefaultApproval.__gap (contracts/erc-721/ERC721DefaultApproval.sol#24)
        - ERC721Upgradeable.__gap (contracts/erc-721/ERC721Upgradeable.sol#495)
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#74)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
ERC721BurnableUpgradeable.__gap (contracts/erc-721/ERC721BurnableUpgradeable.sol#40) shadows:
        - ERC721Upgradeable.__gap (contracts/erc-721/ERC721Upgradeable.sol#495)
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
ERC721DefaultApproval.__gap (contracts/erc-721/ERC721DefaultApproval.sol#24) shadows:
        - ERC721Upgradeable.__gap (contracts/erc-721/ERC721Upgradeable.sol#495)
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
ERC721Lazy.__gap (contracts/erc-721/ERC721Lazy.sol#114) shadows:
        - Mint721Validator.__gap (contracts/Mint721Validator.sol#16)
        - ERC1271Validator.__gap (contracts/erc-1271/ERC1271Validator.sol#34)
        - EIP712Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#120)
        - ERC721Upgradeable.__gap (contracts/erc-721/ERC721Upgradeable.sol#495)
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
ERC721Lazy.INTERFACE_ID_ERC165 (contracts/erc-721/ERC721Lazy.sol#17) shadows:
        - ERC165Upgradeable.INTERFACE_ID_ERC165 (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#18)
ERC721Lazy.INTERFACE_ID_ERC721 (contracts/erc-721/ERC721Lazy.sol#18) shadows:
        - ERC721Upgradeable.INTERFACE_ID_ERC721 (contracts/erc-721/ERC721Upgradeable.sol#72)
ERC721Lazy.INTERFACE_ID_ERC721_METADATA (contracts/erc-721/ERC721Lazy.sol#19) shadows:
        - ERC721Upgradeable.INTERFACE_ID_ERC721_METADATA (contracts/erc-721/ERC721Upgradeable.sol#81)
ERC721Lazy.INTERFACE_ID_ERC721_ENUMERABLE (contracts/erc-721/ERC721Lazy.sol#20) shadows:
        - ERC721Upgradeable.INTERFACE_ID_ERC721_ENUMERABLE (contracts/erc-721/ERC721Upgradeable.sol#90)
ERC721Rarible.__gap (contracts/erc-721/ERC721Rarible.sol#36) shadows:
        - ERC721Base.__gap (contracts/erc-721/ERC721Base.sol#30)
        - HasContractURI.__gap (contracts/HasContractURI.sol#29)
        - ERC721Lazy.__gap (contracts/erc-721/ERC721Lazy.sol#114)
        - Mint721Validator.__gap (contracts/Mint721Validator.sol#16)
        - ERC1271Validator.__gap (contracts/erc-1271/ERC1271Validator.sol#34)
        - EIP712Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#120)
        - ERC721BurnableUpgradeable.__gap (contracts/erc-721/ERC721BurnableUpgradeable.sol#40)
        - ERC721DefaultApproval.__gap (contracts/erc-721/ERC721DefaultApproval.sol#24)
        - ERC721Upgradeable.__gap (contracts/erc-721/ERC721Upgradeable.sol#495)
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#74)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
ERC721Upgradeable.__gap (contracts/erc-721/ERC721Upgradeable.sol#495) shadows:
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variable-shadowing

ERC1271Validator.validate1271(address,bytes32,bytes).signerFromSig (contracts/erc-1271/ERC1271Validator.sol#19) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

ERC721Lazy._mint(address,uint256) (contracts/erc-721/ERC721Lazy.sol#72-90) ignores return value by _holderTokens[to].add(tokenId) (contracts/erc-721/ERC721Lazy.sol#79)
ERC721Lazy._mint(address,uint256) (contracts/erc-721/ERC721Lazy.sol#72-90) ignores return value by _tokenOwners.set(tokenId,to) (contracts/erc-721/ERC721Lazy.sol#81)
ERC721Upgradeable._mint(address,uint256) (contracts/erc-721/ERC721Upgradeable.sol#344-356) ignores return value by _holderTokens[to].add(tokenId) (contracts/erc-721/ERC721Upgradeable.sol#351)
ERC721Upgradeable._mint(address,uint256) (contracts/erc-721/ERC721Upgradeable.sol#344-356) ignores return value by _tokenOwners.set(tokenId,to) (contracts/erc-721/ERC721Upgradeable.sol#353)
ERC721Upgradeable._burn(uint256) (contracts/erc-721/ERC721Upgradeable.sol#368-386) ignores return value by _holderTokens[owner].remove(tokenId) (contracts/erc-721/ERC721Upgradeable.sol#381)
ERC721Upgradeable._burn(uint256) (contracts/erc-721/ERC721Upgradeable.sol#368-386) ignores return value by _tokenOwners.remove(tokenId) (contracts/erc-721/ERC721Upgradeable.sol#383)
ERC721Upgradeable._transfer(address,address,uint256) (contracts/erc-721/ERC721Upgradeable.sol#409-424) ignores return value by _holderTokens[from].remove(tokenId) (contracts/erc-721/ERC721Upgradeable.sol#418)
ERC721Upgradeable._transfer(address,address,uint256) (contracts/erc-721/ERC721Upgradeable.sol#409-424) ignores return value by _holderTokens[to].add(tokenId) (contracts/erc-721/ERC721Upgradeable.sol#419)
ERC721Upgradeable._transfer(address,address,uint256) (contracts/erc-721/ERC721Upgradeable.sol#409-424) ignores return value by _tokenOwners.set(tokenId,to) (contracts/erc-721/ERC721Upgradeable.sol#421)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

ERC721Base.isApprovedForAll(address,address).owner (contracts/erc-721/ERC721Base.sol#18) shadows:
        - OwnableUpgradeable.owner() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#41-43) (function)
ERC721Rarible._ERC721Rarible_init(string,string,string,string,address,address)._name (contracts/erc-721/ERC721Rarible.sol#12) shadows:
        - ERC721Upgradeable._name (contracts/erc-721/ERC721Upgradeable.sol#47) (state variable)
ERC721Rarible.__ERC721Rarible_init(string,string,string,string,address,address)._symbol (contracts/erc-721/ERC721Rarible.sol#12) shadows:
        - ERC721Upgradeable._symbol (contracts/erc-721/ERC721Upgradeable.sol#50) (state variable)
ERC721Rarible.__ERC721Rarible_init(string,string,string,string,address,address).baseURI (contracts/erc-721/ERC721Rarible.sol#12) shadows:
        - ERC721Upgradeable.baseURI() (contracts/erc-721/ERC721Upgradeable.sol#169-171) (function)
ERC721Rarible.__ERC721Rarible_init(string,string,string,string,address,address).contractURI (contracts/erc-721/ERC721Rarible.sol#12) shadows:
        - HasContractURI.contractURI (contracts/HasContractURI.sol#9) (state variable)
ERC721Rarible.__ERC721Rarible_init_unchained(string,string,string,string,address,address)._name (contracts/erc-721/ERC721Rarible.sol#17) shadows:
        - ERC721Upgradeable._name (contracts/erc-721/ERC721Upgradeable.sol#47) (state variable)
ERC721Rarible.__ERC721Rarible_init_unchained(string,string,string,string,address,address)._symbol (contracts/erc-721/ERC721Rarible.sol#17) shadows:
        - ERC721Upgradeable._symbol (contracts/erc-721/ERC721Upgradeable.sol#50) (state variable)
ERC721Rarible.__ERC721Rarible_init_unchained(string,string,string,string,address,address).baseURI (contracts/erc-721/ERC721Rarible.sol#17) shadows:
        - ERC721Upgradeable.baseURI() (contracts/erc-721/ERC721Upgradeable.sol#169-171) (function)
ERC721Rarible.__ERC721Rarible_init_unchained(string,string,string,string,address,address).contractURI (contracts/erc-721/ERC721Rarible.sol#17) shadows:
        - HasContractURI.contractURI (contracts/HasContractURI.sol#9) (state variable)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

ERC1271Validator.validate1271(address,bytes32,bytes) (contracts/erc-1271/ERC1271Validator.sol#16-33) has external calls inside a loop: require(bool,string)(ERC1271(signer).isValidSignature(hash,signature) == MAGICVALUE,SIGNATURE_ERROR) (
contracts/erc-1271/ERC1271Validator.sol#25-28)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation/#calls-inside-a-loop

Reentrancy in ERC721Lazy.mintAndTransfer(LibERC721LazyMint.Mint721Data,address) (contracts/erc-721/ERC721Lazy.sol#50-70):
        External calls:
        - _safeMint(to,data.tokenId) (contracts/erc-721/ERC721Lazy.sol#66)
                - returndata = to.functionCall(abi.encodeWithSelector(IERC721ReceiverUpgradeable(to).onERC721Received.selector,_msgSender(),from,tokenId,_data),ERC721: transfer to non ERC721Receiver implementer) (contracts/erc-721/ERC721
Upgradeable.sol#463-469)
                - (success,returndata) = target.call{value: value}(data) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#119)
        External calls sending eth:
        - _safeMint(to,data.tokenId) (contracts/erc-721/ERC721Lazy.sol#66)
                - (success,returndata) = target.call{value: value}(data) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#119)
        State variables written after the call(s):
        - _setTokenURI(data.tokenId,data.tokenURI) (contracts/erc-721/ERC721Lazy.sol#69)
                - _tokenURIs[tokenId] = _tokenURI (contracts/erc-721/ERC721Upgradeable.sol#435)
        - _saveCreators(data.tokenId,data.creators) (contracts/erc-721/ERC721Lazy.sol#68)
                - creatorsOfToken.push(_creators[i]) (contracts/erc-721/ERC721Lazy.sol#98)
        - _saveRoyalties(data.tokenId,data.royalties) (contracts/erc-721/ERC721Lazy.sol#67)
                - royalties[id].push(_royalties[i]) (node_modules/@rarible/royalties/contracts/impl/AbstractRoyalties.sol#16)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2

Reentrancy in ERC721Lazy.mintAndTransfer(LibERC721LazyMint.Mint721Data,address) (contracts/erc-721/ERC721Lazy.sol#50-70):
        External calls:
        - _safeMint(to,data.tokenId) (contracts/erc-721/ERC721Lazy.sol#66)
                - returndata = to.functionCall(abi.encodeWithSelector(IERC721ReceiverUpgradeable(to).onERC721Received.selector,_msgSender(),from,tokenId,_data),ERC721: transfer to non ERC721Receiver implementer) (contracts/erc-721/ERC721
Upgradeable.sol#463-469)
                - (success,returndata) = target.call{value: value}(data) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#119)
        External calls sending eth:
        - _safeMint(to,data.tokenId) (contracts/erc-721/ERC721Lazy.sol#66)
                - (success,returndata) = target.call{value: value}(data) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#119)
        Event emitted after the call(s):
        - Creators(tokenId,_creators) (contracts/erc-721/ERC721Lazy.sol#102)
                - _saveCreators(data.tokenId,data.creators) (contracts/erc-721/ERC721Lazy.sol#68)
        - RoyaltiesSet(id,_royalties) (node_modules/@rarible/royalties/contracts/impl/RoyaltiesV2Impl.sol#16)
                - _saveRoyalties(data.tokenId,data.royalties) (contracts/erc-721/ERC721Lazy.sol#67)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

EIP712Upgradeable._getChainId() (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#93-99) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#96-98)
AddressUpgradeable.isContract(address) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#26-35) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#33)
AddressUpgradeable._verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#147-164) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#154-159)
LibSignature.recover(bytes32,bytes) (node_modules/@rarible/libraries/contracts/LibSignature.sol#20-45) uses assembly
        - INLINE ASM (node_modules/@rarible/libraries/contracts/LibSignature.sol#38-42)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Different versions of Solidity is used:
        - Version used: ['0.7.6', '>=0.4.24<0.8.0', '>=0.6.0<0.8.0', '>=0.6.2<0.8.0', '^0.7.0']
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/introspection/IERC165Upgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#3)
        - >=0.4.24<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/proxy/Initializable.sol#4)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC721/ERC721EnumerableUpgradeable.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC721/ERC721MetadataUpgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC721/ERC721ReceiverUpgradeable.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC721/ERC721Upgradeable.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/EnumerableMapUpgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/EnumerableSetUpgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/StringsUpgradeable.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@rarible/lazy-mint/contracts/erc-721/ERC721LazyMint.sol#3)
```

```
          - v2 (node_modules/@rarible/lazy-mint/contracts/erc-721/IERC721LazyMint.sol#4)
          - >=0.6.2<0.8.0 (node_modules/@rarible/lazy-mint/contracts/erc-721/LibERC721LazyMint.sol#3)
          - ^0.7.0 (node_modules/@rarible/libraries/contracts/LibSignature.sol#3)
          - >=0.6.2<0.8.0 (node_modules/@rarible/royalties-upgradeable/contracts/RoyaltiesV2Upgradeable.sol#3)
          - v2 (node_modules/@rarible/royalties-upgradeable/contracts/RoyaltiesV2Upgradeable.sol#4)
          - >=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/LibPart.sol#3)
          - >=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/LibRoyaltiesV2.sol#3)
          - >=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/RoyaltiesV2.sol#3)
          - v2 (node_modules/@rarible/royalties/contracts/RoyaltiesV2.sol#4)
          - >=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/impl/AbstractRoyalties.sol#3)
          - >=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/impl/RoyaltiesV2Impl.sol#3)
          - v2 (node_modules/@rarible/royalties/contracts/impl/RoyaltiesV2Impl.sol#4)
          - 0.7.6 (contracts/HasContractURI.sol#3)
          - 0.7.6 (contracts/LibURI.sol#3)
          - 0.7.6 (contracts/Mint721Validator.sol#3)
          - 0.7.6 (contracts/erc-1271/ERC1271.sol#3)
          - 0.7.6 (contracts/erc-1271/ERC1271Validator.sol#3)
          - 0.7.6 (contracts/erc-721/ERC721Base.sol#3)
          - v2 (contracts/erc-721/ERC721Base.sol#4)
          - >=0.6.0<0.8.0 (contracts/erc-721/ERC721BurnableUpgradeable.sol#3)
          - 0.7.6 (contracts/erc-721/ERC721DefaultApproval.sol#3)
          - 0.7.6 (contracts/erc-721/ERC721Lazy.sol#3)
          - v2 (contracts/erc-721/ERC721Lazy.sol#4)
          - 0.7.6 (contracts/erc-721/ERC721Rarible.sol#3)
          - v2 (contracts/erc-721/ERC721Rarible.sol#4)
          - >=0.6.0<0.8.0 (contracts/erc-721/ERC721Upgradeable.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

AbstractRoyalties._onRoyaltiesSet(uint256,LibPart.Part[]) (node_modules/@rarible/royalties/contracts/impl/AbstractRoyalties.sol#31) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#79-81) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#104-106) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#129-131) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#139-145) is never used and should be removed
AddressUpgradeable.sendValue(address,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#53-59) is never used and should be removed
ContextUpgradeable.__Context_init() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#17-19) is never used and should be removed
ContextUpgradeable._msgData() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#27-30) is never used and should be removed
EIP712Upgradeable.__EIP712_init(string,string) (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#44-46) is never used and should be removed
ERC1271.returnIsValidSignatureMagicNumber(bool) (contracts/erc-1271/ERC1271.sol#22-24) is never used and should be removed
ERC165Upgradeable.__ERC165_init() (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#25-27) is never used and should be removed
ERC721BurnableUpgradeable.__ERC721Burnable_init() (contracts/erc-721/ERC721BurnableUpgradeable.sol#14-16) is never used and should be removed
ERC721Upgradeable.__ERC721_init(string,string) (contracts/erc-721/ERC721Upgradeable.sol#98-102) is never used and should be removed
ERC721Upgradeable._mint(address,uint256) (contracts/erc-721/ERC721Upgradeable.sol#344-356) is never used and should be removed
EnumerableMapUpgradeable._get(EnumerableMapUpgradeable.Map,bytes32) (node_modules/@openzeppelin/contracts-upgradeable/utils/EnumerableMapUpgradeable.sol#163-167) is never used and should be removed
EnumerableMapUpgradeable._tryGet(EnumerableMapUpgradeable.Map,bytes32) (node_modules/@openzeppelin/contracts-upgradeable/utils/EnumerableMapUpgradeable.sol#150-154) is never used and should be removed
EnumerableMapUpgradeable.get(EnumerableMapUpgradeable.UintToAddressMap,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/EnumerableMapUpgradeable.sol#253-255) is never used and should be removed
EnumerableMapUpgradeable.tryGet(EnumerableMapUpgradeable.UintToAddressMap,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/EnumerableMapUpgradeable.sol#241-244) is never used and should be removed
EnumerableSetUpgradeable.add(EnumerableSetUpgradeable.AddressSet,address) (node_modules/@openzeppelin/contracts-upgradeable/utils/EnumerableSetUpgradeable.sol#201-203) is never used and should be removed
EnumerableSetUpgradeable.add(EnumerableSetUpgradeable.Bytes32Set,bytes32) (node_modules/@openzeppelin/contracts-upgradeable/utils/EnumerableSetUpgradeable.sol#147-149) is never used and should be removed
EnumerableSetUpgradeable.at(EnumerableSetUpgradeable.AddressSet,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/EnumerableSetUpgradeable.sol#239-241) is never used and should be removed
EnumerableSetUpgradeable.at(EnumerableSetUpgradeable.Bytes32Set,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/EnumerableSetUpgradeable.sol#185-187) is never used and should be removed
EnumerableSetUpgradeable.contains(EnumerableSetUpgradeable.AddressSet,address) (node_modules/@openzeppelin/contracts-upgradeable/utils/EnumerableSetUpgradeable.sol#215-220) is never used and should be removed
EnumerableSetUpgradeable.contains(EnumerableSetUpgradeable.Bytes32Set,bytes32) (node_modules/@openzeppelin/contracts-upgradeable/utils/EnumerableSetUpgradeable.sol#164-166) is never used and should be removed
EnumerableSetUpgradeable.contains(EnumerableSetUpgradeable.UintSet,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/EnumerableSetUpgradeable.sol#273-275) is never used and should be removed
EnumerableSetUpgradeable.length(EnumerableSetUpgradeable.AddressSet) (node_modules/@openzeppelin/contracts-upgradeable/utils/EnumerableSetUpgradeable.sol#225-227) is never used and should be removed
EnumerableSetUpgradeable.length(EnumerableSetUpgradeable.Bytes32Set) (node_modules/@openzeppelin/contracts-upgradeable/utils/EnumerableSetUpgradeable.sol#171-173) is never used and should be removed
EnumerableSetUpgradeable.remove(EnumerableSetUpgradeable.AddressSet,address) (node_modules/@openzeppelin/contracts-upgradeable/utils/EnumerableSetUpgradeable.sol#211-213) is never used and should be removed
EnumerableSetUpgradeable.remove(EnumerableSetUpgradeable.Bytes32Set,bytes32) (node_modules/@openzeppelin/contracts-upgradeable/utils/EnumerableSetUpgradeable.sol#157-159) is never used and should be removed
HasContractURI._setContractURI(string) (contracts/HasContractURI.sol#25-27) is never used and should be removed
OwnableUpgradeable.__Ownable_init() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#27-30) is never used and should be removed
SafeMathUpgradeable.div(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#135-138) is never used and should be removed
SafeMathUpgradeable.div(uint256,uint256,string) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#190-193) is never used and should be removed
SafeMathUpgradeable.mod(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#152-155) is never used and should be removed
SafeMathUpgradeable.mod(uint256,uint256,string) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#210-213) is never used and should be removed

SafeMathUpgradeable.mul(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#116-121) is never used and should be removed
SafeMathUpgradeable.sub(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#101-104) is never used and should be removed
SafeMathUpgradeable.sub(uint256,uint256,string) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#170-173) is never used and should be removed
SafeMathUpgradeable.tryAdd(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#24-28) is never used and should be removed
SafeMathUpgradeable.tryDiv(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#60-63) is never used and should be removed
SafeMathUpgradeable.tryMod(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#70-73) is never used and should be removed
SafeMathUpgradeable.tryMul(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#45-53) is never used and should be removed
SafeMathUpgradeable.trySub(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#35-38) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/introspection/IERC165Upgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#3) is too complex
Pragma version>=0.4.24<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/proxy/Initializable.sol#4) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC721/IERC721EnumerableUpgradeable.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC721/IERC721MetadataUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC721/IERC721ReceiverUpgradeable.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC721/IERC721Upgradeable.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/EnumerableMapUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/EnumerableSetUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/StringsUpgradeable.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/lazy-mint/contracts/erc-721/IERC721LazyMint.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/lazy-mint/contracts/erc-721/LibERC721LazyMint.sol#3) is too complex
Pragma version^0.7.0 (node_modules/@rarible/libraries/contracts/LibSignature.sol#3) allows old versions
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties-upgradeable/contracts/RoyaltiesV2Upgradeable.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/LibPart.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/LibRoyaltiesV2.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/RoyaltiesV2.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/impl/AbstractRoyalties.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/impl/RoyaltiesV2Impl.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (contracts/erc-721/ERC721BurnableUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (contracts/erc-721/ERC721Upgradeable.sol#3) is too complex
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

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

Function OwnableUpgradeable.__Ownable_init() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#27-30) is not in mixedCase
Function OwnableUpgradeable.__Ownable_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#32-36) is not in mixedCase
Variable OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#74) is not in mixedCase
Function EIP712Upgradeable.__EIP712_init(string,string) (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#44-46) is not in mixedCase
Function EIP712Upgradeable.__EIP712_init_unchained(string,string) (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#48-53) is not in mixedCase
Function EIP712Upgradeable._EIP712NameHash() (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#107-109) is not in mixedCase
Function EIP712Upgradeable._EIP712VersionHash() (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#117-119) is not in mixedCase
Variable EIP712Upgradeable._HASHED_NAME (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#27) is not in mixedCase
Variable EIP712Upgradeable._HASHED_VERSION (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#28) is not in mixedCase
Variable EIP712Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#120) is not in mixedCase
Function ERC165Upgradeable.__ERC165_init() (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#25-27) is not in mixedCase
Function ERC165Upgradeable.__ERC165_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#29-33) is not in mixedCase
Variable ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59) is not in mixedCase
Function ContextUpgradeable.__Context_init() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#17-19) is not in mixedCase
Function ContextUpgradeable.__Context_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#21-22) is not in mixedCase
Variable ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31) is not in mixedCase
Function RoyaltiesV2Upgradeable.__RoyaltiesV2Upgradeable_init_unchained() (node_modules/@rarible/royalties-upgradeable/contracts/RoyaltiesV2Upgradeable.sol#11-13) is not in mixedCase

Function HasContractURI.__HasContractURI_init_unchained(string) (contracts/HasContractURI.sol#16-19) is not in mixedCase
Parameter HasContractURI.__HasContractURI_init_unchained(string)._contractURI (contracts/HasContractURI.sol#16) is not in mixedCase
Variable HasContractURI.__gap (contracts/HasContractURI.sol#29) is not in mixedCase
Parameter LibURI.checkPrefix(string,string)._tokenURI (contracts/LibURI.sol#7) is not in mixedCase
Function Mint721Validator.__Mint721Validator_init_unchained() (contracts/Mint721Validator.sol#9-11) is not in mixedCase
Variable Mint721Validator.__gap (contracts/Mint721Validator.sol#14) is not in mixedCase
Variable ERC1271Validator.__gap (contracts/erc-1271/ERC1271Validator.sol#34) is not in mixedCase
Variable ERC721Base.__gap (contracts/erc-721/ERC721Base.sol#30) is not in mixedCase
Function ERC721BurnableUpgradeable.__ERC721Burnable_init() (contracts/erc-721/ERC721BurnableUpgradeable.sol#14-18) is not in mixedCase
Function ERC721BurnableUpgradeable.__ERC721Burnable_init_unchained() (contracts/erc-721/ERC721BurnableUpgradeable.sol#20-21) is not in mixedCase
Variable ERC721BurnableUpgradeable.__gap (contracts/erc-721/ERC721BurnableUpgradeable.sol#40) is not in mixedCase
Variable ERC721DefaultApproval.__gap (contracts/erc-721/ERC721DefaultApproval.sol#24) is not in mixedCase
Function ERC721Lazy.__ERC721Lazy_init_unchained() (contracts/erc-721/ERC721Lazy.sol#25-27) is not in mixedCase
Parameter ERC721Lazy.updateAccount(uint256,address,address)._id (contracts/erc-721/ERC721Lazy.sol#105) is not in mixedCase
Parameter ERC721Lazy.updateAccount(uint256,address,address)._from (contracts/erc-721/ERC721Lazy.sol#105) is not in mixedCase
Parameter ERC721Lazy.updateAccount(uint256,address,address)._to (contracts/erc-721/ERC721Lazy.sol#105) is not in mixedCase
Parameter ERC721Lazy.getCreators(uint256)._id (contracts/erc-721/ERC721Lazy.sol#110) is not in mixedCase
Variable ERC721Lazy.__gap (contracts/erc-721/ERC721Lazy.sol#114) is not in mixedCase
Function ERC721Rarible.__ERC721Rarible_init(string,string,string,address,address) (contracts/erc-721/ERC721Rarible.sol#12-15) is not in mixedCase
Parameter ERC721Rarible.__ERC721Rarible_init(string,string,string,string,address,address,address)._name (contracts/erc-721/ERC721Rarible.sol#12) is not in mixedCase
Parameter ERC721Rarible.__ERC721Rarible_init(string,string,string,string,address,address,address)._symbol (contracts/erc-721/ERC721Rarible.sol#12) is not in mixedCase
Function ERC721Rarible.__ERC721Rarible_init_unchained(string,string,string,address,address) (contracts/erc-721/ERC721Rarible.sol#17-32) is not in mixedCase
Parameter ERC721Rarible.__ERC721Rarible_init_unchained(string,string,string,string,address,address,address)._name (contracts/erc-721/ERC721Rarible.sol#17) is not in mixedCase
Parameter ERC721Rarible.__ERC721Rarible_init_unchained(string,string,string,string,address,address,address)._symbol (contracts/erc-721/ERC721Rarible.sol#17) is not in mixedCase
Variable ERC721Rarible.__gap (contracts/erc-721/ERC721Rarible.sol#34) is not in mixedCase
Function ERC721Upgradeable.__ERC721_init(string,string) (contracts/erc-721/ERC721Upgradeable.sol#98-102) is not in mixedCase
Function ERC721Upgradeable.__ERC721_init_unchained(string,string) (contracts/erc-721/ERC721Upgradeable.sol#104-112) is not in mixedCase
Parameter ERC721Upgradeable.safeTransferFrom(address,address,uint256,bytes)._data (contracts/erc-721/ERC721Upgradeable.sol#256) is not in mixedCase
Variable ERC721Upgradeable.__gap (contracts/erc-721/ERC721Upgradeable.sol#495) is not in mixedCase
Variable ERC721Upgradeable._holderTokens (contracts/erc-721/ERC721Upgradeable.sol#35) is not in mixedCase
Variable ERC721Upgradeable._tokensOwners (contracts/erc-721/ERC721Upgradeable.sol#38) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#94)" inEIP712Upgradeable (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#25-121)
Redundant expression "this (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#28)" inContextUpgradeable (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#16-32)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements
```

```
ERC1271.slitherConstructorConstantVariables() (contracts/erc-1271/ERC1271.sol#5-26) uses literals with too many digits:
        - ERC1271_RETURN_INVALID_SIGNATURE = 0x00000000 (contracts/erc-1271/ERC1271.sol#9)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits

ERC1271 (contracts/erc-1271/ERC1271.sol#5-26) does not implement functions:
        - ERC1271.isValidSignature(bytes32,bytes) (contracts/erc-1271/ERC1271.sol#20)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unimplemented-functions

LibERC721LazyMint._INTERFACE_ID_MINT_AND_TRANSFER (node_modules/@rarible/lazy-mint/contracts/erc-721/LibERC721LazyMint.sol#9) is never used in LibERC721LazyMint (node_modules/@rarible/lazy-mint/contracts/erc-721/LibERC721LazyMint.sol#7-3
9)
ERC721Rarible.__gap (contracts/erc-721/ERC721Rarible.sol#34) is never used in ERC721Rarible (contracts/erc-721/ERC721Rarible.sol#8-35)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

renounceOwnership() should be declared external:
        - OwnableUpgradeable.renounceOwnership() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#60-63)
transferOwnership(address) should be declared external:
        - OwnableUpgradeable.transferOwnership(address) (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#69-73)
isValidSignature(bytes32,bytes) should be declared external:
        - ERC1271.isValidSignature(bytes32,bytes) (contracts/erc-1271/ERC1271.sol#20)
burn(uint256) should be declared external:
        - ERC721BurnableUpgradeable.burn(uint256) (contracts/erc-721/ERC721BurnableUpgradeable.sol#29-39)
balanceOf(address) should be declared external:
        - ERC721Upgradeable.balanceOf(address) (contracts/erc-721/ERC721Upgradeable.sol#117-120)
name() should be declared external:
        - ERC721Upgradeable.name() (contracts/erc-721/ERC721Upgradeable.sol#132-134)
symbol() should be declared external:
        - ERC721Upgradeable.symbol() (contracts/erc-721/ERC721Upgradeable.sol#139-141)
tokenURI(uint256) should be declared external:
        - ERC721Upgradeable.tokenURI(uint256) (contracts/erc-721/ERC721Upgradeable.sol#146-162)
tokenOfOwnerByIndex(address,uint256) should be declared external:
        - ERC721Upgradeable.tokenOfOwnerByIndex(address,uint256) (contracts/erc-721/ERC721Upgradeable.sol#176-178)
totalSupply() should be declared external:
        - ERC721Upgradeable.totalSupply() (contracts/erc-721/ERC721Upgradeable.sol#183-186)
tokenByIndex(uint256) should be declared external:
        - ERC721Upgradeable.tokenByIndex(uint256) (contracts/erc-721/ERC721Upgradeable.sol#191-194)
approve(address,uint256) should be declared external:
        - ERC721Upgradeable.approve(address,uint256) (contracts/erc-721/ERC721Upgradeable.sol#199-208)
setApprovalForAll(address,bool) should be declared external:
        - ERC721Upgradeable.setApprovalForAll(address,bool) (contracts/erc-721/ERC721Upgradeable.sol#222-227)
transferFrom(address,address,uint256) should be declared external:
        - ERC721Upgradeable.transferFrom(address,address,uint256) (contracts/erc-721/ERC721Upgradeable.sol#239-244)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

ERC721WoonklyNFTMinimal.sol

```
OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#74) shadows:
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
ERC721UpgradeableMinimal.__gap (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#443) shadows:
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
HasContractURI.__gap (contracts/HasContractURI.sol#38) shadows:
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
Mint721Validator.__gap (contracts/Mint721Validator.sol#16) shadows:
        - ERC1271Validator.__gap (contracts/erc-1271/ERC1271Validator.sol#34)
        - EIP712Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#120)
ERC1271Validator.__gap (contracts/erc-1271/ERC1271Validator.sol#34) shadows:
        - EIP712Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#120)
ERC721BaseMinimal.__gap (contracts/erc-721-minimal/ERC721BaseMinimal.sol#30) shadows:
        - HasContractURI.__gap (contracts/HasContractURI.sol#29)
        - ERC721LazyMinimal.__gap (contracts/erc-721-minimal/ERC721LazyMinimal.sol#112)
        - ERC721URI.__gap (contracts/erc-721-minimal/ERC721URI.sol#92)
        - Mint721Validator.__gap (contracts/Mint721Validator.sol#16)
        - ERC721BurnableUpgradeableMinimal.__gap (contracts/erc-721-minimal/ERC721BurnableUpgradeableMinimal.sol#41)
        - ERC721DefaultApprovalMinimal.__gap (contracts/erc-721-minimal/ERC721DefaultApprovalMinimal.sol#24)
        - ERC721UpgradeableMinimal.__gap (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#443)
        - ERC1271Validator.__gap (contracts/erc-1271/ERC1271Validator.sol#34)
        - EIP712Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#120)
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#74)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
ERC721BurnableUpgradeableMinimal.__gap (contracts/erc-721-minimal/ERC721BurnableUpgradeableMinimal.sol#41) shadows:
        - ERC721UpgradeableMinimal.__gap (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#443)
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
ERC721DefaultApprovalMinimal.__gap (contracts/erc-721-minimal/ERC721DefaultApprovalMinimal.sol#24) shadows:
        - ERC721UpgradeableMinimal.__gap (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#443)
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
ERC721LazyMinimal.__gap (contracts/erc-721-minimal/ERC721LazyMinimal.sol#112) shadows:
        - ERC721URI.__gap (contracts/erc-721-minimal/ERC721URI.sol#92)
        - Mint721Validator.__gap (contracts/Mint721Validator.sol#16)
        - ERC721UpgradeableMinimal.__gap (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#443)
        - ERC1271Validator.__gap (contracts/erc-1271/ERC1271Validator.sol#34)
        - EIP712Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#120)
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
ERC721LazyMinimal._INTERFACE_ID_ERC165 (contracts/erc-721-minimal/ERC721LazyMinimal.sol#17) shadows:
        - ERC165Upgradeable._INTERFACE_ID_ERC165 (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#18)
ERC721LazyMinimal._INTERFACE_ID_ERC721 (contracts/erc-721-minimal/ERC721LazyMinimal.sol#18) shadows:
        - ERC721UpgradeableMinimal._INTERFACE_ID_ERC721 (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#54)
ERC721LazyMinimal._INTERFACE_ID_ERC721_METADATA (contracts/erc-721-minimal/ERC721LazyMinimal.sol#19) shadows:
        - ERC721UpgradeableMinimal._INTERFACE_ID_ERC721_METADATA (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#63)
ERC721RaribleMinimal.__gap (contracts/erc-721-minimal/ERC721RaribleMinimal.sol#57) shadows:
        - ERC721BaseMinimal.__gap (contracts/erc-721-minimal/ERC721BaseMinimal.sol#38)
        - HasContractURI.__gap (contracts/HasContractURI.sol#29)
        - ERC721LazyMinimal.__gap (contracts/erc-721-minimal/ERC721LazyMinimal.sol#112)
        - ERC721URI.__gap (contracts/erc-721-minimal/ERC721URI.sol#92)
        - Mint721Validator.__gap (contracts/Mint721Validator.sol#16)
        - ERC721BurnableUpgradeableMinimal.__gap (contracts/erc-721-minimal/ERC721BurnableUpgradeableMinimal.sol#41)
        - ERC721DefaultApprovalMinimal.__gap (contracts/erc-721-minimal/ERC721DefaultApprovalMinimal.sol#24)
        - ERC721UpgradeableMinimal.__gap (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#443)
        - ERC1271Validator.__gap (contracts/erc-1271/ERC1271Validator.sol#34)
        - EIP712Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#120)
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#74)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
ERC721URI.__gap (contracts/erc-721-minimal/ERC721URI.sol#52) shadows:
        - ERC721UpgradeableMinimal.__gap (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#443)
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variable-shadowing

ERC1271Validator.validate1271(address,bytes32,bytes).signerFromSig (contracts/erc-1271/ERC1271Validator.sol#19) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

ERC721UpgradeableMinimal._checkOnERC721Received(address,address,uint256,bytes) (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#401-422) ignores return value by IERC721ReceiverUpgradeable(to).onERC721R
eceived(_msgSender(),from,tokenId,_data) (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#408-418)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

ERC721BaseMinimal.isApprovedForAll(address,address).owner (contracts/erc-721-minimal/ERC721BaseMinimal.sol#18) shadows:
        - OwnableUpgradeable.owner() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#41-43) (function)
ERC721RaribleMinimal.__ERC721RaribleUser_init(string,string,string,string,address[],address,address).._name (contracts/erc-721-minimal/ERC721RaribleMinimal.sol#15) shadows:
        - ERC721UpgradeableMinimal._name (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#23) (state variable)
ERC721RaribleMinimal.__ERC721RaribleUser_init(string,string,string,string,address[],address,address)._symbol (contracts/erc-721-minimal/ERC721RaribleMinimal.sol#15) shadows:
        - ERC721UpgradeableMinimal._symbol (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#26) (state variable)
ERC721RaribleMinimal.__ERC721RaribleUser_init(string,string,string,string,address[],address,address).baseURI (contracts/erc-721-minimal/ERC721RaribleMinimal.sol#15) shadows:
        - ERC721URI.baseURI() (contracts/erc-721-minimal/ERC721URI.sol#50-52) (function)
ERC721RaribleMinimal.__ERC721RaribleUser_init(string,string,string,string,address[],address,address).contractURI (contracts/erc-721-minimal/ERC721RaribleMinimal.sol#15) shadows:
        - HasContractURI.contractURI (contracts/HasContractURI.sol#9) (state variable)
ERC721RaribleMinimal.__ERC721Rarible_init(string,string,string,string,address,address)._name (contracts/erc-721-minimal/ERC721RaribleMinimal.sol#26) shadows:
        - ERC721UpgradeableMinimal._name (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#23) (state variable)
ERC721RaribleMinimal.__ERC721Rarible_init(string,string,string,string,address,address)._symbol (contracts/erc-721-minimal/ERC721RaribleMinimal.sol#26) shadows:
        - ERC721UpgradeableMinimal._symbol (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#26) (state variable)
ERC721RaribleMinimal.__ERC721Rarible_init(string,string,string,string,address,address).baseURI (contracts/erc-721-minimal/ERC721RaribleMinimal.sol#26) shadows:
        - ERC721URI.baseURI() (contracts/erc-721-minimal/ERC721URI.sol#50-52) (function)
ERC721RaribleMinimal.__ERC721Rarible_init(string,string,string,string,address,address).contractURI (contracts/erc-721-minimal/ERC721RaribleMinimal.sol#26) shadows:
        - HasContractURI.contractURI (contracts/HasContractURI.sol#9) (state variable)
ERC721RaribleMinimal.__ERC721Rarible_init_unchained(string,string,string,string,address,address)._name (contracts/erc-721-minimal/ERC721RaribleMinimal.sol#33) shadows:
        - ERC721UpgradeableMinimal._name (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#23) (state variable)
ERC721RaribleMinimal.__ERC721Rarible_init_unchained(string,string,string,string,address,address)._symbol (contracts/erc-721-minimal/ERC721RaribleMinimal.sol#33) shadows:
        - ERC721UpgradeableMinimal._symbol (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#26) (state variable)
ERC721RaribleMinimal.__ERC721Rarible_init_unchained(string,string,string,string,address,address).baseURI (contracts/erc-721-minimal/ERC721RaribleMinimal.sol#33) shadows:
        - ERC721URI.baseURI() (contracts/erc-721-minimal/ERC721URI.sol#50-52) (function)
ERC721RaribleMinimal.__ERC721Rarible_init_unchained(string,string,string,string,address,address).contractURI (contracts/erc-721-minimal/ERC721RaribleMinimal.sol#33) shadows:
        - HasContractURI.contractURI (contracts/HasContractURI.sol#9) (state variable)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

Variable 'ERC721UpgradeableMinimal._checkOnERC721Received(address,address,uint256,bytes).retval (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#408)' in ERC721UpgradeableMinimal._checkOnERC721Received
(address,address,uint256,bytes) (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#401-422) potentially used before declaration: retval == IERC721ReceiverUpgradeable.onERC721Received.selector (node_modul
es/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#409)
Variable 'ERC721UpgradeableMinimal._checkOnERC721Received(address,address,uint256,bytes).reason (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#410)' in ERC721UpgradeableMinimal._checkOnERC721Received
(address,address,uint256,bytes) (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#401-422) potentially used before declaration: reason.length == 0 (node_modules/@rarible/tokens-minimal/contracts/erc-721
/ERC721UpgradeableMinimal.sol#411)
Variable 'ERC721UpgradeableMinimal._checkOnERC721Received(address,address,uint256,bytes).reason (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#410)' in ERC721UpgradeableMinimal._checkOnERC721Received
(address,address,uint256,bytes) (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#401-422) potentially used before declaration: revert(uint256,uint256)(32 + reason,mload(uint256)(reason)) (node_modules/
@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#415)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables

Reentrancy in ERC721LazyMinimal.mintAndTransfer(LibERC721LazyMint.Mint721Data,address) (contracts/erc-721-minimal/ERC721LazyMinimal.sol#50-70):
        External calls:
        - _safeMint(to,data.tokenId) (contracts/erc-721-minimal/ERC721LazyMinimal.sol#66)
                - IERC721ReceiverUpgradeable(to).onERC721Received(_msgSender(),from,tokenId,_data) (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#408-418)
```

AUTOMATED TESTING

```
        State variables written after the call(s):
        - _setTokenURI(data.tokenId,data.tokenURI) (contracts/erc-721-minimal/ERC721LazyMinimal.sol#69)
            - _tokenURIs[tokenId] = _tokenURI (contracts/erc-721-minimal/ERC721URI.sol#33)
        - _saveCreators(data.tokenId,data.creators) (contracts/erc-721-minimal/ERC721LazyMinimal.sol#68)
            - creatorsOfToken.push(_creators[i]) (contracts/erc-721-minimal/ERC721LazyMinimal.sol#88)
        - _saveRoyalties(data.tokenId,data.royalties) (contracts/erc-721-minimal/ERC721LazyMinimal.sol#67)
            - royalties[id].push(_royalties[i]) (node_modules/@rarible/royalties/contracts/impl/AbstractRoyalties.sol#16)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2

Reentrancy in ERC721LazyMinimal.mintAndTransfer(LibERC721LazyMint.Mint721Data,address) (contracts/erc-721-minimal/ERC721LazyMinimal.sol#50-70):
        External calls:
        - _safeMint(to,data.tokenId) (contracts/erc-721-minimal/ERC721LazyMinimal.sol#66)
            - IERC721ReceiverUpgradeable(to).onERC721Received(_msgSender(),from,tokenId,_data) (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#408-418)
        Event emitted after the call(s):
        - Creators(tokenId,_creators) (contracts/erc-721-minimal/ERC721LazyMinimal.sol#92)
            - _saveCreators(data.tokenId,data.creators) (contracts/erc-721-minimal/ERC721LazyMinimal.sol#68)
        - RoyaltiesSet(id,_royalties) (node_modules/@rarible/royalties/contracts/impl/RoyaltiesV2Impl.sol#16)
            - _saveRoyalties(data.tokenId,data.royalties) (contracts/erc-721-minimal/ERC721LazyMinimal.sol#67)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

EIP712Upgradeable._getChainId() (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#93-99) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#96-98)
AddressUpgradeable.isContract(address) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#26-35) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#33)
AddressUpgradeable._verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#147-164) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#156-159)
LibSignature.recover(bytes32,bytes) (node_modules/@rarible/libraries/contracts/LibSignature.sol#20-45) uses assembly
        - INLINE ASM (node_modules/@rarible/libraries/contracts/LibSignature.sol#38-42)
ERC721UpgradeableMinimal._checkOnERC721Received(address,address,uint256,bytes) (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#401-422) uses assembly
        - INLINE ASM (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#414-416)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Different versions of Solidity is used:
        - Version used: ['0.7.6', '>=0.4.24<0.8.0', '>=0.6.0<0.8.0', '>=0.6.2<0.8.0', '^0.7.0']
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/introspection/IERC165Upgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#3)
        - >=0.4.24<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/proxy/Initializable.sol#4)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC721/ERC721MetadataUpgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC721/ERC721ReceiverUpgradeable.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC721/IERC721Upgradeable.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/StringsUpgradeable.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@rarible/lazy-mint/contracts/erc-721/IERC721LazyMint.sol#3)
        - v2 (node_modules/@rarible/lazy-mint/contracts/erc-721/IERC721LazyMint.sol#4)
        - >=0.6.2<0.8.0 (node_modules/@rarible/lazy-mint/contracts/erc-721/LibERC721LazyMint.sol#3)
        - ^0.7.0 (node_modules/@rarible/libraries/contracts/LibSignature.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@rarible/royalties-upgradeable/contracts/RoyaltiesV2Upgradeable.sol#3)
        - v2 (node_modules/@rarible/royalties-upgradeable/contracts/RoyaltiesV2Upgradeable.sol#4)
        - >=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/LibPart.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/LibRoyaltiesV2.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/RoyaltiesV2.sol#3)
        - v2 (node_modules/@rarible/royalties/contracts/RoyaltiesV2.sol#4)
        - >=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/impl/AbstractRoyalties.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/impl/RoyaltiesV2Impl.sol#3)
        - v2 (node_modules/@rarible/royalties/contracts/impl/RoyaltiesV2Impl.sol#4)
        - 0.7.6 (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#3)
        - 0.7.6 (contracts/HasContractURI.sol#3)
        - 0.7.6 (contracts/LibURI.sol#3)
        - 0.7.6 (contracts/Mint721Validator.sol#3)
        - 0.7.6 (contracts/erc-1271/ERC1271.sol#3)
        - 0.7.6 (contracts/erc-1271/ERC1271Validator.sol#3)
        - 0.7.6 (contracts/erc-721-minimal/ERC721BaseMinimal.sol#3)
        - v2 (contracts/erc-721-minimal/ERC721BaseMinimal.sol#4)
        - >=0.6.0<0.8.0 (contracts/erc-721-minimal/ERC721BurnableUpgradeableMinimal.sol#3)
        - 0.7.6 (contracts/erc-721-minimal/ERC721DefaultApprovalMinimal.sol#3)
        - 0.7.6 (contracts/erc-721-minimal/ERC721LazyMinimal.sol#3)
        - v2 (contracts/erc-721-minimal/ERC721LazyMinimal.sol#4)
        - 0.7.6 (contracts/erc-721-minimal/ERC721RaribleMinimal.sol#3)
        - v2 (contracts/erc-721-minimal/ERC721RaribleMinimal.sol#4)
        - >=0.6.0<0.8.0 (contracts/erc-721-minimal/ERC721URI.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

AbstractRoyalties._onRoyaltiesSet(uint256,LibPart.Part[]) (node_modules/@rarible/royalties/contracts/impl/AbstractRoyalties.sol#31) is never used and should be removed
AddressUpgradeable._verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#147-164) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#79-81) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#89-91) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#104-106) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#114-121) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#129-131) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#139-145) is never used and should be removed
AddressUpgradeable.sendValue(address,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#53-59) is never used and should be removed
ContextUpgradeable.__Context_init() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#17-19) is never used and should be removed
ContextUpgradeable._msgData() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#27-30) is never used and should be removed
EIP712Upgradeable.__EIP712_init(string,string) (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#44-46) is never used and should be removed
ERC1271.returnIsValidSignatureMagicNumber(uint) (contracts/erc-1271/ERC1271.sol#22-24) is never used and should be removed
ERC165Upgradeable.__ERC165_init() (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#25-27) is never used and should be removed
ERC721BurnableUpgradeableMinimal.__ERC721Burnable_init() (contracts/erc-721-minimal/ERC721BurnableUpgradeableMinimal.sol#14-18) is never used and should be removed
ERC721UpgradeableMinimal.__ERC721_init(string,string) (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#71-75) is never used and should be removed
ERC721UpgradeableMinimal._clearMetadata(uint256) (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#347-348) is never used and should be removed
ERC721UpgradeableMinimal._emitMintEvent(address,uint256) (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#310-312) is never used and should be removed
HasContractURI._setContractURI(string) (contracts/HasContractURI.sol#25-27) is never used and should be removed
OwnableUpgradeable.__Ownable_init() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#27-30) is never used and should be removed
SafeMathUpgradeable.div(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#135-138) is never used and should be removed
SafeMathUpgradeable.div(uint256,uint256,string) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#190-193) is never used and should be removed
SafeMathUpgradeable.mod(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#152-155) is never used and should be removed
SafeMathUpgradeable.mod(uint256,uint256,string) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#210-213) is never used and should be removed
SafeMathUpgradeable.mul(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#116-121) is never used and should be removed
SafeMathUpgradeable.sub(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#101-104) is never used and should be removed
SafeMathUpgradeable.sub(uint256,uint256,string) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#170-173) is never used and should be removed
SafeMathUpgradeable.tryAdd(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#24-28) is never used and should be removed
SafeMathUpgradeable.tryDiv(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#60-63) is never used and should be removed
SafeMathUpgradeable.tryMod(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#70-73) is never used and should be removed
SafeMathUpgradeable.tryMul(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#45-53) is never used and should be removed
SafeMathUpgradeable.trySub(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#35-38) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/introspection/IERC165Upgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#3) is too complex
Pragma version>=0.4.24<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/proxy/Initializable.sol#4) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC721/IERC721MetadataUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC721/IERC721ReceiverUpgradeable.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC721/IERC721Upgradeable.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/StringsUpgradeable.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/lazy-mint/contracts/erc-721/IERC721LazyMint.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/lazy-mint/contracts/erc-721/LibERC721LazyMint.sol#3) is too complex
Pragma version^0.7.0 (node_modules/@rarible/libraries/contracts/LibSignature.sol#3) allows old versions
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties-upgradeable/contracts/RoyaltiesV2Upgradeable.sol#3) is too complex

Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/LibPart.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/LibRoyaltiesV2.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/RoyaltiesV2.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/impl/AbstractRoyalties.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/impl/RoyaltiesV2Impl.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (contracts/erc-721-minimal/ERC721BurnableUpgradeableMinimal.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (contracts/erc-721-minimal/ERC721URI.sol#3) is too complex
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

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

Function OwnableUpgradeable.__Ownable_init() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#27-30) is not in mixedCase
Function OwnableUpgradeable.__Ownable_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#32-36) is not in mixedCase
Variable OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#74) is not in mixedCase
Function EIP712Upgradeable.__EIP712_init(string,string) (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#44-46) is not in mixedCase
Function EIP712Upgradeable.__EIP712_init_unchained(string,string) (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#48-53) is not in mixedCase
Function EIP712Upgradeable._EIP712NameHash() (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#107-109) is not in mixedCase
Function EIP712Upgradeable._EIP712VersionHash() (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#117-119) is not in mixedCase
Variable EIP712Upgradeable._HASHED_NAME (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#27) is not in mixedCase
Variable EIP712Upgradeable._HASHED_VERSION (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#28) is not in mixedCase
Variable EIP712Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#120) is not in mixedCase
Function ERC165Upgradeable.__ERC165_init() (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#25-27) is not in mixedCase
Function ERC165Upgradeable.__ERC165_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#29-33) is not in mixedCase
Variable ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59) is not in mixedCase
Function ContextUpgradeable.__Context_init() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#17-19) is not in mixedCase
Function ContextUpgradeable.__Context_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#21-22) is not in mixedCase
Variable ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31) is not in mixedCase
Function RoyaltiesV2Upgradeable.__RoyaltiesV2Upgradeable_init() (node_modules/@rarible/royalties-upgradeable/contracts/RoyaltiesV2Upgradeable.sol#11-13) is not in mixedCase
Function ERC721UpgradeableMinimal.__ERC721_init(string,string) (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#71-75) is not in mixedCase
Function ERC721UpgradeableMinimal.__ERC721_init_unchained(string,string) (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#77-84) is not in mixedCase
Parameter ERC721UpgradeableMinimal._safeTransferFrom(address,address,uint256,bytes)._data (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#196) is not in mixedCase
Variable ERC721UpgradeableMinimal.__gap (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#443) is not in mixedCase
Function HasContractURI._HasContractURI_init_unchained(string) (contracts/HasContractURI.sol#16-19) is not in mixedCase
Parameter HasContractURI._setContractURI(string)._contractURI (contracts/HasContractURI.sol#16) is not in mixedCase
Variable HasContractURI.__gap (contracts/HasContractURI.sol#28) is not in mixedCase
Parameter LibURI.checkPrefix(string,string)._tokenURI (contracts/LibURI.sol#7) is not in mixedCase
Function Mint721Validator._Mint721Validator_init_unchained() (contracts/Mint721Validator.sol#9-11) is not in mixedCase
Variable Mint721Validator.__gap (contracts/Mint721Validator.sol#16) is not in mixedCase
Variable ERC1271Validator.__gap (contracts/erc-1271/ERC1271Validator.sol#34) is not in mixedCase
Variable ERC721BaseMinimal.__gap (contracts/erc-721-minimal/ERC721BaseMinimal.sol#38) is not in mixedCase
Function ERC721BurnableUpgradeableMinimal._ERC721Burnable_init() (contracts/erc-721-minimal/ERC721BurnableUpgradeableMinimal.sol#14-18) is not in mixedCase
Function ERC721BurnableUpgradeableMinimal.__ERC721Burnable_init_unchained() (contracts/erc-721-minimal/ERC721BurnableUpgradeableMinimal.sol#20-21) is not in mixedCase
```

```
Variable ERC721BurnableUpgradeableMinimal.__gap (contracts/erc-721-minimal/ERC721BurnableUpgradeableMinimal.sol#41) is not in mixedCase
Variable ERC721DefaultApprovalMinimal.__gap (contracts/erc-721-minimal/ERC721DefaultApprovalMinimal.sol#24) is not in mixedCase
Function ERC721LazyMinimal._ERC721Lazy_init_unchained() (contracts/erc-721-minimal/ERC721LazyMinimal.sol#25-27) is not in mixedCase
Parameter ERC721LazyMinimal.updateAccount(uint256,address,address)._id (contracts/erc-721-minimal/ERC721LazyMinimal.sol#95) is not in mixedCase
Parameter ERC721LazyMinimal.updateAccount(uint256,address,address)._from (contracts/erc-721-minimal/ERC721LazyMinimal.sol#95) is not in mixedCase
Parameter ERC721LazyMinimal.updateAccount(uint256,address,address)._to (contracts/erc-721-minimal/ERC721LazyMinimal.sol#95) is not in mixedCase
Parameter ERC721LazyMinimal.getCreators(uint256)._id (contracts/erc-721-minimal/ERC721LazyMinimal.sol#100) is not in mixedCase
Variable ERC721LazyMinimal.__gap (contracts/erc-721-minimal/ERC721LazyMinimal.sol#112) is not in mixedCase
Function ERC721RaribleMinimal._ERC721RaribleUser_init(string,string,string,string,address[],address,address) (contracts/erc-721-minimal/ERC721RaribleMinimal.sol#15-24) is not in mixedCase
Parameter ERC721RaribleMinimal._ERC721RaribleUser_init(string,string,string,string,address[],address,address)._name (contracts/erc-721-minimal/ERC721RaribleMinimal.sol#15) is not in mixedCase
Parameter ERC721RaribleMinimal._ERC721RaribleUser_init(string,string,string,string,address[],address,address)._symbol (contracts/erc-721-minimal/ERC721RaribleMinimal.sol#15) is not in mixedCase
Function ERC721RaribleMinimal._ERC721Rarible_init(string,string,string,string,address,address) (contracts/erc-721-minimal/ERC721RaribleMinimal.sol#26-31) is not in mixedCase
Parameter ERC721RaribleMinimal._ERC721Rarible_init(string,string,string,string,address,address)._name (contracts/erc-721-minimal/ERC721RaribleMinimal.sol#26) is not in mixedCase
Parameter ERC721RaribleMinimal._ERC721Rarible_init(string,string,string,string,address,address)._symbol (contracts/erc-721-minimal/ERC721RaribleMinimal.sol#26) is not in mixedCase
Function ERC721RaribleMinimal._ERC721Rarible_init_unchained(string,string,string,string,address,address) (contracts/erc-721-minimal/ERC721RaribleMinimal.sol#33-48) is not in mixedCase
Parameter ERC721RaribleMinimal._ERC721Rarible_init_unchained(string,string,string,string,address,address)._name (contracts/erc-721-minimal/ERC721RaribleMinimal.sol#33) is not in mixedCase
Parameter ERC721RaribleMinimal._ERC721Rarible_init_unchained(string,string,string,string,address,address)._symbol (contracts/erc-721-minimal/ERC721RaribleMinimal.sol#33) is not in mixedCase
Variable ERC721RaribleMinimal.__gap (contracts/erc-721-minimal/ERC721RaribleMinimal.sol#57) is not in mixedCase
Variable ERC721URI.__gap (contracts/erc-721-minimal/ERC721URI.sol#52) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#94)" inEIP712Upgradeable (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#25-121)
Redundant expression "this (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#28)" inContextUpgradeable (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#16-32)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

ERC1271.slitherConstructorConstantVariables() (contracts/erc-1271/ERC1271.sol#5-26) uses literals with too many digits:
        - ERC1271.RETURN_INVALID_SIGNATURE = 0x00000000 (contracts/erc-1271/ERC1271.sol#9)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits

ERC1271 (contracts/erc-1271/ERC1271.sol#5-26) does not implement functions:
        - ERC1271.isValidSignature(bytes32,bytes) (contracts/erc-1271/ERC1271.sol#20)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unimplemented-functions

LibERC721LazyMint._INTERFACE_ID_MINT_AND_TRANSFER (node_modules/@rarible/lazy-mint/contracts/erc-721/LibERC721LazyMint.sol#9) is never used in LibERC721LazyMint (node_modules/@rarible/lazy-mint/contracts/erc-721/LibERC721LazyMint.sol#7-3
9)
ERC721RaribleMinimal.__gap (contracts/erc-721-minimal/ERC721RaribleMinimal.sol#57) is never used in ERC721RaribleMinimal (contracts/erc-721-minimal/ERC721RaribleMinimal.sol#8-58)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

renounceOwnership() should be declared external:
        - OwnableUpgradeable.renounceOwnership() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#60-63)
transferOwnership(address) should be declared external:
        - OwnableUpgradeable.transferOwnership(address) (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#69-73)
balanceOf(address) should be declared external:
        - ERC721UpgradeableMinimal.balanceOf(address) (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#89-92)
name() should be declared external:
        - ERC721UpgradeableMinimal.name() (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#106-108)
symbol() should be declared external:
        - ERC721UpgradeableMinimal.symbol() (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#113-115)
approve(address,uint256) should be declared external:
        - ERC721UpgradeableMinimal.approve(address,uint256) (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#126-136)
transferFrom(address,address,uint256) should be declared external:
        - ERC721UpgradeableMinimal.transferFrom(address,address,uint256) (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#167-176)
isValidSignature(bytes32,bytes) should be declared external:
        - ERC1271.isValidSignature(bytes32,bytes) (contracts/erc-1271/ERC1271.sol#20)
burn(uint256) should be declared external:
        - ERC721BurnableUpgradeableMinimal.burn(uint256) (contracts/erc-721-minimal/ERC721BurnableUpgradeableMinimal.sol#29-39)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

ERC721WoonklyNFTRevealWave.sol

```
OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#74) shadows:
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
HasContractURI.__gap (contracts/HasContractURI.sol#29) shadows:
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
Mint721Validator.__gap (contracts/Mint721Validator.sol#16) shadows:
        - ERC1271Validator.__gap (contracts/erc-1271/ERC1271Validator.sol#34)
        - EIP712Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#120)
ERC1271Validator.__gap (contracts/erc-1271/ERC1271Validator.sol#34) shadows:
        - EIP712Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#120)
ERC721Base.__gap (contracts/erc-721/ERC721Base.sol#30) shadows:
        - HasContractURI.__gap (contracts/HasContractURI.sol#29)
        - ERC721Lazy.__gap (contracts/erc-721/ERC721Lazy.sol#114)
        - Mint721Validator.__gap (contracts/Mint721Validator.sol#16)
        - ERC1271Validator.__gap (contracts/erc-1271/ERC1271Validator.sol#34)
        - EIP712Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#120)
        - ERC721BurnableUpgradeable.__gap (contracts/erc-721/ERC721BurnableUpgradeable.sol#40)
        - ERC721DefaultApproval.__gap (contracts/erc-721/ERC721DefaultApproval.sol#24)
        - ERC721Upgradeable.__gap (contracts/erc-721/ERC721Upgradeable.sol#495)
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#74)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
ERC721BurnableUpgradeable.__gap (contracts/erc-721/ERC721BurnableUpgradeable.sol#40) shadows:
        - ERC721Upgradeable.__gap (contracts/erc-721/ERC721Upgradeable.sol#495)
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
ERC721DefaultApproval.__gap (contracts/erc-721/ERC721DefaultApproval.sol#24) shadows:
        - ERC721Upgradeable.__gap (contracts/erc-721/ERC721Upgradeable.sol#495)
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
ERC721Lazy.__gap (contracts/erc-721/ERC721Lazy.sol#114) shadows:
        - Mint721Validator.__gap (contracts/Mint721Validator.sol#16)
        - ERC1271Validator.__gap (contracts/erc-1271/ERC1271Validator.sol#34)
        - EIP712Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#120)
        - ERC721Upgradeable.__gap (contracts/erc-721/ERC721Upgradeable.sol#495)
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
ERC721Lazy._INTERFACE_ID_ERC165 (contracts/erc-721/ERC721Lazy.sol#17) shadows:
        - ERC165Upgradeable._INTERFACE_ID_ERC165 (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#18)
ERC721Lazy._INTERFACE_ID_ERC721 (contracts/erc-721/ERC721Lazy.sol#18) shadows:
        - ERC721Upgradeable._INTERFACE_ID_ERC721 (contracts/erc-721/ERC721Upgradeable.sol#72)
ERC721Lazy._INTERFACE_ID_ERC721_METADATA (contracts/erc-721/ERC721Lazy.sol#19) shadows:
        - ERC721Upgradeable._INTERFACE_ID_ERC721_METADATA (contracts/erc-721/ERC721Upgradeable.sol#81)
ERC721Lazy._INTERFACE_ID_ERC721_ENUMERABLE (contracts/erc-721/ERC721Lazy.sol#20) shadows:
        - ERC721Upgradeable._INTERFACE_ID_ERC721_ENUMERABLE (contracts/erc-721/ERC721Upgradeable.sol#90)
ERC721RaribleRevealWave.__gap (contracts/erc-721/ERC721RaribleRevealWave.sol#245) shadows:
        - ERC721Base.__gap (contracts/erc-721/ERC721Base.sol#30)
        - HasContractURI.__gap (contracts/HasContractURI.sol#29)
        - ERC721Lazy.__gap (contracts/erc-721/ERC721Lazy.sol#114)
        - Mint721Validator.__gap (contracts/Mint721Validator.sol#16)
        - ERC1271Validator.__gap (contracts/erc-1271/ERC1271Validator.sol#34)
        - EIP712Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#120)
        - ERC721BurnableUpgradeable.__gap (contracts/erc-721/ERC721BurnableUpgradeable.sol#40)
        - ERC721DefaultApproval.__gap (contracts/erc-721/ERC721DefaultApproval.sol#24)
        - ERC721Upgradeable.__gap (contracts/erc-721/ERC721Upgradeable.sol#495)
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#74)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
ERC721Upgradeable.__gap (contracts/erc-721/ERC721Upgradeable.sol#495) shadows:
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variable-shadowing

ERC721RaribleRevealWave.isPrivate (contracts/erc-721/ERC721RaribleRevealWave.sol#24) is never initialized. It is used in:
        - ERC721RaribleRevealWave.mintAndTransfer(LibERC721LazyMint.Mint721Data,address) (contracts/erc-721/ERC721RaribleRevealWave.sol#217-229)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-state-variables

ERC721RaribleRevealWave.tokenURI(uint256)._tokenURI (contracts/erc-721/ERC721RaribleRevealWave.sol#192) is a local variable never initialized
ERC1271Validator.validate1271(address,bytes32,bytes).signerFromSig (contracts/erc-1271/ERC1271Validator.sol#19) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

ERC721Lazy._mint(address,uint256) (contracts/erc-721/ERC721Lazy.sol#72-90) ignores return value by _holderTokens[to].add(tokenId) (contracts/erc-721/ERC721Lazy.sol#79)
ERC721Lazy._mint(address,uint256) (contracts/erc-721/ERC721Lazy.sol#72-90) ignores return value by _tokenOwners.set(tokenId,to) (contracts/erc-721/ERC721Lazy.sol#81)
ERC721Upgradeable._mint(address,uint256) (contracts/erc-721/ERC721Upgradeable.sol#344-356) ignores return value by _holderTokens[to].add(tokenId) (contracts/erc-721/ERC721Upgradeable.sol#351)
ERC721Upgradeable._mint(address,uint256) (contracts/erc-721/ERC721Upgradeable.sol#344-356) ignores return value by _tokenOwners.set(tokenId,to) (contracts/erc-721/ERC721Upgradeable.sol#353)
ERC721Upgradeable._burn(uint256) (contracts/erc-721/ERC721Upgradeable.sol#368-386) ignores return value by _holderTokens[owner].remove(tokenId) (contracts/erc-721/ERC721Upgradeable.sol#381)
ERC721Upgradeable._burn(uint256) (contracts/erc-721/ERC721Upgradeable.sol#368-386) ignores return value by _tokenOwners.remove(tokenId) (contracts/erc-721/ERC721Upgradeable.sol#383)
ERC721Upgradeable._transfer(address,address,uint256) (contracts/erc-721/ERC721Upgradeable.sol#409-424) ignores return value by _holderTokens[from].remove(tokenId) (contracts/erc-721/ERC721Upgradeable.sol#418)
ERC721Upgradeable._transfer(address,address,uint256) (contracts/erc-721/ERC721Upgradeable.sol#409-424) ignores return value by _holderTokens[to].add(tokenId) (contracts/erc-721/ERC721Upgradeable.sol#419)
ERC721Upgradeable._transfer(address,address,uint256) (contracts/erc-721/ERC721Upgradeable.sol#409-424) ignores return value by _tokenOwners.set(tokenId,to) (contracts/erc-721/ERC721Upgradeable.sol#421)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

ERC721Base.isApprovedForAll(address,address).owner (contracts/erc-721/ERC721Base.sol#18) shadows:
        - OwnableUpgradeable.owner() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#41-43) (function)
ERC721RaribleRevealWave.__ERC721RaribleUser_init(string,string,string,string,address[],address,address)._name (contracts/erc-721/ERC721RaribleRevealWave.sol#30) shadows:
        - ERC721Upgradeable._name (contracts/erc-721/ERC721Upgradeable.sol#47) (state variable)
ERC721RaribleRevealWave.__ERC721RaribleUser_init(string,string,string,string,address[],address,address)._symbol (contracts/erc-721/ERC721RaribleRevealWave.sol#31) shadows:
        - ERC721Upgradeable._symbol (contracts/erc-721/ERC721Upgradeable.sol#50) (state variable)
ERC721RaribleRevealWave.__ERC721RaribleUser_init(string,string,string,string,address[],address,address).baseURI (contracts/erc-721/ERC721RaribleRevealWave.sol#32) shadows:
        - ERC721Upgradeable.baseURI() (contracts/erc-721/ERC721Upgradeable.sol#169-171) (function)
ERC721RaribleRevealWave.__ERC721RaribleUser_init(string,string,string,string,address[],address,address).contractURI (contracts/erc-721/ERC721RaribleRevealWave.sol#33) shadows:
        - HasContractURI.contractURI (contracts/HasContractURI.sol#8) (state variable)
ERC721RaribleRevealWave.__ERC721Rarible_init(string,string,string,string,address,address)._name (contracts/erc-721/ERC721RaribleRevealWave.sol#55) shadows:
        - ERC721Upgradeable._name (contracts/erc-721/ERC721Upgradeable.sol#47) (state variable)
ERC721RaribleRevealWave.__ERC721Rarible_init(string,string,string,string,address,address)._symbol (contracts/erc-721/ERC721RaribleRevealWave.sol#56) shadows:
        - ERC721Upgradeable._symbol (contracts/erc-721/ERC721Upgradeable.sol#50) (state variable)
ERC721RaribleRevealWave.__ERC721Rarible_init(string,string,string,string,address,address).baseURI (contracts/erc-721/ERC721RaribleRevealWave.sol#57) shadows:
        - ERC721Upgradeable.baseURI() (contracts/erc-721/ERC721Upgradeable.sol#169-171) (function)
```

AUTOMATED TESTING

```
ERC721RaribleRevealWave.__ERC721RaribleUser_init(string,string,string,string,address[],address,address).contractURI (contracts/erc-721/ERC721RaribleRevealWave.sol#33) shadows:
        - HasContractURI.contractURI (contracts/HasContractURI.sol#9) (state variable)
ERC721RaribleRevealWave.__ERC721Rarible_init(string,string,string,string,address,address)._name (contracts/erc-721/ERC721RaribleRevealWave.sol#55) shadows:
        - ERC721Upgradeable._name (contracts/erc-721/ERC721Upgradeable.sol#47) (state variable)
ERC721RaribleRevealWave.__ERC721Rarible_init(string,string,string,string,address,address)._symbol (contracts/erc-721/ERC721RaribleRevealWave.sol#56) shadows:
        - ERC721Upgradeable._symbol (contracts/erc-721/ERC721Upgradeable.sol#50) (state variable)
ERC721RaribleRevealWave.__ERC721Rarible_init(string,string,string,string,address,address).baseURI (contracts/erc-721/ERC721RaribleRevealWave.sol#57) shadows:
        - ERC721Upgradeable.baseURI() (contracts/erc-721/ERC721Upgradeable.sol#169-171) (function)
ERC721RaribleRevealWave.__ERC721Rarible_init(string,string,string,string,address,address).contractURI (contracts/erc-721/ERC721RaribleRevealWave.sol#58) shadows:
        - HasContractURI.contractURI (contracts/HasContractURI.sol#9) (state variable)
ERC721RaribleRevealWave.__ERC721Rarible_init_unchained(string,string,string,string,address,address)._name (contracts/erc-721/ERC721RaribleRevealWave.sol#74) shadows:
        - ERC721Upgradeable._name (contracts/erc-721/ERC721Upgradeable.sol#47) (state variable)
ERC721RaribleRevealWave.__ERC721Rarible_init_unchained(string,string,string,string,address,address)._symbol (contracts/erc-721/ERC721RaribleRevealWave.sol#75) shadows:
        - ERC721Upgradeable._symbol (contracts/erc-721/ERC721Upgradeable.sol#50) (state variable)
ERC721RaribleRevealWave.__ERC721Rarible_init_unchained(string,string,string,string,address,address).baseURI (contracts/erc-721/ERC721RaribleRevealWave.sol#76) shadows:
        - ERC721Upgradeable.baseURI() (contracts/erc-721/ERC721Upgradeable.sol#169-171) (function)
ERC721RaribleRevealWave.__ERC721Rarible_init_unchained(string,string,string,string,address,address).contractURI (contracts/erc-721/ERC721RaribleRevealWave.sol#77) shadows:
        - HasContractURI.contractURI (contracts/HasContractURI.sol#9) (state variable)
ERC721RaribleRevealWave.setRevealWave(uint256,string,string,bool,uint256)._name (contracts/erc-721/ERC721RaribleRevealWave.sol#106) shadows:
        - ERC721Upgradeable._name (contracts/erc-721/ERC721Upgradeable.sol#47) (state variable)
ERC721RaribleRevealWave.changeHiddenBaseURI(uint256,string)._baseURI (contracts/erc-721/ERC721RaribleRevealWave.sol#121) shadows:
        - ERC721Upgradeable._baseURI (contracts/erc-721/ERC721Upgradeable.sol#56) (state variable)
ERC721RaribleRevealWave.changeName(uint256,string)._name (contracts/erc-721/ERC721RaribleRevealWave.sol#166) shadows:
        - ERC721Upgradeable._name (contracts/erc-721/ERC721Upgradeable.sol#47) (state variable)
ERC721RaribleRevealWave.tokenURI(uint256)._baseURI (contracts/erc-721/ERC721RaribleRevealWave.sol#193) shadows:
        - ERC721Upgradeable._baseURI (contracts/erc-721/ERC721Upgradeable.sol#56) (state variable)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

Reentrancy in ERC721Lazy.mintAndTransfer(LibERC721LazyMint.Mint721Data,address) (contracts/erc-721/ERC721Lazy.sol#50-70):
        External calls:
        - _safeMint(to,data.tokenId) (contracts/erc-721/ERC721Lazy.sol#66)
                - returndata = to.functionCall(abi.encodeWithSelector(IERC721ReceiverUpgradeable(to).onERC721Received.selector,_msgSender(),from,tokenId,_data),ERC721: transfer to non ERC721Receiver implementer) (contracts/erc-721/ERC721
Upgradeable.sol#463-469)
                - (success,returndata) = target.call{value: value}(data) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#119)
        External calls sending eth:
        - _safeMint(to,data.tokenId) (contracts/erc-721/ERC721Lazy.sol#66)
                - (success,returndata) = target.call{value: value}(data) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#119)
        State variables written after the call(s):
        - _setTokenURI(data.tokenId,data.tokenURI) (contracts/erc-721/ERC721Lazy.sol#69)
                - _tokenURIs[tokenId] = tokenURI (contracts/erc-721/ERC721Upgradeable.sol#435)
        - _saveCreators(data.tokenId,data.creators) (contracts/erc-721/ERC721Lazy.sol#68)
                - creatorsOfToken.push(_creators[i]) (contracts/erc-721/ERC721Lazy.sol#98)
        - _saveRoyalties(data.tokenId,data.royalties) (contracts/erc-721/ERC721Lazy.sol#67)
                - royalties[id].push(_royalties[i]) (node_modules/@rarible/royalties/contracts/impl/AbstractRoyalties.sol#16)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2

Reentrancy in ERC721Lazy.mintAndTransfer(LibERC721LazyMint.Mint721Data,address) (contracts/erc-721/ERC721Lazy.sol#50-70):
        External calls:
        - _safeMint(to,data.tokenId) (contracts/erc-721/ERC721Lazy.sol#66)
                - returndata = to.functionCall(abi.encodeWithSelector(IERC721ReceiverUpgradeable(to).onERC721Received.selector,_msgSender(),from,tokenId,_data),ERC721: transfer to non ERC721Receiver implementer) (contracts/erc-721/ERC721
Upgradeable.sol#463-469)
                - (success,returndata) = target.call{value: value}(data) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#119)
        External calls sending eth:
        - _safeMint(to,data.tokenId) (contracts/erc-721/ERC721Lazy.sol#66)
                - (success,returndata) = target.call{value: value}(data) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#119)
        Event emitted after the call(s):
        - Creators(tokenId,_creators) (contracts/erc-721/ERC721Lazy.sol#102)
                - _saveCreators(data.tokenId,data.creators) (contracts/erc-721/ERC721Lazy.sol#68)
        - RoyaltiesSet(id,_royalties) (node_modules/@rarible/royalties/contracts/impl/RoyaltiesV2Impl.sol#16)
                - _saveRoyalties(data.tokenId,data.royalties) (contracts/erc-721/ERC721Lazy.sol#67)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

EIP712Upgradeable._getChainId() (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#93-99) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#96-98)
AddressUpgradeable.isContract(address) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#26-35) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#33)
AddressUpgradeable._verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#147-164) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#156-159)
LibSignature.recover(bytes32,bytes) (node_modules/@rarible/libraries/contracts/LibSignature.sol#20-45) uses assembly
        - INLINE ASM (node_modules/@rarible/libraries/contracts/LibSignature.sol#38-42)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

ERC721RaribleRevealWave.tokenURI(uint256) (contracts/erc-721/ERC721RaribleRevealWave.sol#186-214) compares to a boolean constant:
        -revealWave.isRevealed == false (contracts/erc-721/ERC721RaribleRevealWave.sol#200)
ERC721RaribleRevealWave.tokenURI(uint256) (contracts/erc-721/ERC721RaribleRevealWave.sol#186-214) compares to a boolean constant:
        -revealWave.isRevealed == true || revealWave.addTokenNftToRandomBaseURI == true (contracts/erc-721/ERC721RaribleRevealWave.sol#195)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#boolean-equality

Different versions of Solidity is used:
        - Version used: ['0.7.6', '>=0.4.24<0.8.0', '>=0.6.0<0.8.0', '>=0.6.2<0.8.0', '^0.7.0']
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/introspection/IERC165Upgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/introspection/IERC165Upgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#3)
        - >=0.4.24<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/proxy/Initializable.sol#4)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC721/IERC721EnumerableUpgradeable.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC721/IERC721MetadataUpgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC721/IERC721ReceiverUpgradeable.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC721/IERC721Upgradeable.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/EnumerableMapUpgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/EnumerableSetUpgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/StringsUpgradeable.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@rarible/lazy-mint/contracts/erc-721/ERC721LazyMint.sol#3)
        - v2 (node_modules/@rarible/lazy-mint/contracts/erc-721/IERC721LazyMint.sol#4)
        - >=0.6.2<0.8.0 (node_modules/@rarible/lazy-mint/contracts/erc-721/LibERC721LazyMint.sol#3)
        - ^0.7.0 (node_modules/@rarible/libraries/contracts/LibSignature.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@rarible/royalties-upgradeable/contracts/RoyaltiesV2Upgradeable.sol#3)
        - v2 (node_modules/@rarible/royalties-upgradeable/contracts/RoyaltiesV2Upgradeable.sol#4)
        - >=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/LibPart.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/LibRoyaltiesV2.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/RoyaltiesV2.sol#3)
        - v2 (node_modules/@rarible/royalties/contracts/RoyaltiesV2.sol#4)
        - >=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/impl/AbstractRoyalties.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/impl/RoyaltiesV2Impl.sol#3)
        - v2 (node_modules/@rarible/royalties/contracts/impl/RoyaltiesV2Impl.sol#4)
        - 0.7.6 (contracts/HasContractURI.sol#3)
        - 0.7.6 (contracts/LibURI.sol#3)
        - 0.7.6 (contracts/Mint721Validator.sol#3)
        - 0.7.6 (contracts/erc-1271/ERC1271.sol#3)
        - 0.7.6 (contracts/erc-1271/ERC1271Validator.sol#3)
        - 0.7.6 (contracts/erc-721/ERC721Base.sol#3)
        - v2 (contracts/erc-721/ERC721Base.sol#4)
        - >=0.6.0<0.8.0 (contracts/erc-721/ERC721BurnableUpgradeable.sol#3)
        - 0.7.6 (contracts/erc-721/ERC721DefaultApprove1.sol#3)
        - 0.7.6 (contracts/erc-721/ERC721Lazy.sol#3)
        - v2 (contracts/erc-721/ERC721Lazy.sol#4)
        - 0.7.6 (contracts/erc-721/ERC721RaribleRevealWave.sol#3)
        - v2 (contracts/erc-721/ERC721RaribleRevealWave.sol#4)
        - >=0.6.0<0.8.0 (contracts/erc-721/ERC721Upgradeable.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

AbstractRoyalties._onRoyaltiesSet(uint256,LibPart.Part[]) (node_modules/@rarible/royalties/contracts/impl/AbstractRoyalties.sol#31) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#79-81) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#104-106) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#128-131) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#139-145) is never used and should be removed
AddressUpgradeable.sendValue(address,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#53-59) is never used and should be removed
ContextUpgradeable._msgData() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#17-19) is never used and should be removed
ContextUpgradeable._msgData() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#27-30) is never used and should be removed
EIP712Upgradeable.__EIP712_init(string,string) (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#44-46) is never used and should be removed
ERC1271.returnIsValidSignatureMagicNumber(bool) (contracts/erc-1271/ERC1271.sol#22-24) is never used and should be removed
ERC165Upgradeable.__ERC165_init() (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#25-27) is never used and should be removed
ERC721BurnableUpgradeable.__ERC721Burnable_init() (contracts/erc-721/ERC721BurnableUpgradeable.sol#14-18) is never used and should be removed
ERC721Upgradeable.__ERC721_init(string,string) (contracts/erc-721/ERC721Upgradeable.sol#89-102) is never used and should be removed
ERC721Upgradeable._mint(address,uint256) (contracts/erc-721/ERC721Upgradeable.sol#344-356) is never used and should be removed
EnumerableMapUpgradeable._get(EnumerableMapUpgradeable.Map,bytes32) (node_modules/@openzeppelin/contracts-upgradeable/utils/EnumerableMapUpgradeable.sol#163-167) is never used and should be removed
EnumerableMapUpgradeable._tryGet(EnumerableMapUpgradeable.Map,bytes32) (node_modules/@openzeppelin/contracts-upgradeable/utils/EnumerableMapUpgradeable.sol#150-154) is never used and should be removed
EnumerableMapUpgradeable.get(EnumerableMapUpgradeable.UintToAddressMap,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/EnumerableMapUpgradeable.sol#253-255) is never used and should be removed
EnumerableMapUpgradeable.tryGet(EnumerableMapUpgradeable.UintToAddressMap,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/EnumerableMapUpgradeable.sol#241-244) is never used and should be removed
EnumerableSetUpgradeable.add(EnumerableSetUpgradeable.AddressSet,address) (node_modules/@openzeppelin/contracts-upgradeable/utils/EnumerableSetUpgradeable.sol#201-203) is never used and should be removed
EnumerableSetUpgradeable.add(EnumerableSetUpgradeable.Bytes32Set,bytes32) (node_modules/@openzeppelin/contracts-upgradeable/utils/EnumerableSetUpgradeable.sol#147-149) is never used and should be removed
EnumerableSetUpgradeable.at(EnumerableSetUpgradeable.AddressSet,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/EnumerableSetUpgradeable.sol#239-241) is never used and should be removed
EnumerableSetUpgradeable.at(EnumerableSetUpgradeable.Bytes32Set,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/EnumerableSetUpgradeable.sol#185-187) is never used and should be removed
EnumerableSetUpgradeable.contains(EnumerableSetUpgradeable.AddressSet,address) (node_modules/@openzeppelin/contracts-upgradeable/utils/EnumerableSetUpgradeable.sol#218-220) is never used and should be removed
EnumerableSetUpgradeable.contains(EnumerableSetUpgradeable.Bytes32Set,bytes32) (node_modules/@openzeppelin/contracts-upgradeable/utils/EnumerableSetUpgradeable.sol#164-166) is never used and should be removed
EnumerableSetUpgradeable.contains(EnumerableSetUpgradeable.UintSet,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/EnumerableSetUpgradeable.sol#273-275) is never used and should be removed
EnumerableSetUpgradeable.length(EnumerableSetUpgradeable.AddressSet) (node_modules/@openzeppelin/contracts-upgradeable/utils/EnumerableSetUpgradeable.sol#225-227) is never used and should be removed
EnumerableSetUpgradeable.length(EnumerableSetUpgradeable.Bytes32Set) (node_modules/@openzeppelin/contracts-upgradeable/utils/EnumerableSetUpgradeable.sol#171-173) is never used and should be removed
EnumerableSetUpgradeable.remove(EnumerableSetUpgradeable.AddressSet,address) (node_modules/@openzeppelin/contracts-upgradeable/utils/EnumerableSetUpgradeable.sol#211-213) is never used and should be removed
EnumerableSetUpgradeable.remove(EnumerableSetUpgradeable.Bytes32Set,bytes32) (node_modules/@openzeppelin/contracts-upgradeable/utils/EnumerableSetUpgradeable.sol#157-159) is never used and should be removed
HasContractURI._setContractURI(string) (contracts/HasContractURI.sol#25-27) is never used and should be removed
OwnableUpgradeable.__Ownable_init() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#27-30) is never used and should be removed
SafeMathUpgradeable.div(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#135-138) is never used and should be removed
SafeMathUpgradeable.div(uint256,uint256,string) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#190-193) is never used and should be removed
SafeMathUpgradeable.mod(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#152-155) is never used and should be removed
SafeMathUpgradeable.mod(uint256,uint256,string) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#210-213) is never used and should be removed
SafeMathUpgradeable.mul(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#116-121) is never used and should be removed
SafeMathUpgradeable.sub(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#101-104) is never used and should be removed
SafeMathUpgradeable.sub(uint256,uint256,string) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#170-173) is never used and should be removed
SafeMathUpgradeable.tryAdd(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#24-28) is never used and should be removed
SafeMathUpgradeable.tryDiv(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#60-63) is never used and should be removed
SafeMathUpgradeable.tryMod(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#70-73) is never used and should be removed
SafeMathUpgradeable.tryMul(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#45-53) is never used and should be removed
SafeMathUpgradeable.trySub(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#35-38) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/introspection/IERC165Upgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#3) is too complex
Pragma version>=0.4.24<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/proxy/Initializable.sol#4) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC721/IERC721EnumerableUpgradeable.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC721/IERC721MetadataUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC721/IERC721ReceiverUpgradeable.sol#3) is too complex
```

AUTOMATED TESTING

```
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC721/IERC721Upgradeable.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/EnumerableMapUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/EnumerableSetUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/StringsUpgradeable.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/lazy-mint/contracts/erc-721/IERC721LazyMint.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/lazy-mint/contracts/erc-721/LibERC721LazyMint.sol#3) is too complex
Pragma version^0.7.0 (node_modules/@rarible/libraries/contracts/LibSignature.sol#3) allows old versions
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties-upgradeable/contracts/RoyaltiesV2Upgradeable.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/LibPart.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/LibRoyaltiesV2.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/RoyaltiesV2.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/impl/AbstractRoyalties.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/impl/RoyaltiesV2Impl.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (contracts/erc-721/ERC721BurnableUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (contracts/erc-721/ERC721Upgradeable.sol#3) is too complex
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

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

Function OwnableUpgradeable.__Ownable_init() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#27-30) is not in mixedCase
Function OwnableUpgradeable.__Ownable_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#32-36) is not in mixedCase
Variable OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#74) is not in mixedCase
Function EIP712Upgradeable.__EIP712_init(string,string) (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#44-46) is not in mixedCase
Function EIP712Upgradeable.__EIP712_init_unchained(string,string) (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#48-53) is not in mixedCase
Function EIP712Upgradeable._EIP712NameHash() (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#107-109) is not in mixedCase
Function EIP712Upgradeable._EIP712VersionHash() (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#117-119) is not in mixedCase
Variable EIP712Upgradeable._HASHED_NAME (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#27) is not in mixedCase
Variable EIP712Upgradeable._HASHED_VERSION (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#28) is not in mixedCase
Variable EIP712Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#120) is not in mixedCase
Function ERC165Upgradeable.__ERC165_init() (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#25-27) is not in mixedCase
Function ERC165Upgradeable.__ERC165_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#29-33) is not in mixedCase
Variable ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59) is not in mixedCase
Function ContextUpgradeable.__Context_init() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#17-19) is not in mixedCase
Function ContextUpgradeable.__Context_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#21-22) is not in mixedCase
Variable ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31) is not in mixedCase
Function RoyaltiesV2Upgradeable.__RoyaltiesV2Upgradeable_init_unchained() (node_modules/@rarible/royalties-upgradeable/contracts/RoyaltiesV2Upgradeable.sol#11-13) is not in mixedCase
Function HasContractURI.__HasContractURI_init_unchained(string) (contracts/HasContractURI.sol#16-19) is not in mixedCase
Parameter HasContractURI.__HasContractURI_init_unchained(string)._contractURI (contracts/HasContractURI.sol#16) is not in mixedCase
Variable HasContractURI.__gap (contracts/HasContractURI.sol#29) is not in mixedCase
Parameter LibURI.checkPrefix(string,string)._tokenURI (contracts/LibURI.sol#7) is not in mixedCase
Function Mint721Validator.__Mint721Validator_init_unchained() (contracts/Mint721Validator.sol#9-11) is not in mixedCase
Variable Mint721Validator.__gap (contracts/Mint721Validator.sol#16) is not in mixedCase
Variable ERC1271Validator.__gap (contracts/erc-1271/ERC1271Validator.sol#36) is not in mixedCase
Variable ERC721Base.__gap (contracts/erc-721/ERC721Base.sol#30) is not in mixedCase
Function ERC721BurnableUpgradeable.__ERC721Burnable_init() (contracts/erc-721/ERC721BurnableUpgradeable.sol#14-18) is not in mixedCase
Function ERC721BurnableUpgradeable.__ERC721Burnable_init_unchained() (contracts/erc-721/ERC721BurnableUpgradeable.sol#20-21) is not in mixedCase
Variable ERC721BurnableUpgradeable.__gap (contracts/erc-721/ERC721BurnableUpgradeable.sol#40) is not in mixedCase
Variable ERC721DefaultApproval.__gap (contracts/erc-721/ERC721DefaultApproval.sol#24) is not in mixedCase
Function ERC721Lazy.__ERC721Lazy_init_unchained() (contracts/erc-721/ERC721Lazy.sol#25-27) is not in mixedCase
Parameter ERC721Lazy.updateAccount(uint256,address,address)._id (contracts/erc-721/ERC721Lazy.sol#105) is not in mixedCase
Parameter ERC721Lazy.updateAccount(uint256,address,address)._from (contracts/erc-721/ERC721Lazy.sol#105) is not in mixedCase
Parameter ERC721Lazy.updateAccount(uint256,address,address)._to (contracts/erc-721/ERC721Lazy.sol#105) is not in mixedCase
Parameter ERC721Lazy.getCreators(uint256)._id (contracts/erc-721/ERC721Lazy.sol#110) is not in mixedCase
Variable ERC721Lazy.__gap (contracts/erc-721/ERC721Lazy.sol#114) is not in mixedCase
Function ERC721RaribleRevealWave.__ERC721RaribleUser_init(string,string,string,string,address[],address,address) (contracts/erc-721/ERC721RaribleRevealWave.sol#29-52) is not in mixedCase
Parameter ERC721RaribleRevealWave.__ERC721RaribleUser_init(string,string,string,string,address[],address,address)._name (contracts/erc-721/ERC721RaribleRevealWave.sol#30) is not in mixedCase
Parameter ERC721RaribleRevealWave.__ERC721RaribleUser_init(string,string,string,string,address[],address,address)._symbol (contracts/erc-721/ERC721RaribleRevealWave.sol#31) is not in mixedCase
Function ERC721RaribleRevealWave.__ERC721Rarible_init(string,string,string,string,address,address) (contracts/erc-721/ERC721RaribleRevealWave.sol#54-71) is not in mixedCase
Parameter ERC721RaribleRevealWave.__ERC721Rarible_init(string,string,string,string,address,address)._name (contracts/erc-721/ERC721RaribleRevealWave.sol#55) is not in mixedCase
Parameter ERC721RaribleRevealWave.__ERC721Rarible_init(string,string,string,string,address,address)._symbol (contracts/erc-721/ERC721RaribleRevealWave.sol#56) is not in mixedCase
Function ERC721RaribleRevealWave.__ERC721Rarible_init_unchained(string,string,string,string,address,address) (contracts/erc-721/ERC721RaribleRevealWave.sol#73-96) is not in mixedCase
Parameter ERC721RaribleRevealWave.__ERC721Rarible_init_unchained(string,string,string,string,address,address)._name (contracts/erc-721/ERC721RaribleRevealWave.sol#74) is not in mixedCase
Parameter ERC721RaribleRevealWave.__ERC721Rarible_init_unchained(string,string,string,string,address,address)._symbol (contracts/erc-721/ERC721RaribleRevealWave.sol#75) is not in mixedCase
Parameter ERC721RaribleRevealWave.getRevealWaveIdByTokenId(uint256)._tokenId (contracts/erc-721/ERC721RaribleRevealWave.sol#98) is not in mixedCase
Parameter ERC721RaribleRevealWave.setRevealWave(uint256,string,bool,uint256)._waveId (contracts/erc-721/ERC721RaribleRevealWave.sol#105) is not in mixedCase
Parameter ERC721RaribleRevealWave.setRevealWave(uint256,string,bool,uint256)._name (contracts/erc-721/ERC721RaribleRevealWave.sol#106) is not in mixedCase
Parameter ERC721RaribleRevealWave.setRevealWave(uint256,string,bool,uint256)._hiddenBaseURI (contracts/erc-721/ERC721RaribleRevealWave.sol#107) is not in mixedCase
Parameter ERC721RaribleRevealWave.setRevealWave(uint256,string,bool,uint256)._addTokenURIToHiddenBaseURI (contracts/erc-721/ERC721RaribleRevealWave.sol#108) is not in mixedCase
Parameter ERC721RaribleRevealWave.setRevealWave(uint256,string,bool,uint256)._revealDate (contracts/erc-721/ERC721RaribleRevealWave.sol#109) is not in mixedCase
Parameter ERC721RaribleRevealWave.changeHiddenBaseURI(uint256,string)._waveId (contracts/erc-721/ERC721RaribleRevealWave.sol#120) is not in mixedCase
Parameter ERC721RaribleRevealWave.changeHiddenBaseURI(uint256,string)._baseURI (contracts/erc-721/ERC721RaribleRevealWave.sol#121) is not in mixedCase
Parameter ERC721RaribleRevealWave.changeAddTokenURIToHiddenBaseURI(uint256,bool)._waveId (contracts/erc-721/ERC721RaribleRevealWave.sol#129) is not in mixedCase
Parameter ERC721RaribleRevealWave.changeAddTokenURIToHiddenBaseURI(uint256,bool)._addTokenURIToHiddenBaseURI (contracts/erc-721/ERC721RaribleRevealWave.sol#130) is not in mixedCase
Parameter ERC721RaribleRevealWave.changeRevealBaseURI(uint256,string)._waveId (contracts/erc-721/ERC721RaribleRevealWave.sol#137) is not in mixedCase
Parameter ERC721RaribleRevealWave.changeRevealBaseURI(uint256,string)._revealBaseURI (contracts/erc-721/ERC721RaribleRevealWave.sol#138) is not in mixedCase
Parameter ERC721RaribleRevealWave.changeRevealDate(uint256,uint256)._waveId (contracts/erc-721/ERC721RaribleRevealWave.sol#148) is not in mixedCase
Parameter ERC721RaribleRevealWave.changeRevealDate(uint256,uint256)._revealDate (contracts/erc-721/ERC721RaribleRevealWave.sol#149) is not in mixedCase
Parameter ERC721RaribleRevealWave.resetIsRevealedAndRevealURI(uint256)._waveId (contracts/erc-721/ERC721RaribleRevealWave.sol#156) is not in mixedCase
Parameter ERC721RaribleRevealWave.changeName(uint256,string)._waveId (contracts/erc-721/ERC721RaribleRevealWave.sol#165) is not in mixedCase
Parameter ERC721RaribleRevealWave.changeName(uint256,string)._name (contracts/erc-721/ERC721RaribleRevealWave.sol#166) is not in mixedCase
Parameter ERC721RaribleRevealWave.assignRevealWaveIdToToken(uint256,uint256)._tokenId (contracts/erc-721/ERC721RaribleRevealWave.sol#174) is not in mixedCase
Parameter ERC721RaribleRevealWave.assignRevealWaveIdToToken(uint256,uint256)._waveId (contracts/erc-721/ERC721RaribleRevealWave.sol#174) is not in mixedCase
Parameter ERC721RaribleRevealWave.tokenURI(uint256)._tokenId (contracts/erc-721/ERC721RaribleRevealWave.sol#186) is not in mixedCase
Parameter ERC721RaribleRevealWave.mintAndTransferReveal(LibERC721LazyMint.Mint721Data,address,uint256)._waveId (contracts/erc-721/ERC721RaribleRevealWave.sol#236) is not in mixedCase
Variable ERC721RaribleRevealWave.__gap (contracts/erc-721/ERC721RaribleRevealWave.sol#245) is not in mixedCase
Function ERC721Upgradeable.__ERC721_init(string,string) (contracts/erc-721/ERC721Upgradeable.sol#98-102) is not in mixedCase
Function ERC721Upgradeable.__ERC721_init_unchained(string,string) (contracts/erc-721/ERC721Upgradeable.sol#104-112) is not in mixedCase
Parameter ERC721Upgradeable.safeTransferFrom(address,address,uint256,bytes)._data (contracts/erc-721/ERC721Upgradeable.sol#256) is not in mixedCase
Variable ERC721Upgradeable.__gap (contracts/erc-721/ERC721Upgradeable.sol#495) is not in mixedCase
Variable ERC721Upgradeable._holderTokens (contracts/erc-721/ERC721Upgradeable.sol#85) is not in mixedCase
Variable ERC721Upgradeable._tokenOwners (contracts/erc-721/ERC721Upgradeable.sol#38) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#94)" inEIP712Upgradeable (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#25-121)
Redundant expression "this (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#28)" inContextUpgradeable (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#16-32)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

ERC1271.slitherConstructorConstantVariables() (contracts/erc-1271/ERC1271.sol#5-26) uses literals with too many digits:
        - ERC1271_RETURN_INVALID_SIGNATURE = 0x00000000 (contracts/erc-1271/ERC1271.sol#9)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits

ERC1271 (contracts/erc-1271/ERC1271.sol#5-26) does not implement functions:
        - ERC1271.isValidSignature(bytes32,bytes) (contracts/erc-1271/ERC1271.sol#20)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unimplemented-functions

LibERC721LazyMint._INTERFACE_ID_MINT_AND_TRANSFER (node_modules/@rarible/lazy-mint/contracts/erc-721/LibERC721LazyMint.sol#9) is never used in LibERC721LazyMint (node_modules/@rarible/lazy-mint/contracts/erc-721/LibERC721LazyMint.sol#7-3
9)
ERC721RaribleRevealWave.__gap (contracts/erc-721/ERC721RaribleRevealWave.sol#245) is never used in ERC721RaribleRevealWave (contracts/erc-721/ERC721RaribleRevealWave.sol#8-246)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

ERC721RaribleRevealWave.isPrivate (contracts/erc-721/ERC721RaribleRevealWave.sol#24) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant

renounceOwnership() should be declared external:
        - OwnableUpgradeable.renounceOwnership() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#60-63)
transferOwnership(address) should be declared external:
        - OwnableUpgradeable.transferOwnership(address) (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#69-73)
isValidSignature(bytes32,bytes) should be declared external:
        - ERC1271.isValidSignature(bytes32,bytes) (contracts/erc-1271/ERC1271.sol#20)
burn(uint256) should be declared external:
        - ERC721BurnableUpgradeable.burn(uint256) (contracts/erc-721/ERC721BurnableUpgradeable.sol#29-39)
setRevealWave(uint256,string,string,bool,uint256) should be declared external:
        - ERC721RaribleRevealWave.setRevealWave(uint256,string,string,bool,uint256) (contracts/erc-721/ERC721RaribleRevealWave.sol#104-117)
changeHiddenBaseURI(uint256,string) should be declared external:
        - ERC721RaribleRevealWave.changeHiddenBaseURI(uint256,string) (contracts/erc-721/ERC721RaribleRevealWave.sol#119-126)
changeAddTokenURIToHiddenBaseURI(uint256,bool) should be declared external:
        - ERC721RaribleRevealWave.changeAddTokenURIToHiddenBaseURI(uint256,bool) (contracts/erc-721/ERC721RaribleRevealWave.sol#128-134)
changeRevealBaseURI(uint256,string) should be declared external:
        - ERC721RaribleRevealWave.changeRevealBaseURI(uint256,string) (contracts/erc-721/ERC721RaribleRevealWave.sol#136-145)
changeRevealDate(uint256,uint256) should be declared external:
        - ERC721RaribleRevealWave.changeRevealDate(uint256,uint256) (contracts/erc-721/ERC721RaribleRevealWave.sol#147-153)
resetIsRevealedAndRevealURI(uint256) should be declared external:
        - ERC721RaribleRevealWave.resetIsRevealedAndRevealURI(uint256) (contracts/erc-721/ERC721RaribleRevealWave.sol#155-161)
changeName(uint256,string) should be declared external:
        - ERC721RaribleRevealWave.changeName(uint256,string) (contracts/erc-721/ERC721RaribleRevealWave.sol#164-171)
mintAndTransferReveal(LibERC721LazyMint.Mint721Data,address,uint256) should be declared external:
        - ERC721RaribleRevealWave.mintAndTransferReveal(LibERC721LazyMint.Mint721Data,address,uint256) (contracts/erc-721/ERC721RaribleRevealWave.sol#233-243)
balanceOf(address) should be declared external:
        - ERC721Upgradeable.balanceOf(address) (contracts/erc-721/ERC721Upgradeable.sol#117-120)
name() should be declared external:
        - ERC721Upgradeable.name() (contracts/erc-721/ERC721Upgradeable.sol#132-134)
symbol() should be declared external:
        - ERC721Upgradeable.symbol() (contracts/erc-721/ERC721Upgradeable.sol#139-141)
tokenOfOwnerByIndex(address,uint256) should be declared external:
        - ERC721Upgradeable.tokenOfOwnerByIndex(address,uint256) (contracts/erc-721/ERC721Upgradeable.sol#176-178)
totalSupply() should be declared external:
        - ERC721Upgradeable.totalSupply() (contracts/erc-721/ERC721Upgradeable.sol#183-186)
tokenByIndex(uint256) should be declared external:
        - ERC721Upgradeable.tokenByIndex(uint256) (contracts/erc-721/ERC721Upgradeable.sol#191-194)
approve(address,uint256) should be declared external:
        - ERC721Upgradeable.approve(address,uint256) (contracts/erc-721/ERC721Upgradeable.sol#199-208)
transferFrom(address,address,uint256) should be declared external:
        - ERC721Upgradeable.transferFrom(address,address,uint256) (contracts/erc-721/ERC721Upgradeable.sol#239-244)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

ERC721WoonklyNFTMinimalRevealWave.sol

```
OwnableUpgradeable._gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#74) shadows:
        - ContextUpgradeable._gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
ERC721UpgradeableMinimal._gap (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#443) shadows:
        - ERC165Upgradeable._gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - ContextUpgradeable._gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
HasContractURI.__gap (contracts/HasContractURI.sol#29) shadows:
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
Mint721Validator.__gap (contracts/Mint721Validator.sol#16) shadows:
        - ERC1271Validator.__gap (contracts/erc-1271/ERC1271Validator.sol#34)
        - EIP712Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#120)
ERC1271Validator.__gap (contracts/erc-1271/ERC1271Validator.sol#34) shadows:
        - EIP712Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#120)
ERC721BaseMinimal.__gap (contracts/erc-721-minimal/ERC721BaseMinimal.sol#38) shadows:
        - HasContractURI.__gap (contracts/HasContractURI.sol#29)
        - ERC721LazyMinimal.__gap (contracts/erc-721-minimal/ERC721LazyMinimal.sol#112)
        - ERC721URI.__gap (contracts/erc-721-minimal/ERC721URI.sol#92)
        - Mint721Validator.__gap (contracts/Mint721Validator.sol#16)
        - ERC721BurnableUpgradeableMinimal.__gap (contracts/erc-721-minimal/ERC721BurnableUpgradeableMinimal.sol#41)
        - ERC721DefaultApprovalMinimal.__gap (contracts/erc-721-minimal/ERC721DefaultApprovalMinimal.sol#24)
        - ERC721UpgradeableMinimal.__gap (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#443)
        - ERC1271Validator.__gap (contracts/erc-1271/ERC1271Validator.sol#34)
        - EIP712Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#120)
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#74)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
ERC721BurnableUpgradeableMinimal.__gap (contracts/erc-721-minimal/ERC721BurnableUpgradeableMinimal.sol#41) shadows:
        - ERC721UpgradeableMinimal.__gap (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#443)
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
ERC721DefaultApprovalMinimal.__gap (contracts/erc-721-minimal/ERC721DefaultApprovalMinimal.sol#24) shadows:
        - ERC721UpgradeableMinimal.__gap (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#443)
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
ERC721LazyMinimal.__gap (contracts/erc-721-minimal/ERC721LazyMinimal.sol#112) shadows:
        - ERC721URI.__gap (contracts/erc-721-minimal/ERC721URI.sol#92)
        - Mint721Validator.__gap (contracts/Mint721Validator.sol#16)
        - ERC721UpgradeableMinimal.__gap (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#443)
        - ERC1271Validator.__gap (contracts/erc-1271/ERC1271Validator.sol#34)
        - EIP712Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#120)
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
ERC721LazyMinimal._INTERFACE_ID_ERC165 (contracts/erc-721-minimal/ERC721LazyMinimal.sol#117) shadows:
        - ERC165Upgradeable._INTERFACE_ID_ERC165 (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#18)
ERC721LazyMinimal._INTERFACE_ID_ERC721 (contracts/erc-721-minimal/ERC721LazyMinimal.sol#118) shadows:
        - ERC721UpgradeableMinimal._INTERFACE_ID_ERC721 (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#54)
ERC721LazyMinimal._INTERFACE_ID_ERC721_METADATA (contracts/erc-721-minimal/ERC721LazyMinimal.sol#119) shadows:
        - ERC721UpgradeableMinimal._INTERFACE_ID_ERC721_METADATA (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#63)
ERC721RaribleMinimalRevealWave.__gap (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#211) shadows:
        - ERC721BaseMinimal.__gap (contracts/erc-721-minimal/ERC721BaseMinimal.sol#38)
        - HasContractURI.__gap (contracts/HasContractURI.sol#29)
        - ERC721LazyMinimal.__gap (contracts/erc-721-minimal/ERC721LazyMinimal.sol#112)
        - ERC721URI.__gap (contracts/erc-721-minimal/ERC721URI.sol#92)
        - Mint721Validator.__gap (contracts/Mint721Validator.sol#16)
        - ERC721BurnableUpgradeableMinimal.__gap (contracts/erc-721-minimal/ERC721BurnableUpgradeableMinimal.sol#41)
        - ERC721DefaultApprovalMinimal.__gap (contracts/erc-721-minimal/ERC721DefaultApprovalMinimal.sol#24)
        - ERC721UpgradeableMinimal.__gap (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#443)
        - ERC1271Validator.__gap (contracts/erc-1271/ERC1271Validator.sol#34)
        - EIP712Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#120)
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#74)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
ERC721URI.__gap (contracts/erc-721-minimal/ERC721URI.sol#92) shadows:
        - ERC721UpgradeableMinimal.__gap (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#443)
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variable-shadowing

ERC1271Validator.validate1271(address,bytes32,bytes).signerFromSig (contracts/erc-1271/ERC1271Validator.sol#19) is a local variable never initialized
ERC721RaribleMinimalRevealWave.tokenURI(uint256).tokenTRIT (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#158) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

ERC721UpgradeableMinimal._checkOnERC721Received(address,address,uint256,bytes) (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#401-422) ignores return value by IERC721ReceiverUpgradeable(to).onERC721R
eceived(_msgSender(),from,tokenId,_data) (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#408-418)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

ERC721BaseMinimal.isApprovedForAll(address,address).owner (contracts/erc-721-minimal/ERC721BaseMinimal.sol#18) shadows:
        - OwnableUpgradeable.owner() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#41-43) (function)
ERC721RaribleMinimalRevealWave._ERC721RaribleUser_init(string,string,string,string,address[],address,address)._name (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#29) shadows:
        - ERC721UpgradeableMinimal._name (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#23) (state variable)
ERC721RaribleMinimalRevealWave._ERC721RaribleUser_init(string,string,string,string,address[],address,address)._symbol (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#29) shadows:
        - ERC721UpgradeableMinimal._symbol (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#26) (state variable)
ERC721RaribleMinimalRevealWave._ERC721RaribleUser_init(string,string,string,string,address[],address,address).baseURI (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#29) shadows:
        - ERC721URI.baseURI() (contracts/erc-721-minimal/ERC721URI.sol#50-52) (function)
ERC721RaribleMinimalRevealWave._ERC721RaribleUser_init(string,string,string,string,address[],address,address).contractURI (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#29) shadows:
        - HasContractURI.contractURI (contracts/HasContractURI.sol#9) (state variable)
ERC721RaribleMinimalRevealWave._ERC721Rarible_init(string,string,string,string,address,address)._name (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#40) shadows:
        - ERC721UpgradeableMinimal._name (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#23) (state variable)
ERC721RaribleMinimalRevealWave._ERC721Rarible_init(string,string,string,string,address,address)._symbol (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#40) shadows:
        - ERC721UpgradeableMinimal._symbol (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#26) (state variable)
ERC721RaribleMinimalRevealWave._ERC721Rarible_init(string,string,string,string,address,address).baseURI (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#40) shadows:
        - ERC721URI.baseURI() (contracts/erc-721-minimal/ERC721URI.sol#50-52) (function)
ERC721RaribleMinimalRevealWave._ERC721Rarible_init(string,string,string,string,address,address).contractURI (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#40) shadows:
        - HasContractURI.contractURI (contracts/HasContractURI.sol#9) (state variable)
ERC721RaribleMinimalRevealWave._ERC721Rarible_init_unchained(string,string,string,string,address,address)._name (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#47) shadows:
        - ERC721UpgradeableMinimal._name (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#23) (state variable)
ERC721RaribleMinimalRevealWave._ERC721Rarible_init_unchained(string,string,string,string,address,address)._symbol (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#47) shadows:
        - ERC721UpgradeableMinimal._symbol (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#26) (state variable)
ERC721RaribleMinimalRevealWave._ERC721Rarible_init_unchained(string,string,string,string,address,address).baseURI (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#47) shadows:
        - ERC721URI.baseURI() (contracts/erc-721-minimal/ERC721URI.sol#50-52) (function)
ERC721RaribleMinimalRevealWave._ERC721Rarible_init_unchained(string,string,string,string,address,address).contractURI (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#47) shadows:
        - HasContractURI.contractURI (contracts/HasContractURI.sol#9) (state variable)
ERC721RaribleMinimalRevealWave.setRevealWave(uint256,string,string,bool,uint256)._name (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#72) shadows:
        - ERC721UpgradeableMinimal._name (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#23) (state variable)
ERC721RaribleMinimalRevealWave.changeHiddenBaseURI(uint256,string)._baseURI (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#87) shadows:
        - ERC721URI._baseURI (contracts/erc-721-minimal/ERC721URI.sol#22) (state variable)
ERC721RaribleMinimalRevealWave.changeName(uint256,string)._name (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#132) shadows:
        - ERC721UpgradeableMinimal._name (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#23) (state variable)
ERC721RaribleMinimalRevealWave.tokenURI(uint256)._baseURI (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#159) shadows:
        - ERC721URI._baseURI (contracts/erc-721-minimal/ERC721URI.sol#22) (state variable)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

Variable 'ERC721UpgradeableMinimal._checkOnERC721Received(address,address,uint256,bytes).retval (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#408)' in ERC721UpgradeableMinimal._checkOnERC721Received
(address,address,uint256,bytes) (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#401-422) potentially used before declaration: retval == IERC721ReceiverUpgradeable.onERC721Received.selector (node_modul
es/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#409)
Variable 'ERC721UpgradeableMinimal._checkOnERC721Received(address,address,uint256,bytes).reason (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#410)' in ERC721UpgradeableMinimal._checkOnERC721Received
(address,address,uint256,bytes) (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#401-422) potentially used before declaration: reason.length == 0 (node_modules/@rarible/tokens-minimal/contracts/erc-721
/ERC721UpgradeableMinimal.sol#411)
Variable 'ERC721UpgradeableMinimal._checkOnERC721Received(address,address,uint256,bytes).reason (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#410)' in ERC721UpgradeableMinimal._checkOnERC721Received
(address,address,uint256,bytes) (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#401-422) potentially used before declaration: revert(uint256,uint256)(32 + reason,mload(uint256)(reason)) (node_modules/
@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#415)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables

Reentrancy in ERC721LazyMinimal.mintAndTransfer(LibERC721LazyMint.Mint721Data,address) (contracts/erc-721-minimal/ERC721LazyMinimal.sol#50-70):
        External calls:
        - _safeMint(to,data.tokenId) (contracts/erc-721-minimal/ERC721LazyMinimal.sol#66)
            - IERC721ReceiverUpgradeable(to).onERC721Received(_msgSender(),from,tokenId,_data) (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#408-418)
        State variables written after the call(s):
        - _setTokenURI(data.tokenId,data.tokenURI) (contracts/erc-721-minimal/ERC721LazyMinimal.sol#69)
            - _tokenURIs[tokenId] = _tokenURI (contracts/erc-721-minimal/ERC721URI.sol#33)
        - _saveCreators(data.tokenId,data.creators) (contracts/erc-721-minimal/ERC721LazyMinimal.sol#68)
            - creatorsOfToken.push(_creators[i]) (contracts/erc-721-minimal/ERC721LazyMinimal.sol#88)
        - _saveRoyalties(data.tokenId,data.royalties) (contracts/erc-721-minimal/ERC721LazyMinimal.sol#67)
            - _royalties[id].push(_royalties[i]) (node_modules/@rarible/royalties/contracts/impl/AbstractRoyalties.sol#16)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2

Reentrancy in ERC721LazyMinimal.mintAndTransfer(LibERC721LazyMint.Mint721Data,address) (contracts/erc-721-minimal/ERC721LazyMinimal.sol#50-70):
        External calls:
        - _safeMint(to,data.tokenId) (contracts/erc-721-minimal/ERC721LazyMinimal.sol#66)
            - IERC721ReceiverUpgradeable(to).onERC721Received(_msgSender(),from,tokenId,_data) (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#408-418)
        Event emitted after the call(s):
        - Creators(tokenId,_creators) (contracts/erc-721-minimal/ERC721LazyMinimal.sol#92)
            - _saveCreators(data.tokenId,data.creators) (contracts/erc-721-minimal/ERC721LazyMinimal.sol#68)
        - RoyaltiesSet(id,_royalties) (node_modules/@rarible/royalties/contracts/impl/RoyaltiesV2Impl.sol#16)
            - _saveRoyalties(data.tokenId,data.royalties) (contracts/erc-721-minimal/ERC721LazyMinimal.sol#67)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

EIP712Upgradeable._getChainId() (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#93-99) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#96-98)
AddressUpgradeable.isContract(address) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#26-35) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#33)
AddressUpgradeable._verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#147-164) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#156-159)
LibSignature.recover(bytes32,bytes) (node_modules/@rarible/libraries/contracts/LibSignature.sol#20-45) uses assembly
        - INLINE ASM (node_modules/@rarible/libraries/contracts/LibSignature.sol#38-42)
ERC721UpgradeableMinimal._checkOnERC721Received(address,address,uint256,bytes) (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#401-422) uses assembly
        - INLINE ASM (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#414-416)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

ERC721RaribleMinimalRevealWave.tokenURI(uint256) (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#152-180) compares to a boolean constant:
        -revealWave.isRevealed == true || revealWave.addTokenURITOHiddenBaseURI == true (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#161)
ERC721RaribleMinimalRevealWave.tokenURI(uint256) (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#152-180) compares to a boolean constant:
        -revealWave.isRevealed == false (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#166)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#boolean-equality

Different versions of Solidity is used:
        - Version used: ['0.7.6', '>=0.4.24<0.8.0', '>=0.6.0<0.8.0', '>=0.6.2<0.8.0', '^0.7.0']
```

```
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/introspection/IERC165Upgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#3)
        - >=0.4.24<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/proxy/Initializable.sol#4)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC721/ERC721MetadataUpgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC721/IERC721ReceiverUpgradeable.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC721/IERC721Upgradeable.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/StringsUpgradeable.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@rarible/lazy-mint/contracts/erc-721/IERC721LazyMint.sol#3)
        - v2 (node_modules/@rarible/lazy-mint/contracts/erc-721/IERC721LazyMint.sol#4)
        - >=0.6.2<0.8.0 (node_modules/@rarible/lazy-mint/contracts/erc-721/LibERC721LazyMint.sol#3)
        - ^0.7.0 (node_modules/@rarible/libraries/contracts/LibSignature.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@rarible/royalties-upgradeable/contracts/RoyaltiesV2Upgradeable.sol#3)
        - v2 (node_modules/@rarible/royalties-upgradeable/contracts/RoyaltiesV2Upgradeable.sol#4)
        - >=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/LibPart.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/LibRoyaltiesV2.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/RoyaltiesV2.sol#3)
        - v2 (node_modules/@rarible/royalties/contracts/RoyaltiesV2.sol#4)
        - >=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/impl/AbstractRoyalties.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/impl/RoyaltiesV2Impl.sol#3)
        - v2 (node_modules/@rarible/royalties/contracts/impl/RoyaltiesV2Impl.sol#4)
        - 0.7.6 (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#3)
        - 0.7.6 (contracts/HasContractURI.sol#3)
        - 0.7.6 (contracts/LibURI.sol#3)
        - 0.7.6 (contracts/Mint721Validator.sol#3)
        - 0.7.6 (contracts/erc-1271/ERC1271.sol#3)
        - 0.7.6 (contracts/erc-1271/ERC1271Validator.sol#3)
        - 0.7.6 (contracts/erc-721-minimal/ERC721BaseMinimal.sol#3)
        - v2 (contracts/erc-721-minimal/ERC721BaseMinimal.sol#4)
        - >=0.6.0<0.8.0 (contracts/erc-721-minimal/ERC721BurnableUpgradeableMinimal.sol#3)
        - 0.7.6 (contracts/erc-721-minimal/ERC721DefaultApprovalMinimal.sol#3)
        - 0.7.6 (contracts/erc-721-minimal/ERC721LazyMinimal.sol#3)
        - v2 (contracts/erc-721-minimal/ERC721LazyMinimal.sol#4)
        - 0.7.6 (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#3)
        - v2 (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#4)
        - >=0.6.0<0.8.0 (contracts/erc-721-minimal/ERC721URI.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

AbstractRoyalties._onRoyaltiesSet(uint256,LibPart.Part[]) (node_modules/@rarible/royalties/contracts/impl/AbstractRoyalties.sol#31) is never used and should be removed
AddressUpgradeable._verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#147-164) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#79-81) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#88-91) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#104-106) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#114-121) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#129-131) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#139-141) is never used and should be removed
AddressUpgradeable.sendValue(address,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#53-59) is never used and should be removed
ContextUpgradeable.__Context_init() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#17-19) is never used and should be removed
ContextUpgradeable._msgData() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#27-30) is never used and should be removed
EIP712Upgradeable.__EIP712_init(string,string) (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#44-46) is never used and should be removed
ERC1271.returnIsValidSignatureMagicNumber(bool) (contracts/erc-1271/ERC1271.sol#22-24) is never used and should be removed
ERC165Upgradeable.__ERC165_init() (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#23-27) is never used and should be removed
ERC721BurnableUpgradeableMinimal.__ERC721Burnable_init() (contracts/erc-721-minimal/ERC721BurnableUpgradeableMinimal.sol#14-18) is never used and should be removed
ERC721UpgradeableMinimal.__ERC721_init(string,string) (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#71-75) is never used and should be removed
ERC721UpgradeableMinimal._clearMetadata(uint256) (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#347-348) is never used and should be removed
ERC721UpgradeableMinimal._emitMintEvent(address,uint256) (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#310-312) is never used and should be removed
HasContractURI._setContractURI(string) (contracts/HasContractURI.sol#25-27) is never used and should be removed
OwnableUpgradeable.__Ownable_init() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#27-30) is never used and should be removed
SafeMathUpgradeable.div(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#135-138) is never used and should be removed
SafeMathUpgradeable.div(uint256,uint256,string) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#190-193) is never used and should be removed
SafeMathUpgradeable.mod(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#152-155) is never used and should be removed
SafeMathUpgradeable.mod(uint256,uint256,string) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#210-213) is never used and should be removed
SafeMathUpgradeable.mul(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#116-121) is never used and should be removed
SafeMathUpgradeable.sub(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#101-104) is never used and should be removed
SafeMathUpgradeable.sub(uint256,uint256,string) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#170-173) is never used and should be removed
SafeMathUpgradeable.tryAdd(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#24-28) is never used and should be removed
SafeMathUpgradeable.tryDiv(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#60-63) is never used and should be removed
SafeMathUpgradeable.tryMod(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#70-73) is never used and should be removed
SafeMathUpgradeable.tryMul(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#45-53) is never used and should be removed
SafeMathUpgradeable.trySub(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#35-38) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/introspection/IERC165Upgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#3) is too complex
Pragma version>=0.4.24<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/proxy/Initializable.sol#4) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC721/ERC721MetadataUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC721/IERC721ReceiverUpgradeable.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC721/IERC721Upgradeable.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/StringsUpgradeable.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/lazy-mint/contracts/erc-721/IERC721LazyMint.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/lazy-mint/contracts/erc-721/LibERC721LazyMint.sol#3) is too complex
Pragma version^0.7.0 (node_modules/@rarible/libraries/contracts/LibSignature.sol#3) allows old versions
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties-upgradeable/contracts/RoyaltiesV2Upgradeable.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/LibPart.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/LibRoyaltiesV2.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/RoyaltiesV2.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/impl/AbstractRoyalties.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/impl/RoyaltiesV2Impl.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (contracts/erc-721-minimal/ERC721BurnableUpgradeableMinimal.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (contracts/erc-721-minimal/ERC721URI.sol#3) is too complex
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

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

Function OwnableUpgradeable.__Ownable_init() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#27-30) is not in mixedCase
Function OwnableUpgradeable.__Ownable_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#32-36) is not in mixedCase
Variable OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#74) is not in mixedCase
Function EIP712Upgradeable.__EIP712_init(string,string) (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#44-46) is not in mixedCase
Function EIP712Upgradeable.__EIP712_init_unchained(string,string) (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#48-53) is not in mixedCase
Function EIP712Upgradeable._EIP712NameHash() (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#107-109) is not in mixedCase
Function EIP712Upgradeable._EIP712VersionHash() (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#117-119) is not in mixedCase
Variable EIP712Upgradeable._HASHED_NAME (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#27) is not in mixedCase
Variable EIP712Upgradeable._HASHED_VERSION (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#28) is not in mixedCase
Variable EIP712Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#120) is not in mixedCase
Function ERC165Upgradeable.__ERC165_init() (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#23-27) is not in mixedCase
Function ERC165Upgradeable.__ERC165_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#29-33) is not in mixedCase
Variable ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59) is not in mixedCase
Function ContextUpgradeable.__Context_init() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#17-19) is not in mixedCase
Function ContextUpgradeable.__Context_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#21-22) is not in mixedCase
Variable ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31) is not in mixedCase
Function RoyaltiesV2Upgradeable.__RoyaltiesV2Upgradeable_init_unchained() (node_modules/@rarible/royalties-upgradeable/contracts/RoyaltiesV2Upgradeable.sol#11-13) is not in mixedCase

Function ERC721UpgradeableMinimal.__ERC721_init(string,string) (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#71-75) is not in mixedCase
Function ERC721UpgradeableMinimal.__ERC721_init_unchained(string,string) (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#77-84) is not in mixedCase
Parameter ERC721UpgradeableMinimal.safeTransferFrom(address,address,uint256,bytes)._data (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#196) is not in mixedCase
Variable ERC721UpgradeableMinimal.__gap (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#443) is not in mixedCase
Function HasContractURI.__HasContractURI_init_unchained(string) (contracts/HasContractURI.sol#16-19) is not in mixedCase
Parameter HasContractURI.__HasContractURI_init_unchained(string)._contractURI (contracts/HasContractURI.sol#16) is not in mixedCase
Variable HasContractURI.__gap (contracts/HasContractURI.sol#29) is not in mixedCase
Parameter LibURI.toBytPrefix(string,string)._toURI (contracts/LibURI.sol#9) is not in mixedCase
Function Mint721Validator.__Mint721Validator_init_unchained() (contracts/Mint721Validator.sol#9-11) is not in mixedCase
Variable Mint721Validator.__gap (contracts/Mint721Validator.sol#16) is not in mixedCase
Variable ERC1271Validator.__gap (contracts/erc-1271/ERC1271Validator.sol#36) is not in mixedCase
Variable ERC721BaseMinimal.__gap (contracts/erc-721-minimal/ERC721BaseMinimal.sol#38) is not in mixedCase
Function ERC721BurnableUpgradeableMinimal.__ERC721Burnable_init() (contracts/erc-721-minimal/ERC721BurnableUpgradeableMinimal.sol#14-18) is not in mixedCase
Function ERC721BurnableUpgradeableMinimal.__ERC721Burnable_init_unchained() (contracts/erc-721-minimal/ERC721BurnableUpgradeableMinimal.sol#20-21) is not in mixedCase
Variable ERC721BurnableUpgradeableMinimal.__gap (contracts/erc-721-minimal/ERC721BurnableUpgradeableMinimal.sol#41) is not in mixedCase
Variable ERC721DefaultApprovalMinimal.__gap (contracts/erc-721-minimal/ERC721DefaultApprovalMinimal.sol#24) is not in mixedCase
Function ERC721LazyMinimal.__ERC721Lazy_init_unchained() (contracts/erc-721-minimal/ERC721LazyMinimal.sol#25-27) is not in mixedCase
Parameter ERC721LazyMinimal.updateAccount(uint256,address,address)._id (contracts/erc-721-minimal/ERC721LazyMinimal.sol#95) is not in mixedCase
Parameter ERC721LazyMinimal.updateAccount(uint256,address,address)._from (contracts/erc-721-minimal/ERC721LazyMinimal.sol#95) is not in mixedCase
Parameter ERC721LazyMinimal.updateAccount(uint256,address,address)._to (contracts/erc-721-minimal/ERC721LazyMinimal.sol#95) is not in mixedCase
Parameter ERC721LazyMinimal.getCreators(uint256)._id (contracts/erc-721-minimal/ERC721LazyMinimal.sol#100) is not in mixedCase
Variable ERC721LazyMinimal.__gap (contracts/erc-721-minimal/ERC721LazyMinimal.sol#112) is not in mixedCase
Function ERC721RaribleMinimalRevealWave.__ERC721RaribleUser_init(string,string,string,string,address[],address,address) (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#29-38) is not in mixedCase
Parameter ERC721RaribleMinimalRevealWave.__ERC721RaribleUser_init(string,string,string,string,address[],address,address)._name (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#29) is not in mixedCase
Parameter ERC721RaribleMinimalRevealWave.__ERC721RaribleUser_init(string,string,string,string,address[],address,address)._symbol (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#29) is not in mixedCase
Function ERC721RaribleMinimalRevealWave.__ERC721Rarible_init(string,string,string,string,address,address) (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#40-45) is not in mixedCase
Parameter ERC721RaribleMinimalRevealWave.__ERC721Rarible_init(string,string,string,string,address,address)._name (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#40) is not in mixedCase
Parameter ERC721RaribleMinimalRevealWave.__ERC721Rarible_init(string,string,string,string,address,address)._symbol (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#40) is not in mixedCase
Function ERC721RaribleMinimalRevealWave.__ERC721Rarible_init_unchained(string,string,string,string,address,address) (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#47-62) is not in mixedCase
Parameter ERC721RaribleMinimalRevealWave.__ERC721Rarible_init_unchained(string,string,string,string,address,address)._name (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#47) is not in mixedCase
Parameter ERC721RaribleMinimalRevealWave.__ERC721Rarible_init_unchained(string,string,string,string,address,address)._symbol (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#47) is not in mixedCase
Parameter ERC721RaribleMinimalRevealWave.getRevealWaveIdByTokenId(uint256)._tokenId (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#64) is not in mixedCase
Parameter ERC721RaribleMinimalRevealWave.setRevealWave(uint256,string,string,bool,uint256)._waveId (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#71) is not in mixedCase
Parameter ERC721RaribleMinimalRevealWave.setRevealWave(uint256,string,string,bool,uint256)._name (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#72) is not in mixedCase
Parameter ERC721RaribleMinimalRevealWave.setRevealWave(uint256,string,string,bool,uint256)._hiddenBaseURI (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#73) is not in mixedCase
Parameter ERC721RaribleMinimalRevealWave.setRevealWave(uint256,string,string,bool,uint256)._addTokenURIToHiddenBaseURI (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#74) is not in mixedCase
Parameter ERC721RaribleMinimalRevealWave.setRevealWave(uint256,string,string,bool,uint256)._revealBaseURI (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#75) is not in mixedCase
Parameter ERC721RaribleMinimalRevealWave.changeHiddenBaseURI(uint256,string)._waveId (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#86) is not in mixedCase
Parameter ERC721RaribleMinimalRevealWave.changeHiddenBaseURI(uint256,string)._baseURI (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#87) is not in mixedCase
Parameter ERC721RaribleMinimalRevealWave.changeAddTokenURIToHiddenBaseURI(uint256,bool)._waveId (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#95) is not in mixedCase
Parameter ERC721RaribleMinimalRevealWave.changeAddTokenURIToHiddenBaseURI(uint256,bool)._addTokenURIToHiddenBaseURI (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#96) is not in mixedCase
Parameter ERC721RaribleMinimalRevealWave.changeRevealBaseURI(uint256,string)._waveId (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#103) is not in mixedCase
Parameter ERC721RaribleMinimalRevealWave.changeRevealBaseURI(uint256,string)._revealBaseURI (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#104) is not in mixedCase
Parameter ERC721RaribleMinimalRevealWave.changeRevealDate(uint256,uint256)._waveId (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#114) is not in mixedCase
Parameter ERC721RaribleMinimalRevealWave.changeRevealDate(uint256,uint256)._revealDate (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#115) is not in mixedCase
Parameter ERC721RaribleMinimalRevealWave.getRevealAndRevealURI(uint256)._waveId (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#122) is not in mixedCase
Parameter ERC721RaribleMinimalRevealWave.changeName(uint256,string)._waveId (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#131) is not in mixedCase
Parameter ERC721RaribleMinimalRevealWave.changeName(uint256,string)._name (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#132) is not in mixedCase
Parameter ERC721RaribleMinimalRevealWave.assignRevealWaveIdToTokenId(uint256,uint256)._tokenId (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#140) is not in mixedCase
Parameter ERC721RaribleMinimalRevealWave.assignRevealWaveIdToTokenId(uint256,uint256)._waveId (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#140) is not in mixedCase
```

```
Parameter ERC721RaribleMinimalRevealWave.tokenURI(uint256)._tokenId (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#152) is not in mixedCase
Parameter ERC721RaribleMinimalRevealWave.mintAndTransferReveal(LibERC721LazyMint.Mint721Data,address,uint256)._waveId (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#202) is not in mixedCase
Variable ERC721RaribleMinimalRevealWave.__gap (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#211) is not in mixedCase
Variable ERC721URI.__gap (contracts/erc-721-minimal/ERC721URI.sol#92) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#94)" inEIP712Upgradeable (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#25-121)
Redundant expression "this (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#28)" inContextUpgradeable (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#16-32)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

ERC1271.slitherConstructorConstantVariables() (contracts/erc-1271/ERC1271.sol#5-26) uses literals with too many digits:
        - ERC1271.RETURN_INVALID_SIGNATURE = 0x00000000 (contracts/erc-1271/ERC1271.sol#9)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits

ERC1271 (contracts/erc-1271/ERC1271.sol#5-26) does not implement functions:
        - ERC1271.isValidSignature(bytes32,bytes) (contracts/erc-1271/ERC1271.sol#20)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unimplemented-functions

LibERC721LazyMint._INTERFACE_ID_MINT_AND_TRANSFER (node_modules/@rarible/lazy-mint/contracts/erc-721/LibERC721LazyMint.sol#9) is never used in LibERC721LazyMint (node_modules/@rarible/lazy-mint/contracts/erc-721/LibERC721LazyMint.sol#7-3
9)
ERC721RaribleMinimalRevealWave.__gap (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#211) is never used in ERC721RaribleMinimalRevealWave (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#8-213)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

renounceOwnership() should be declared external:
        - OwnableUpgradeable.renounceOwnership() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#60-63)
transferOwnership(address) should be declared external:
        - OwnableUpgradeable.transferOwnership(address) (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#69-73)
balanceOf(address) should be declared external:
        - ERC721UpgradeableMinimal.balanceOf(address) (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#89-92)
name() should be declared external:
        - ERC721UpgradeableMinimal.name() (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#106-108)
symbol() should be declared external:
        - ERC721UpgradeableMinimal.symbol() (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#113-115)
approve(address,uint256) should be declared external:
        - ERC721UpgradeableMinimal.approve(address,uint256) (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#126-136)
transferFrom(address,address,uint256) should be declared external:
        - ERC721UpgradeableMinimal.transferFrom(address,address,uint256) (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#167-176)
isValidSignature(bytes32,bytes) should be declared external:
        - ERC1271.isValidSignature(bytes32,bytes) (contracts/erc-1271/ERC1271.sol#20)
burn(uint256) should be declared external:
        - ERC721BurnableUpgradeableMinimal.burn(uint256) (contracts/erc-721-minimal/ERC721BurnableUpgradeableMinimal.sol#29-39)
setRevealWave(uint256,string,string,bool,uint256) should be declared external:
        - ERC721RaribleMinimalRevealWave.setRevealWave(uint256,string,string,bool,uint256) (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#70-83)
changeHiddenBaseURI(uint256,string) should be declared external:
        - ERC721RaribleMinimalRevealWave.changeHiddenBaseURI(uint256,string) (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#85-92)
changeAddTokenURIToHiddenBaseURI(uint256,bool) should be declared external:
        - ERC721RaribleMinimalRevealWave.changeAddTokenURIToHiddenBaseURI(uint256,bool) (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#94-100)
changeRevealBaseURI(uint256,string) should be declared external:
        - ERC721RaribleMinimalRevealWave.changeRevealBaseURI(uint256,string) (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#102-111)
changeRevealDate(uint256,uint256) should be declared external:
        - ERC721RaribleMinimalRevealWave.changeRevealDate(uint256,uint256) (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#113-119)
resetIsRevealedAndSRevealURI(uint256) should be declared external:
        - ERC721RaribleMinimalRevealWave.resetIsRevealedAndSRevealURI(uint256) (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#121-127)
changeName(uint256,string) should be declared external:
        - ERC721RaribleMinimalRevealWave.changeName(uint256,string) (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#130-137)
mintAndTransferReveal(LibERC721LazyMint.Mint721Data,address,uint256) should be declared external:
        - ERC721RaribleMinimalRevealWave.mintAndTransferReveal(LibERC721LazyMint.Mint721Data,address,uint256) (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#199-209)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

ERC721WoonklyNFTFactoryC2.sol

```
OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#74) shadows:
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
ERC721UpgradeableMinimal.__gap (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#463) shadows:
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
HasContractURI.__gap (contracts/HasContractURI.sol#29) shadows:
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
Mint721Validator.__gap (contracts/Mint721Validator.sol#16) shadows:
        - ERC1271Validator.__gap (contracts/erc-1271/ERC1271Validator.sol#34)
        - EIP712Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#120)
ERC1271Validator.__gap (contracts/erc-1271/ERC1271Validator.sol#34) shadows:
        - EIP712Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#120)
ERC721BaseMinimal.__gap (contracts/erc-721-minimal/ERC721BaseMinimal.sol#38) shadows:
        - HasContractURI.__gap (contracts/HasContractURI.sol#29)
        - ERC721LazyMinimal.__gap (contracts/erc-721-minimal/ERC721LazyMinimal.sol#112)
        - ERC721URI.__gap (contracts/erc-721-minimal/ERC721URI.sol#92)
        - Mint721Validator.__gap (contracts/Mint721Validator.sol#16)
        - ERC721BurnableUpgradeableMinimal.__gap (contracts/erc-721-minimal/ERC721BurnableUpgradeableMinimal.sol#41)
        - ERC721DefaultApprovalMinimal.__gap (contracts/erc-721-minimal/ERC721DefaultApprovalMinimal.sol#24)
        - ERC721UpgradeableMinimal.__gap (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#463)
        - ERC1271Validator.__gap (contracts/erc-1271/ERC1271Validator.sol#34)
        - EIP712Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#120)
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#74)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
ERC721BurnableUpgradeableMinimal.__gap (contracts/erc-721-minimal/ERC721BurnableUpgradeableMinimal.sol#41) shadows:
        - ERC721UpgradeableMinimal.__gap (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#463)
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
ERC721DefaultApprovalMinimal.__gap (contracts/erc-721-minimal/ERC721DefaultApprovalMinimal.sol#24) shadows:
        - ERC721UpgradeableMinimal.__gap (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#463)
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
ERC721LazyMinimal.__gap (contracts/erc-721-minimal/ERC721LazyMinimal.sol#112) shadows:
        - ERC721URI.__gap (contracts/erc-721-minimal/ERC721URI.sol#92)
        - Mint721Validator.__gap (contracts/Mint721Validator.sol#16)
        - ERC721UpgradeableMinimal.__gap (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#463)
        - ERC1271Validator.__gap (contracts/erc-1271/ERC1271Validator.sol#34)
        - EIP712Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#120)
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
ERC721LazyMinimal._INTERFACE_ID_ERC165 (contracts/erc-721-minimal/ERC721LazyMinimal.sol#17) shadows:
        - ERC165Upgradeable._INTERFACE_ID_ERC165 (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#18)
ERC721LazyMinimal._INTERFACE_ID_ERC721 (contracts/erc-721-minimal/ERC721LazyMinimal.sol#18) shadows:
        - ERC721UpgradeableMinimal._INTERFACE_ID_ERC721 (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#54)
ERC721LazyMinimal._INTERFACE_ID_ERC721_METADATA (contracts/erc-721-minimal/ERC721LazyMinimal.sol#19) shadows:
        - ERC721UpgradeableMinimal._INTERFACE_ID_ERC721_METADATA (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#63)
ERC721RaribleMinimal.__gap (contracts/erc-721-minimal/ERC721RaribleMinimal.sol#57) shadows:
        - ERC721BaseMinimal.__gap (contracts/erc-721-minimal/ERC721BaseMinimal.sol#38)
        - HasContractURI.__gap (contracts/HasContractURI.sol#29)
        - ERC721LazyMinimal.__gap (contracts/erc-721-minimal/ERC721LazyMinimal.sol#112)
        - ERC721URI.__gap (contracts/erc-721-minimal/ERC721URI.sol#92)
        - Mint721Validator.__gap (contracts/Mint721Validator.sol#16)
        - ERC721BurnableUpgradeableMinimal.__gap (contracts/erc-721-minimal/ERC721BurnableUpgradeableMinimal.sol#41)
        - ERC721DefaultApprovalMinimal.__gap (contracts/erc-721-minimal/ERC721DefaultApprovalMinimal.sol#24)
        - ERC721UpgradeableMinimal.__gap (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#463)
        - ERC1271Validator.__gap (contracts/erc-1271/ERC1271Validator.sol#34)
        - EIP712Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#120)
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#74)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
ERC721URI.__gap (contracts/erc-721-minimal/ERC721URI.sol#52) shadows:
        - ERC721UpgradeableMinimal.__gap (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#463)
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variable-shadowing

ERC1271Validator.validate1271(address,bytes32,bytes).signerFromSig (contracts/erc-1271/ERC1271Validator.sol#19) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

BeaconProxy._setBeacon(address,bytes) (node_modules/@openzeppelin/contracts/proxy/BeaconProxy.sol#68-87) ignores return value by Address.functionDelegateCall(_implementation(),data,BeaconProxy: function call failed) (node_modules/@openze
ppelin/contracts/proxy/BeaconProxy.sol#85)
ERC721UpgradeableMinimal._checkOnERC721Received(address,address,uint256,bytes) (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#401-422) ignores return value by IERC721ReceiverUpgradeable(to).onERC721R
eceived(_msgSender(),from,tokenId,_data) (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#405-418)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

ERC721BaseMinimal.isApprovedForAll(address,address).owner (contracts/erc-721-minimal/ERC721BaseMinimal.sol#18) shadows:
        - OwnableUpgradeable.owner() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#41-43) (function)
ERC721RaribleMinimal.__ERC721RaribleUser_init(string,string,string,string,address[],address,address)._name (contracts/erc-721-minimal/ERC721RaribleMinimal.sol#15) shadows:
        - ERC721UpgradeableMinimal._name (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#23) (state variable)
ERC721RaribleMinimal.__ERC721RaribleUser_init(string,string,string,string,address[],address,address)._symbol (contracts/erc-721-minimal/ERC721RaribleMinimal.sol#15) shadows:
        - ERC721UpgradeableMinimal._symbol (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#26) (state variable)
ERC721RaribleMinimal.__ERC721RaribleUser_init(string,string,string,string,address[],address,address).baseURI (contracts/erc-721-minimal/ERC721RaribleMinimal.sol#15) shadows:
        - ERC721URI.baseURI() (contracts/erc-721-minimal/ERC721URI.sol#50-52) (function)
ERC721RaribleMinimal.__ERC721RaribleUser_init(string,string,string,string,address[],address,address).contractURI (contracts/erc-721-minimal/ERC721RaribleMinimal.sol#15) shadows:
        - HasContractURI.contractURI (contracts/HasContractURI.sol#9) (state variable)
ERC721RaribleMinimal.__ERC721Rarible_init(string,string,string,string,address,address)._name (contracts/erc-721-minimal/ERC721RaribleMinimal.sol#26) shadows:
        - ERC721UpgradeableMinimal._name (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#23) (state variable)
ERC721RaribleMinimal.__ERC721Rarible_init(string,string,string,string,address,address)._symbol (contracts/erc-721-minimal/ERC721RaribleMinimal.sol#26) shadows:
        - ERC721UpgradeableMinimal._symbol (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#26) (state variable)
ERC721RaribleMinimal.__ERC721Rarible_init(string,string,string,string,address,address).baseURI (contracts/erc-721-minimal/ERC721RaribleMinimal.sol#26) shadows:
        - ERC721URI.baseURI() (contracts/erc-721-minimal/ERC721URI.sol#50-52) (function)
ERC721RaribleMinimal.__ERC721Rarible_init(string,string,string,string,address,address).contractURI (contracts/erc-721-minimal/ERC721RaribleMinimal.sol#26) shadows:
        - HasContractURI.contractURI (contracts/HasContractURI.sol#9) (state variable)
ERC721RaribleMinimal.__ERC721Rarible_init_unchained(string,string,string,string,address,address)._name (contracts/erc-721-minimal/ERC721RaribleMinimal.sol#33) shadows:
        - ERC721UpgradeableMinimal._name (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#23) (state variable)
ERC721RaribleMinimal.__ERC721Rarible_init_unchained(string,string,string,string,address,address)._symbol (contracts/erc-721-minimal/ERC721RaribleMinimal.sol#33) shadows:
        - ERC721UpgradeableMinimal._symbol (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#26) (state variable)
ERC721RaribleMinimal.__ERC721Rarible_init_unchained(string,string,string,string,address,address).baseURI (contracts/erc-721-minimal/ERC721RaribleMinimal.sol#33) shadows:
        - ERC721URI.baseURI() (contracts/erc-721-minimal/ERC721URI.sol#50-52) (function)
ERC721RaribleMinimal.__ERC721Rarible_init_unchained(string,string,string,string,address,address).contractURI (contracts/erc-721-minimal/ERC721RaribleMinimal.sol#33) shadows:
        - HasContractURI.contractURI (contracts/HasContractURI.sol#9) (state variable)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing
```

AUTOMATED TESTING

```
ERC721RaribleFactoryC2.constructor(address,address,address)._beacon (contracts/create-2/ERC721RaribleFactoryC2.sol#24) lacks a zero-check on :
            - beacon = _beacon (contracts/create-2/ERC721RaribleFactoryC2.sol#25)
ERC721RaribleFactoryC2.constructor(address,address,address)._transferProxy (contracts/create-2/ERC721RaribleFactoryC2.sol#24) lacks a zero-check on :
            - transferProxy = _transferProxy (contracts/create-2/ERC721RaribleFactoryC2.sol#26)
ERC721RaribleFactoryC2.constructor(address,address,address)._lazyTransferProxy (contracts/create-2/ERC721RaribleFactoryC2.sol#24) lacks a zero-check on :
            - lazyTransferProxy = _lazyTransferProxy (contracts/create-2/ERC721RaribleFactoryC2.sol#27)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation

Variable 'ERC721UpgradeableMinimal._checkOnERC721Received(address,address,uint256,bytes).retval (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#408)' in ERC721UpgradeableMinimal._checkOnERC721Received
(address,address,uint256,bytes) (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#401-422) potentially used before declaration: retval == IERC721ReceiverUpgradeable.onERC721Received.selector (node_modul
es/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#409)
Variable 'ERC721UpgradeableMinimal._checkOnERC721Received(address,address,uint256,bytes).reason (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#410)' in ERC721UpgradeableMinimal._checkOnERC721Received
(address,address,uint256,bytes) (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#401-422) potentially used before declaration: reason.length == 0 (node_modules/@rarible/tokens-minimal/contracts/erc-721
/ERC721UpgradeableMinimal.sol#411)
Variable 'ERC721UpgradeableMinimal._checkOnERC721Received(address,address,uint256,bytes).reason (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#410)' in ERC721UpgradeableMinimal._checkOnERC721Received
(address,address,uint256,bytes) (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#401-422) potentially used before declaration: revert(uint256,uint256)(32 + reason,mload(uint256)(reason)) (node_modules/
@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#415)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables

Reentrancy in ERC721LazyMinimal.mintAndTransfer(LibERC721LazyMint.Mint721Data,address) (contracts/erc-721-minimal/ERC721LazyMinimal.sol#50-70):
        External calls:
        - _safeMint(to,data.tokenId) (contracts/erc-721-minimal/ERC721LazyMinimal.sol#66)
            - IERC721ReceiverUpgradeable(to).onERC721Received(_msgSender(),from,tokenId,_data) (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#408-418)
        State variables written after the call(s):
        - _setTokenURI(data.tokenId,data.tokenURI) (contracts/erc-721-minimal/ERC721LazyMinimal.sol#69)
            - _tokenURIs[tokenId] = _tokenURI (contracts/erc-721-minimal/ERC721URI.sol#53)
        - _saveCreators(data.tokenId,data.creators) (contracts/erc-721-minimal/ERC721LazyMinimal.sol#68)
            - creatorsOfToken.push(_creators[i]) (contracts/erc-721-minimal/ERC721LazyMinimal.sol#88)
        - _saveRoyalties(data.tokenId,data.royalties) (contracts/erc-721-minimal/ERC721LazyMinimal.sol#67)
            - royalties[id].push(_royalties[i]) (node_modules/@rarible/royalties/contracts/impl/AbstractRoyalties.sol#16)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2

Reentrancy in ERC721RaribleFactoryC2.createToken(string,string,string,string,uint256) (contracts/create-2/ERC721RaribleFactoryC2.sol#30-35):
        External calls:
        - token.transferOwnership(_msgSender()) (contracts/create-2/ERC721RaribleFactoryC2.sol#33)
        Event emitted after the call(s):
        - Create721RaribleProxy(beaconProxy) (contracts/create-2/ERC721RaribleFactoryC2.sol#34)
Reentrancy in ERC721RaribleFactoryC2.createToken(string,string,string,string,address[],uint256) (contracts/create-2/ERC721RaribleFactoryC2.sol#37-42):
        External calls:
        - token.transferOwnership(_msgSender()) (contracts/create-2/ERC721RaribleFactoryC2.sol#40)
        Event emitted after the call(s):
        - Create721RaribleUserProxy(beaconProxy) (contracts/create-2/ERC721RaribleFactoryC2.sol#41)
Reentrancy in ERC721LazyMinimal.mintAndTransfer(LibERC721LazyMint.Mint721Data,address) (contracts/erc-721-minimal/ERC721LazyMinimal.sol#50-70):
        External calls:
        - _safeMint(to,data.tokenId) (contracts/erc-721-minimal/ERC721LazyMinimal.sol#66)
            - IERC721ReceiverUpgradeable(to).onERC721Received(_msgSender(),from,tokenId,_data) (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#408-418)
        Event emitted after the call(s):
        - Creators(tokenId,_creators) (contracts/erc-721-minimal/ERC721LazyMinimal.sol#92)
            - _saveCreators(data.tokenId,data.creators) (contracts/erc-721-minimal/ERC721LazyMinimal.sol#68)
        - RoyaltiesSet(id,_royalties) (node_modules/@rarible/royalties/contracts/impl/RoyaltiesV2Impl.sol#16)
            - _saveRoyalties(data.tokenId,data.royalties) (contracts/erc-721-minimal/ERC721LazyMinimal.sol#67)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

EIP712Upgradeable._getChainId() (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#93-99) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#96-98)
AddressUpgradeable.isContract(address) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#26-35) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#33)
AddressUpgradeable._verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#147-164) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#156-159)
BeaconProxy._beacon() (node_modules/@openzeppelin/contracts/proxy/BeaconProxy.sol#43-49) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/proxy/BeaconProxy.sol#46-48)
BeaconProxy._setBeacon(address,bytes) (node_modules/@openzeppelin/contracts/proxy/BeaconProxy.sol#68-87) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/proxy/BeaconProxy.sol#80-82)
Proxy._delegate(address) (node_modules/@openzeppelin/contracts/proxy/Proxy.sol#21-41) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/proxy/Proxy.sol#23-40)
Address.isContract(address) (node_modules/@openzeppelin/contracts/utils/Address.sol#26-35) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/utils/Address.sol#33)
Address._verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#171-188) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/utils/Address.sol#180-183)
LibSignature.recover(bytes32,bytes) (node_modules/@rarible/libraries/contracts/LibSignature.sol#20-45) uses assembly
        - INLINE ASM (node_modules/@rarible/libraries/contracts/LibSignature.sol#38-42)
ERC721UpgradeableMinimal._checkOnERC721Received(address,address,uint256,bytes) (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#401-422) uses assembly
        - INLINE ASM (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#414-416)
ERC721RaribleFactoryC2.deployProxy(bytes,uint256) (contracts/create-2/ERC721RaribleFactoryC2.sol#45-53) uses assembly
        - INLINE ASM (contracts/create-2/ERC721RaribleFactoryC2.sol#47-52)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Different versions of Solidity is used:
        - Version used: ['0.7.6', '>=0.4.24<0.8.0', '>=0.6.0<0.8.0', '>=0.6.2<0.8.0', '^0.7.0']
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/introspection/IERC165Upgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#3)
        - >=0.4.24<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/proxy/Initializable.sol#4)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC721/ERC721MetadataUpgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC721/ERC721ReceiverUpgradeable.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC721/IERC721Upgradeable.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/StringsUpgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/access/Ownable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/proxy/BeaconProxy.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/proxy/IBeacon.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/proxy/Proxy.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts/utils/Address.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@rarible/lazy-mint/contracts/erc-721/IERC721LazyMint.sol#3)
        - v2 (node_modules/@rarible/lazy-mint/contracts/erc-721/ERC721LazyMint.sol#4)
        - >=0.6.2<0.8.0 (node_modules/@rarible/lazy-mint/contracts/erc-721/LibERC721LazyMint.sol#3)
        - ^0.7.0 (node_modules/@rarible/libraries/contracts/LibSignature.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@rarible/royalties-upgradeable/contracts/RoyaltiesV2Upgradeable.sol#3)
        - v2 (node_modules/@rarible/royalties-upgradeable/contracts/RoyaltiesV2Upgradeable.sol#4)
        - >=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/LibPart.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/LibRoyaltiesV2.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/RoyaltiesV2.sol#3)
        - v2 (node_modules/@rarible/royalties/contracts/RoyaltiesV2.sol#4)
        - >=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/impl/AbstractRoyalties.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/impl/RoyaltiesV2Impl.sol#3)
        - v2 (node_modules/@rarible/royalties/contracts/impl/RoyaltiesV2Impl.sol#4)
        - 0.7.6 (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#3)
        - 0.7.6 (contracts/HasContractURI.sol#3)
        - 0.7.6 (contracts/LibURI.sol#3)
        - 0.7.6 (contracts/Mint721Validator.sol#3)
        - >=0.6.2<0.8.0 (contracts/create-2/ERC721RaribleFactoryC2.sol#3)
        - v2 (contracts/create-2/ERC721RaribleFactoryC2.sol#4)
        - 0.7.6 (contracts/erc-1271/ERC1271.sol#3)
        - 0.7.6 (contracts/erc-1271/ERC1271Validator.sol#3)
        - 0.7.6 (contracts/erc-721-minimal/ERC721BaseMinimal.sol#3)
        - v2 (contracts/erc-721-minimal/ERC721BaseMinimal.sol#4)
        - >=0.6.0<0.8.0 (contracts/erc-721-minimal/ERC721BurnableUpgradeableMinimal.sol#3)
        - 0.7.6 (contracts/erc-721-minimal/ERC721DefaultApprovalMinimal.sol#3)
        - 0.7.6 (contracts/erc-721-minimal/ERC721LazyMinimal.sol#3)
        - v2 (contracts/erc-721-minimal/ERC721LazyMinimal.sol#4)
        - 0.7.6 (contracts/erc-721-minimal/ERC721RaribleMinimal.sol#3)
        - v2 (contracts/erc-721-minimal/ERC721RaribleMinimal.sol#4)
        - >=0.6.0<0.8.0 (contracts/erc-721-minimal/ERC721URI.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

AbstractRoyalties._onRoyaltiesSet(uint256,LibPart.Part[]) (node_modules/@rarible/royalties/contracts/impl/AbstractRoyalties.sol#31) is never used and should be removed
Address.functionCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#79-81) is never used and should be removed
Address.functionCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#89-91) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#104-106) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#114-121) is never used and should be removed
Address.functionDelegateCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#153-155) is never used and should be removed
Address.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#129-131) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#139-145) is never used and should be removed
Address.sendValue(address,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#53-59) is never used and should be removed
AddressUpgradeable._verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#147-164) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#79-81) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#89-91) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#104-106) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#114-121) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#129-131) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#139-145) is never used and should be removed
AddressUpgradeable.sendValue(address,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#53-59) is never used and should be removed
Context._msgData() (node_modules/@openzeppelin/contracts/utils/Context.sol#20-23) is never used and should be removed
ContextUpgradeable.__Context_init() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#17-19) is never used and should be removed
ContextUpgradeable._msgData() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#27-30) is never used and should be removed
EIP712Upgradeable.__EIP712_init(string,string) (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#44-46) is never used and should be removed
ERC1271.returnIsValidSignatureMagicNumber(bool) (contracts/erc-1271/ERC1271.sol#22-24) is never used and should be removed
ERC165Upgradeable.__ERC165_init() (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#25-27) is never used and should be removed
ERC721BurnableUpgradeableMinimal.__ERC721Burnable_init() (contracts/erc-721-minimal/ERC721BurnableUpgradeableMinimal.sol#14-18) is never used and should be removed
ERC721UpgradeableMinimal.__ERC721_init(string,string) (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#71-75) is never used and should be removed
ERC721UpgradeableMinimal._clearMetadata(uint256) (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#347-348) is never used and should be removed
ERC721UpgradeableMinimal._emitMintEvent(address,uint256) (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#310-312) is never used and should be removed
HasContractURI._setContractURI(string) (contracts/HasContractURI.sol#25-27) is never used and should be removed
OwnableUpgradeable.__Ownable_init() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#27-30) is never used and should be removed
Proxy._implementation() (node_modules/@openzeppelin/contracts/proxy/Proxy.sol#49) is never used and should be removed
SafeMathUpgradeable.div(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#135-138) is never used and should be removed
SafeMathUpgradeable.div(uint256,uint256,string) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#190-193) is never used and should be removed
SafeMathUpgradeable.mod(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#152-155) is never used and should be removed
SafeMathUpgradeable.mod(uint256,uint256,string) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#210-213) is never used and should be removed
SafeMathUpgradeable.mul(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#116-121) is never used and should be removed
SafeMathUpgradeable.sub(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#101-104) is never used and should be removed
SafeMathUpgradeable.sub(uint256,uint256,string) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#170-173) is never used and should be removed
SafeMathUpgradeable.tryAdd(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#24-28) is never used and should be removed
SafeMathUpgradeable.tryDiv(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#60-63) is never used and should be removed
SafeMathUpgradeable.tryMod(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#70-73) is never used and should be removed
SafeMathUpgradeable.tryMul(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#45-53) is never used and should be removed
SafeMathUpgradeable.trySub(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#35-38) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
```

AUTOMATED TESTING

```
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/introspection/IERC165Upgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#3) is too complex
Pragma version>=0.4.24<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/proxy/Initializable.sol#4) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC721/IERC721MetadataUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC721/IERC721ReceiverUpgradeable.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC721/IERC721Upgradeable.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/StringsUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/access/Ownable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/proxy/BeaconProxy.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/proxy/IBeacon.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/proxy/Proxy.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts/utils/Address.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/lazy-mint/contracts/erc-721/IERC721LazyMint.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/lazy-mint/contracts/erc-721/LibERC721LazyMint.sol#3) is too complex
Pragma version^0.7.0 (node_modules/@rarible/libraries/contracts/LibSignature.sol#3) allows old versions
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties-upgradeable/contracts/RoyaltiesV2Upgradeable.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/LibPart.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/LibRoyaltiesV2.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/RoyaltiesV2.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/impl/AbstractRoyalties.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/impl/RoyaltiesV2Impl.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (contracts/create-2/ERC721RaribleFactoryC2.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (contracts/erc-721-minimal/ERC721BurnableUpgradeableMinimal.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (contracts/erc-721-minimal/ERC721URI.sol#3) is too complex
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

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

Function OwnableUpgradeable.__Ownable_init() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#27-30) is not in mixedCase
Function OwnableUpgradeable.__Ownable_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#32-36) is not in mixedCase
Variable OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#76) is not in mixedCase
Function EIP712Upgradeable.__EIP712_init(string,string) (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#44-46) is not in mixedCase
Function EIP712Upgradeable.__EIP712_init_unchained(string,string) (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#48-53) is not in mixedCase
Function EIP712Upgradeable._EIP712NameHash() (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#107-109) is not in mixedCase
Function EIP712Upgradeable._EIP712VersionHash() (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#117-119) is not in mixedCase
Variable EIP712Upgradeable._HASHED_NAME (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#27) is not in mixedCase
Variable EIP712Upgradeable._HASHED_VERSION (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#28) is not in mixedCase
Variable EIP712Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#120) is not in mixedCase
Function ERC165Upgradeable.__ERC165_init() (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#25-27) is not in mixedCase
Function ERC165Upgradeable.__ERC165_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#29-33) is not in mixedCase
Variable ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59) is not in mixedCase
Function ContextUpgradeable.__Context_init() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#17-19) is not in mixedCase

Function ContextUpgradeable.__Context_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#21-22) is not in mixedCase
Variable ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31) is not in mixedCase
Function RoyaltiesV2Upgradeable.__RoyaltiesV2Upgradeable_init_unchained() (node_modules/@rarible/royalties-upgradeable/contracts/RoyaltiesV2Upgradeable.sol#11-13) is not in mixedCase
Function ERC721UpgradeableMinimal.__ERC721_init(string,string) (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#71-75) is not in mixedCase
Function ERC721UpgradeableMinimal.__ERC721_init_unchained(string,string) (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#77-84) is not in mixedCase
Parameter ERC721UpgradeableMinimal.safeTransferFrom(address,address,uint256,bytes)._data (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#196) is not in mixedCase
Variable ERC721UpgradeableMinimal.__gap (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#443) is not in mixedCase
Function HasContractURI.__HasContractURI_init_unchained(string) (contracts/HasContractURI.sol#16-19) is not in mixedCase
Parameter HasContractURI.__HasContractURI_init_unchained(string)._contractURI (contracts/HasContractURI.sol#16) is not in mixedCase
Variable HasContractURI.__gap (contracts/HasContractURI.sol#20) is not in mixedCase
Parameter LibURI.checkPrefix(string,string)._tokenURI (contracts/LibURI.sol#7) is not in mixedCase
Function Mint721Validator.__Mint721Validator_init_unchained() (contracts/Mint721Validator.sol#9-11) is not in mixedCase
Variable Mint721Validator.__gap (contracts/Mint721Validator.sol#16) is not in mixedCase
Parameter ERC721RaribleFactoryC2.createToken(string,string,string,string,uint256)._name (contracts/create-2/ERC721RaribleFactoryC2.sol#30) is not in mixedCase
Parameter ERC721RaribleFactoryC2.createToken(string,string,string,string,uint256)._symbol (contracts/create-2/ERC721RaribleFactoryC2.sol#30) is not in mixedCase
Parameter ERC721RaribleFactoryC2.createToken(string,string,string,string,address[],uint256)._name (contracts/create-2/ERC721RaribleFactoryC2.sol#37) is not in mixedCase
Parameter ERC721RaribleFactoryC2.createToken(string,string,string,string,address[],uint256)._symbol (contracts/create-2/ERC721RaribleFactoryC2.sol#37) is not in mixedCase
Parameter ERC721RaribleFactoryC2.getCreationBytecode(bytes)._data (contracts/create-2/ERC721RaribleFactoryC2.sol#56) is not in mixedCase
Parameter ERC721RaribleFactoryC2.getAddress(string,string,string,string,uint256)._name (contracts/create-2/ERC721RaribleFactoryC2.sol#61) is not in mixedCase
Parameter ERC721RaribleFactoryC2.getAddress(string,string,string,string,uint256)._symbol (contracts/create-2/ERC721RaribleFactoryC2.sol#61) is not in mixedCase
Parameter ERC721RaribleFactoryC2.getAddress(string,string,string,string,uint256)._salt (contracts/create-2/ERC721RaribleFactoryC2.sol#61) is not in mixedCase
Parameter ERC721RaribleFactoryC2.getData(string,string,string,string)._name (contracts/create-2/ERC721RaribleFactoryC2.sol#75) is not in mixedCase
Parameter ERC721RaribleFactoryC2.getData(string,string,string,string)._symbol (contracts/create-2/ERC721RaribleFactoryC2.sol#75) is not in mixedCase
Parameter ERC721RaribleFactoryC2.getAddress(string,string,string,string,address[],uint256)._name (contracts/create-2/ERC721RaribleFactoryC2.sol#80) is not in mixedCase
Parameter ERC721RaribleFactoryC2.getAddress(string,string,string,string,address[],uint256)._symbol (contracts/create-2/ERC721RaribleFactoryC2.sol#80) is not in mixedCase
Parameter ERC721RaribleFactoryC2.getAddress(string,string,string,string,address[],uint256)._salt (contracts/create-2/ERC721RaribleFactoryC2.sol#80) is not in mixedCase
Parameter ERC721RaribleFactoryC2.getData(string,string,string,string,address[])._name (contracts/create-2/ERC721RaribleFactoryC2.sol#94) is not in mixedCase
Parameter ERC721RaribleFactoryC2.getData(string,string,string,string,address[])._symbol (contracts/create-2/ERC721RaribleFactoryC2.sol#94) is not in mixedCase
Variable ERC1271Validator.__gap (contracts/erc-1271/ERC1271Validator.sol#94) is not in mixedCase
Variable ERC721BaseMinimal.__gap (contracts/erc-721-minimal/ERC721BaseMinimal.sol#38) is not in mixedCase
Function ERC721BurnableUpgradeableMinimal.__ERC721Burnable_init() (contracts/erc-721-minimal/ERC721BurnableUpgradeableMinimal.sol#14-18) is not in mixedCase
Function ERC721BurnableUpgradeableMinimal.__ERC721Burnable_init_unchained() (contracts/erc-721-minimal/ERC721BurnableUpgradeableMinimal.sol#20-21) is not in mixedCase
Variable ERC721BurnableUpgradeableMinimal.__gap (contracts/erc-721-minimal/ERC721BurnableUpgradeableMinimal.sol#41) is not in mixedCase
Variable ERC721DefaultApprovalMinimal.__gap (contracts/erc-721-minimal/ERC721DefaultApprovalMinimal.sol#24) is not in mixedCase
Function ERC721LazyMinimal.__ERC721Lazy_init_unchained() (contracts/erc-721-minimal/ERC721LazyMinimal.sol#25-27) is not in mixedCase
Parameter ERC721LazyMinimal.updateAccount(uint256,address,address)._id (contracts/erc-721-minimal/ERC721LazyMinimal.sol#95) is not in mixedCase
Parameter ERC721LazyMinimal.updateAccount(uint256,address,address)._from (contracts/erc-721-minimal/ERC721LazyMinimal.sol#95) is not in mixedCase
Parameter ERC721LazyMinimal.updateAccount(uint256,address,address)._to (contracts/erc-721-minimal/ERC721LazyMinimal.sol#95) is not in mixedCase
Parameter ERC721LazyMinimal.getCreators(uint256)._id (contracts/erc-721-minimal/ERC721LazyMinimal.sol#100) is not in mixedCase
Variable ERC721LazyMinimal.__gap (contracts/erc-721-minimal/ERC721LazyMinimal.sol#112) is not in mixedCase
Function ERC721RaribleMinimal.__ERC721RaribleUser_init(string,string,string,string,address[],address,address) (contracts/erc-721-minimal/ERC721RaribleMinimal.sol#15-24) is not in mixedCase
Parameter ERC721RaribleMinimal.__ERC721RaribleUser_init(string,string,string,string,address[],address,address)._name (contracts/erc-721-minimal/ERC721RaribleMinimal.sol#15) is not in mixedCase
Parameter ERC721RaribleMinimal.__ERC721RaribleUser_init(string,string,string,string,address[],address,address)._symbol (contracts/erc-721-minimal/ERC721RaribleMinimal.sol#15) is not in mixedCase
Function ERC721RaribleMinimal.__ERC721Rarible_init(string,string,string,string,address,address) (contracts/erc-721-minimal/ERC721RaribleMinimal.sol#26-31) is not in mixedCase
Parameter ERC721RaribleMinimal.__ERC721Rarible_init(string,string,string,string,address,address)._name (contracts/erc-721-minimal/ERC721RaribleMinimal.sol#26) is not in mixedCase
Parameter ERC721RaribleMinimal.__ERC721Rarible_init(string,string,string,string,address,address)._symbol (contracts/erc-721-minimal/ERC721RaribleMinimal.sol#26) is not in mixedCase
Function ERC721RaribleMinimal.__ERC721Rarible_init_unchained(string,string,string,string,address,address) (contracts/erc-721-minimal/ERC721RaribleMinimal.sol#33-48) is not in mixedCase
Parameter ERC721RaribleMinimal.__ERC721Rarible_init_unchained(string,string,string,string,address,address)._name (contracts/erc-721-minimal/ERC721RaribleMinimal.sol#33) is not in mixedCase
Parameter ERC721RaribleMinimal.__ERC721Rarible_init_unchained(string,string,string,string,address,address)._symbol (contracts/erc-721-minimal/ERC721RaribleMinimal.sol#33) is not in mixedCase
Variable ERC721RaribleMinimal.__gap (contracts/erc-721-minimal/ERC721RaribleMinimal.sol#57) is not in mixedCase
Variable ERC721URI.__gap (contracts/erc-721-minimal/ERC721URI.sol#92) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#94)" inEIP712Upgradeable (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#25-121)
Redundant expression "this (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#28)" inContextUpgradeable (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#16-32)
Redundant expression "this (node_modules/@openzeppelin/contracts/utils/Context.sol#21)" inContext (node_modules/@openzeppelin/contracts/utils/Context.sol#15-24)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

ERC721RaribleFactoryC2.getCreationBytecode(bytes) (contracts/create-2/ERC721RaribleFactoryC2.sol#56-58) uses literals with too many digits:
        - abi.encodePacked(type(!(BeaconProxy).creationCode,abi.encode(beacon,_data)) (contracts/create-2/ERC721RaribleFactoryC2.sol#57)
ERC1271.eliterConstructConstantVariables() (contracts/erc-1271/ERC1271.sol#5-26) uses literals with too many digits:
        - ERC1271_RETURN_INVALID_SIGNATURE = 0x00000000 (contracts/erc-1271/ERC1271.sol#9)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits

ERC1271 (contracts/erc-1271/ERC1271.sol#5-26) does not implement functions:
        - ERC1271.isValidSignature(bytes32,bytes) (contracts/erc-1271/ERC1271.sol#20)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unimplemented-functions

LibERC721LazyMint.__INTERFACE_ID_MINT_AND_TRANSFER (node_modules/@rarible/lazy-mint/contracts/erc-721/LibERC721LazyMint.sol#9) is never used in LibERC721LazyMint (node_modules/@rarible/lazy-mint/contracts/erc-721/LibERC721LazyMint.sol#7-3
9)
ERC721RaribleMinimal.__gap (contracts/erc-721-minimal/ERC721RaribleMinimal.sol#57) is never used in ERC721RaribleMinimal (contracts/erc-721-minimal/ERC721RaribleMinimal.sol#8-58)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

renounceOwnership() should be declared external:
        - OwnableUpgradeable.renounceOwnership() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#60-63)
transferOwnership(address) should be declared external:
        - OwnableUpgradeable.transferOwnership(address) (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#69-73)
renounceOwnership() should be declared external:
        - Ownable.renounceOwnership() (node_modules/@openzeppelin/contracts/access/Ownable.sol#54-57)
transferOwnership(address) should be declared external:
        - Ownable.transferOwnership(address) (node_modules/@openzeppelin/contracts/access/Ownable.sol#63-67)
balanceOf(address) should be declared external:
        - ERC721UpgradeableMinimal.balanceOf(address) (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#89-92)
name() should be declared external:
        - ERC721UpgradeableMinimal.name() (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#106-108)
symbol() should be declared external:
        - ERC721UpgradeableMinimal.symbol() (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#113-115)
approve(address,uint256) should be declared external:
        - ERC721UpgradeableMinimal.approve(address,uint256) (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#126-136)
transferFrom(address,address,uint256) should be declared external:
        - ERC721UpgradeableMinimal.transferFrom(address,address,uint256) (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#167-176)
getAddress(string,string,string,string,uint256) should be declared external:
        - ERC721RaribleFactoryC2.getAddress(string,string,string,string,uint256) (contracts/create-2/ERC721RaribleFactoryC2.sol#61-73)
getAddress(string,string,string,string,address[],uint256) should be declared external:
        - ERC721RaribleFactoryC2.getAddress(string,string,string,string,address[],uint256) (contracts/create-2/ERC721RaribleFactoryC2.sol#80-92)
isValidSignature(bytes32,bytes) should be declared external:
        - ERC1271.isValidSignature(bytes32,bytes) (contracts/erc-1271/ERC1271.sol#20)
burn(uint256) should be declared external:
        - ERC721BurnableUpgradeableMinimal.burn(uint256) (contracts/erc-721-minimal/ERC721BurnableUpgradeableMinimal.sol#29-39)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

ERC721WoonklyNFTRevealWaveFactoryC2.sol

```
OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#74) shadows:
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
ERC721UpgradeableMinimal.__gap (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#443) shadows:
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
HasContractURI.__gap (contracts/HasContractURI.sol#29) shadows:
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
Mint721Validator.__gap (contracts/Mint721Validator.sol#16) shadows:
        - ERC1271Validator.__gap (contracts/erc-1271/ERC1271Validator.sol#34)
        - EIP712Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#120)
ERC1271Validator.__gap (contracts/erc-1271/ERC1271Validator.sol#34) shadows:
        - EIP712Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#120)
ERC721BaseMinimal.__gap (contracts/erc-721-minimal/ERC721BaseMinimal.sol#38) shadows:
        - HasContractURI.__gap (contracts/HasContractURI.sol#29)
        - ERC721LazyMinimal.__gap (contracts/erc-721-minimal/ERC721LazyMinimal.sol#112)
        - ERC721URI.__gap (contracts/erc-721-minimal/ERC721URI.sol#92)
        - Mint721Validator.__gap (contracts/Mint721Validator.sol#16)
        - ERC721BurnableUpgradeableMinimal.__gap (contracts/erc-721-minimal/ERC721BurnableUpgradeableMinimal.sol#41)
        - ERC721DefaultApprovalMinimal.__gap (contracts/erc-721-minimal/ERC721DefaultApprovalMinimal.sol#24)
        - ERC721UpgradeableMinimal.__gap (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#443)
        - ERC1271Validator.__gap (contracts/erc-1271/ERC1271Validator.sol#34)
        - EIP712Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#120)
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#74)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
ERC721BurnableUpgradeableMinimal.__gap (contracts/erc-721-minimal/ERC721BurnableUpgradeableMinimal.sol#41) shadows:
        - ERC721UpgradeableMinimal.__gap (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#443)
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
ERC721DefaultApprovalMinimal.__gap (contracts/erc-721-minimal/ERC721DefaultApprovalMinimal.sol#24) shadows:
        - ERC721UpgradeableMinimal.__gap (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#443)
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
ERC721LazyMinimal.__gap (contracts/erc-721-minimal/ERC721LazyMinimal.sol#112) shadows:
        - ERC721URI.__gap (contracts/erc-721-minimal/ERC721URI.sol#92)
        - Mint721Validator.__gap (contracts/Mint721Validator.sol#16)
        - ERC721UpgradeableMinimal.__gap (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#443)
        - ERC1271Validator.__gap (contracts/erc-1271/ERC1271Validator.sol#34)
        - EIP712Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#120)
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
ERC721LazyMinimal._INTERFACE_ID_ERC165 (contracts/erc-721-minimal/ERC721LazyMinimal.sol#17) shadows:
        - ERC165Upgradeable._INTERFACE_ID_ERC165 (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#18)
ERC721LazyMinimal._INTERFACE_ID_ERC721 (contracts/erc-721-minimal/ERC721LazyMinimal.sol#18) shadows:
        - ERC721UpgradeableMinimal._INTERFACE_ID_ERC721 (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#54)
ERC721LazyMinimal._INTERFACE_ID_ERC721_METADATA (contracts/erc-721-minimal/ERC721LazyMinimal.sol#19) shadows:
        - ERC721UpgradeableMinimal._INTERFACE_ID_ERC721_METADATA (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#63)
ERC721RaribleMinimalRevealWave.__gap (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#211) shadows:
        - ERC721BaseMinimal.__gap (contracts/erc-721-minimal/ERC721BaseMinimal.sol#38)
        - HasContractURI.__gap (contracts/HasContractURI.sol#29)
        - ERC721LazyMinimal.__gap (contracts/erc-721-minimal/ERC721LazyMinimal.sol#112)
        - ERC721URI.__gap (contracts/erc-721-minimal/ERC721URI.sol#92)
        - Mint721Validator.__gap (contracts/Mint721Validator.sol#16)
        - ERC721BurnableUpgradeableMinimal.__gap (contracts/erc-721-minimal/ERC721BurnableUpgradeableMinimal.sol#41)
        - ERC721DefaultApprovalMinimal.__gap (contracts/erc-721-minimal/ERC721DefaultApprovalMinimal.sol#24)
        - ERC721UpgradeableMinimal.__gap (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#443)
        - ERC1271Validator.__gap (contracts/erc-1271/ERC1271Validator.sol#34)
        - EIP712Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#120)
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#74)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
ERC721URI.__gap (contracts/erc-721-minimal/ERC721URI.sol#92) shadows:
        - ERC721UpgradeableMinimal.__gap (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#443)
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variable-shadowing

ERC721RaribleMinimalRevealWave.tokenURI(uint256)._tokenURI (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#158) is a local variable never initialized
ERC1271Validator.validate1271(address,bytes32,bytes).signerFromSig (contracts/erc-1271/ERC1271Validator.sol#19) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

BeaconProxy._setBeacon(address,bytes) (node_modules/@openzeppelin/contracts/proxy/BeaconProxy.sol#68-87) ignores return value by Address.functionDelegateCall(_implementation(),data,BeaconProxy: function call failed) (node_modules/@openze
ppelin/contracts/proxy/BeaconProxy.sol#85)
ERC721UpgradeableMinimal._checkOnERC721Received(address,address,uint256,bytes) (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#401-422) ignores return value by IERC721ReceiverUpgradeable(to).onERC721R
eceived(_msgSender(),from,tokenId,_data) (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#408-418)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

ERC721BaseMinimal.isApprovedForAll(address,address).owner (contracts/erc-721-minimal/ERC721BaseMinimal.sol#18) shadows:
        - OwnableUpgradeable.owner() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#41-43) (function)
ERC721RaribleMinimalRevealWave._ERC721RaribleUser_init(string,string,string,string,address[],address,address)._name (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#29) shadows:
        - ERC721UpgradeableMinimal._name (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#23) (state variable)
ERC721RaribleMinimalRevealWave._ERC721RaribleUser_init(string,string,string,string,address[],address,address)._symbol (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#29) shadows:
        - ERC721UpgradeableMinimal._symbol (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#26) (state variable)
ERC721RaribleMinimalRevealWave._ERC721RaribleUser_init(string,string,string,string,address[],address,address).baseURI (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#29) shadows:
        - ERC721URI.baseURI() (contracts/erc-721-minimal/ERC721URI.sol#50-52) (function)
ERC721RaribleMinimalRevealWave._ERC721RaribleUser_init(string,string,string,string,address[],address,address).contractURI (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#29) shadows:
        - HasContractURI.contractURI (contracts/HasContractURI.sol#9) (state variable)
ERC721RaribleMinimalRevealWave._ERC721Rarible_init(string,string,string,string,address,address)._name (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#40) shadows:
        - ERC721UpgradeableMinimal._name (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#23) (state variable)
ERC721RaribleMinimalRevealWave._ERC721Rarible_init(string,string,string,string,address,address)._symbol (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#40) shadows:
        - ERC721UpgradeableMinimal._symbol (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#26) (state variable)
ERC721RaribleMinimalRevealWave._ERC721Rarible_init(string,string,string,string,address,address).baseURI (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#40) shadows:
        - ERC721URI.baseURI() (contracts/erc-721-minimal/ERC721URI.sol#50-52) (function)
ERC721RaribleMinimalRevealWave._ERC721Rarible_init(string,string,string,string,address,address).contractURI (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#40) shadows:
        - HasContractURI.contractURI (contracts/HasContractURI.sol#9) (state variable)
ERC721RaribleMinimalRevealWave._ERC721Rarible_init_unchained(string,string,string,string,address,address)._name (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#47) shadows:
        - ERC721UpgradeableMinimal._name (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#23) (state variable)
ERC721RaribleMinimalRevealWave._ERC721Rarible_init_unchained(string,string,string,string,address,address)._symbol (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#47) shadows:
        - ERC721UpgradeableMinimal._symbol (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#26) (state variable)
ERC721RaribleMinimalRevealWave._ERC721Rarible_init_unchained(string,string,string,string,address,address).baseURI (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#47) shadows:
        - ERC721URI.baseURI() (contracts/erc-721-minimal/ERC721URI.sol#50-52) (function)
ERC721RaribleMinimalRevealWave._ERC721Rarible_init_unchained(string,string,string,string,address,address).contractURI (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#47) shadows:
        - HasContractURI.contractURI (contracts/HasContractURI.sol#9) (state variable)
ERC721RaribleMinimalRevealWave.setRevealWave(uint256,string,string,bool,uint256)._name (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#72) shadows:
        - ERC721UpgradeableMinimal._name (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#23) (state variable)
ERC721RaribleMinimalRevealWave.changeHiddenBaseURI(uint256,string)._baseURI (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#87) shadows:
        - ERC721URI._baseURI (contracts/erc-721-minimal/ERC721URI.sol#22) (state variable)
ERC721RaribleMinimalRevealWave.changeName(uint256,string)._name (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#132) shadows:
        - ERC721UpgradeableMinimal._name (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#23) (state variable)
ERC721RaribleMinimalRevealWave.tokenURI(uint256)._baseURI (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#159) shadows:
        - ERC721URI._baseURI (contracts/erc-721-minimal/ERC721URI.sol#22) (state variable)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

ERC721RaribleRevealWaveFactoryC2.constructor(address,address,address)._beacon (contracts/create-2/ERC721RaribleRevealWaveFactoryC2.sol#25) lacks a zero-check on :
                - beacon = _beacon (contracts/create-2/ERC721RaribleRevealWaveFactoryC2.sol#26)
ERC721RaribleRevealWaveFactoryC2.constructor(address,address,address)._transferProxy (contracts/create-2/ERC721RaribleRevealWaveFactoryC2.sol#25) lacks a zero-check on :
                - transferProxy = _transferProxy (contracts/create-2/ERC721RaribleRevealWaveFactoryC2.sol#27)
ERC721RaribleRevealWaveFactoryC2.constructor(address,address,address)._lazyTransferProxy (contracts/create-2/ERC721RaribleRevealWaveFactoryC2.sol#25) lacks a zero-check on :
                - lazyTransferProxy = _lazyTransferProxy (contracts/create-2/ERC721RaribleRevealWaveFactoryC2.sol#28)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation

Variable 'ERC721UpgradeableMinimal._checkOnERC721Received(address,address,uint256,bytes).retval (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#408)' in ERC721UpgradeableMinimal._checkOnERC721Received
(address,address,uint256,bytes) (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#401-422) potentially used before declaration: retval == IERC721ReceiverUpgradeable.onERC721Received.selector (node_modul
es/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#409)
Variable 'ERC721UpgradeableMinimal._checkOnERC721Received(address,address,uint256,bytes).reason (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#410)' in ERC721UpgradeableMinimal._checkOnERC721Received
(address,address,uint256,bytes) (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#401-422) potentially used before declaration: reason.length == 0 (node_modules/@rarible/tokens-minimal/contracts/erc-721
/ERC721UpgradeableMinimal.sol#411)
Variable 'ERC721UpgradeableMinimal._checkOnERC721Received(address,address,uint256,bytes).reason (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#410)' in ERC721UpgradeableMinimal._checkOnERC721Received
(address,address,uint256,bytes) (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#401-422) potentially used before declaration: revert(uint256,uint256)(32 + reason,mload(uint256)(reason)) (node_modules/
@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#415)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables

Reentrancy in ERC721LazyMinimal.mintAndTransfer(LibERC721LazyMint.Mint721Data,address) (contracts/erc-721-minimal/ERC721LazyMinimal.sol#50-70):
        External calls:
        - _safeMint(to,data.tokenId) (contracts/erc-721-minimal/ERC721LazyMinimal.sol#66)
                - IERC721ReceiverUpgradeable(to).onERC721Received(_msgSender(),from,tokenId,_data) (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#408-418)
        State variables written after the call(s):
        - _setTokenURI(data.tokenId,data.tokenURI) (contracts/erc-721-minimal/ERC721LazyMinimal.sol#69)
                - _tokenURIs[tokenId] = _tokenURI (contracts/erc-721-minimal/ERC721URI.sol#33)
        - _saveCreators(data.tokenId,data.creators) (contracts/erc-721-minimal/ERC721LazyMinimal.sol#68)
                - creatorsOfToken.push(_creators[i]) (contracts/erc-721-minimal/ERC721LazyMinimal.sol#88)
        - _saveRoyalties(data.tokenId,data.royalties) (contracts/erc-721-minimal/ERC721LazyMinimal.sol#67)
                - royalties[id].push(_royalties[i]) (node_modules/@rarible/royalties/contracts/impl/AbstractRoyalties.sol#16)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2

Reentrancy in ERC721RaribleRevealWaveFactoryC2.createToken(string,string,string,string,uint256) (contracts/create-2/ERC721RaribleRevealWaveFactoryC2.sol#31-37):
        External calls:
        - token.transferOwnership(_msgSender()) (contracts/create-2/ERC721RaribleRevealWaveFactoryC2.sol#35)
        Event emitted after the call(s):
        - Create721RaribleProxy(beaconProxy) (contracts/create-2/ERC721RaribleRevealWaveFactoryC2.sol#36)
Reentrancy in ERC721RaribleRevealWaveFactoryC2.createToken(string,string,string,string,address[],uint256) (contracts/create-2/ERC721RaribleRevealWaveFactoryC2.sol#39-45):
        External calls:
        - token.transferOwnership(_msgSender()) (contracts/create-2/ERC721RaribleRevealWaveFactoryC2.sol#43)
        Event emitted after the call(s):
        - Create721RaribleUserProxy(beaconProxy) (contracts/create-2/ERC721RaribleRevealWaveFactoryC2.sol#44)
Reentrancy in ERC721LazyMinimal.mintAndTransfer(LibERC721LazyMint.Mint721Data,address) (contracts/erc-721-minimal/ERC721LazyMinimal.sol#50-70):
        External calls:
        - _safeMint(to,data.tokenId) (contracts/erc-721-minimal/ERC721LazyMinimal.sol#66)
                - IERC721ReceiverUpgradeable(to).onERC721Received(_msgSender(),from,tokenId,_data) (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#408-418)
        Event emitted after the call(s):
        - Creators(tokenId,_creators) (contracts/erc-721-minimal/ERC721LazyMinimal.sol#92)
                - _saveCreators(data.tokenId,data.creators) (contracts/erc-721-minimal/ERC721LazyMinimal.sol#68)
        - RoyaltiesSet(id,_royalties) (node_modules/@rarible/royalties/contracts/impl/RoyaltiesV2Impl.sol#46)
                - _saveRoyalties(data.tokenId,data.royalties) (contracts/erc-721-minimal/ERC721LazyMinimal.sol#67)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3
```

AUTOMATED TESTING

```
EIP712Upgradeable._getChainId() (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#93-99) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#96-98)
AddressUpgradeable.isContract(address) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#26-35) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#33)
AddressUpgradeable._verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#147-164) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#156-159)
BeaconProxy._beacon() (node_modules/@openzeppelin/contracts/proxy/BeaconProxy.sol#43-49) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/proxy/BeaconProxy.sol#46-48)
BeaconProxy._setBeacon(address,bytes) (node_modules/@openzeppelin/contracts/proxy/BeaconProxy.sol#68-87) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/proxy/BeaconProxy.sol#80-82)
Proxy._delegate(address) (node_modules/@openzeppelin/contracts/proxy/Proxy.sol#21-41) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/proxy/Proxy.sol#23-40)
Address.isContract(address) (node_modules/@openzeppelin/contracts/utils/Address.sol#26-35) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/utils/Address.sol#33)
Address._verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#171-188) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/utils/Address.sol#180-183)
LibSignature.recover(bytes32,bytes) (node_modules/@rarible/libraries/contracts/LibSignature.sol#20-45) uses assembly
        - INLINE ASM (node_modules/@rarible/libraries/contracts/LibSignature.sol#38-42)
ERC721UpgradeableMinimal._checkOnERC721Received(address,address,uint256,bytes) (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#401-422) uses assembly
        - INLINE ASM (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#414-416)
ERC721RaribleRevealWaveFactoryC2.deployProxy(bytes,uint256) (contracts/create-2/ERC721RaribleRevealWaveFactoryC2.sol#48-56) uses assembly
        - INLINE ASM (contracts/create-2/ERC721RaribleRevealWaveFactoryC2.sol#50-55)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

ERC721RaribleMinimalRevealWave.tokenURI(uint256) (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#152-180) compares to a boolean constant:
        -revealWave.isRevealed == true || revealWave.addTokenURIToHiddenBaseURI == true (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#161)
ERC721RaribleMinimalRevealWave.tokenURI(uint256) (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#152-180) compares to a boolean constant:
        -revealWave.isRevealed == false (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#166)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#boolean-equality

Different versions of Solidity is used:
        - Version used: ['0.7.6', '>=0.4.24<0.8.0', '>=0.6.0<0.8.0', '>=0.6.2<0.8.0', '^0.7.0']
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/introspection/IERC165Upgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#3)
        - >=0.4.24<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/proxy/Initializable.sol#4)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC721/ERC721MetadataUpgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC721/ERC721ReceiverUpgradeable.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC721/IERC721Upgradeable.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/StringsUpgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/access/Ownable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/proxy/BeaconProxy.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/proxy/IBeacon.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/proxy/Proxy.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/Address.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@rarible/lazy-mint/contracts/erc-721/IERC721LazyMint.sol#3)
        - v2 (node_modules/@rarible/lazy-mint/contracts/erc-721/IERC721LazyMint.sol#4)
        - >=0.6.2<0.8.0 (node_modules/@rarible/lazy-mint/contracts/erc-721/LibERC721LazyMint.sol#3)
        - ^0.7.0 (node_modules/@rarible/libraries/contracts/LibSignature.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@rarible/royalties-upgradeable/contracts/RoyaltiesV2Upgradeable.sol#3)
        - v2 (node_modules/@rarible/royalties-upgradeable/contracts/RoyaltiesV2Upgradeable.sol#4)
        - >=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/LibPart.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/LibRoyaltiesV2.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/RoyaltiesV2.sol#3)
        - v2 (node_modules/@rarible/royalties/contracts/RoyaltiesV2.sol#4)
        - >=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/impl/AbstractRoyalties.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/impl/RoyaltiesV2Impl.sol#3)
        - v2 (node_modules/@rarible/royalties/contracts/impl/RoyaltiesV2Impl.sol#4)
        - 0.7.6 (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#3)
        - 0.7.6 (contracts/HasContractURI.sol#3)
        - 0.7.6 (contracts/LibURI.sol#3)
        - 0.7.6 (contracts/Mint721Validator.sol#3)
        - >=0.6.2<0.8.0 (contracts/create-2/ERC721RaribleRevealWaveFactoryC2.sol#3)
        - v2 (contracts/create-2/ERC721RaribleRevealWaveFactoryC2.sol#4)
        - 0.7.6 (contracts/erc-1271/ERC1271.sol#3)
        - 0.7.6 (contracts/erc-1271/ERC1271Validator.sol#3)
        - 0.7.6 (contracts/erc-721-minimal/ERC721BaseMinimal.sol#3)
        - v2 (contracts/erc-721-minimal/ERC721BaseMinimal.sol#4)
        - >=0.6.0<0.8.0 (contracts/erc-721-minimal/ERC721BurnableUpgradeableMinimal.sol#3)
        - 0.7.6 (contracts/erc-721-minimal/ERC721DefaultApprovalMinimal.sol#3)
        - 0.7.6 (contracts/erc-721-minimal/ERC721LazyMinimal.sol#3)
        - v2 (contracts/erc-721-minimal/ERC721LazyMinimal.sol#4)
        - 0.7.6 (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#3)
        - v2 (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#4)
        - >=0.6.0<0.8.0 (contracts/erc-721-minimal/ERC721URI.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

AbstractRoyalties._onRoyaltiesSet(uint256,LibPart.Part[]) (node_modules/@rarible/royalties/contracts/impl/AbstractRoyalties.sol#31) is never used and should be removed
Address.functionCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#79-81) is never used and should be removed
Address.functionCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#89-91) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#104-106) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#114-121) is never used and should be removed
Address.functionDelegateCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#153-155) is never used and should be removed
Address.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#129-131) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#139-145) is never used and should be removed
Address.sendValue(address,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#53-59) is never used and should be removed
AddressUpgradeable._verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#147-164) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#79-81) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#89-91) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#104-106) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#114-121) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#129-131) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#139-145) is never used and should be removed
AddressUpgradeable.sendValue(address,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#53-59) is never used and should be removed
Context._msgData() (node_modules/@openzeppelin/contracts/utils/Context.sol#20-23) is never used and should be removed
ContextUpgradeable.__Context_init() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#17-19) is never used and should be removed
ContextUpgradeable._msgData() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#27-30) is never used and should be removed
EIP712Upgradeable.__EIP712_init(string,string) (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#44-46) is never used and should be removed
ERC1271.returnIsValidSignatureMagicNumber(bool) (contracts/erc-1271/ERC1271.sol#22-24) is never used and should be removed
ERC165Upgradeable.__ERC165_init() (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#25-27) is never used and should be removed
ERC721BurnableUpgradeableMinimal.__ERC721Burnable_init() (contracts/erc-721-minimal/ERC721BurnableUpgradeableMinimal.sol#14-18) is never used and should be removed
ERC721UpgradeableMinimal.__ERC721_init(string,string) (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#71-75) is never used and should be removed
ERC721UpgradeableMinimal._clearMetadata(uint256) (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#347-348) is never used and should be removed
ERC721UpgradeableMinimal._emitMintEvent(address,uint256) (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#310-312) is never used and should be removed
HasContractURI._setContractURI(string) (contracts/HasContractURI.sol#25-27) is never used and should be removed
OwnableUpgradeable.__Ownable_init() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#27-30) is never used and should be removed
Proxy._implementation() (node_modules/@openzeppelin/contracts/proxy/Proxy.sol#47) is never used and should be removed
SafeMathUpgradeable.div(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#135-138) is never used and should be removed
SafeMathUpgradeable.div(uint256,uint256,string) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#150-153) is never used and should be removed
SafeMathUpgradeable.mod(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#152-155) is never used and should be removed
SafeMathUpgradeable.mod(uint256,uint256,string) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#210-213) is never used and should be removed
SafeMathUpgradeable.mul(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#116-121) is never used and should be removed
SafeMathUpgradeable.sub(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#101-104) is never used and should be removed
SafeMathUpgradeable.sub(uint256,uint256,string) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#170-173) is never used and should be removed
SafeMathUpgradeable.tryAdd(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#24-28) is never used and should be removed
SafeMathUpgradeable.tryDiv(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#60-63) is never used and should be removed
SafeMathUpgradeable.tryMod(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#70-73) is never used and should be removed
SafeMathUpgradeable.tryMul(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#45-53) is never used and should be removed
SafeMathUpgradeable.trySub(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#35-38) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/introspection/IERC165Upgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#3) is too complex
Pragma version>=0.4.24<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/proxy/Initializable.sol#4) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC721/ERC721MetadataUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC721/ERC721ReceiverUpgradeable.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC721/IERC721Upgradeable.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/StringsUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/access/Ownable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/proxy/BeaconProxy.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/proxy/IBeacon.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/proxy/Proxy.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts/utils/Address.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/lazy-mint/contracts/erc-721/IERC721LazyMint.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/lazy-mint/contracts/erc-721/LibERC721LazyMint.sol#3) is too complex
Pragma version^0.7.0 (node_modules/@rarible/libraries/contracts/LibSignature.sol#3) allows old versions
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties-upgradeable/contracts/RoyaltiesV2Upgradeable.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/LibPart.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/LibRoyaltiesV2.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/RoyaltiesV2.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/impl/AbstractRoyalties.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/impl/RoyaltiesV2Impl.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (contracts/create-2/ERC721RaribleRevealWaveFactoryC2.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (contracts/erc-721-minimal/ERC721BurnableUpgradeableMinimal.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (contracts/erc-721-minimal/ERC721URI.sol#3) is too complex
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in AddressUpgradeable.sendValue(address,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#53-59):
        - (success) = recipient.call{value: amount}() (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#57)
Low level call in AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#114-121):
        - (success,returndata) = target.call{value: value}(data) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#115)
Low level call in AddressUpgradeable.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#139-145):
        - (success,returndata) = target.staticcall(data) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#143)
Low level call in Address.sendValue(address,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#53-59):
        - (success) = recipient.call{value: amount}() (node_modules/@openzeppelin/contracts/utils/Address.sol#57)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#114-121):
        - (success,returndata) = target.call{value: value}(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#119)
Low level call in Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#139-145):
        - (success,returndata) = target.staticcall(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#143)
Low level call in Address.functionDelegateCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#163-169):
        - (success,returndata) = target.delegatecall(data) (node_modules/@openzeppelin/contracts/utils/Address.sol#167)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
```

AUTOMATED TESTING

```
Function OwnableUpgradeable.__Ownable_init() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#27-30) is not in mixedCase
Function OwnableUpgradeable.__Ownable_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#32-36) is not in mixedCase
Variable OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#74) is not in mixedCase
Function EIP712Upgradeable.__EIP712_init(string,string) (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#44-46) is not in mixedCase
Function EIP712Upgradeable.__EIP712_init_unchained(string,string) (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#48-53) is not in mixedCase
Function EIP712Upgradeable._EIP712NameHash() (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#107-109) is not in mixedCase
Function EIP712Upgradeable._EIP712VersionHash() (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#117-119) is not in mixedCase
Variable EIP712Upgradeable._HASHED_NAME (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#27) is not in mixedCase
Variable EIP712Upgradeable._HASHED_VERSION (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#28) is not in mixedCase
Variable EIP712Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#120) is not in mixedCase
Function ERC165Upgradeable.__ERC165_init() (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#25-27) is not in mixedCase
Function ERC165Upgradeable.__ERC165_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#29-33) is not in mixedCase
Variable ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59) is not in mixedCase
Function ContextUpgradeable.__Context_init() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#17-19) is not in mixedCase
Function ContextUpgradeable.__Context_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#21-22) is not in mixedCase
Variable ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#51) is not in mixedCase
Function RoyaltiesV2UpgradeableB26._RoyaltiesV2Upgradeable_init_unchained() (node_modules/@rarible/royalties-upgradable/contracts/RoyaltiesV2Upgradeable.sol#11-13) is not in mixedCase
Function ERC721UpgradeableMinimal.__ERC721_init(string,string) (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#71-75) is not in mixedCase
Function ERC721UpgradeableMinimal.__ERC721_init_unchained(string,string) (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#77-84) is not in mixedCase
Parameter ERC721UpgradeableMinimal.safeTransferFrom(address,address,uint256,bytes)._data (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#196) is not in mixedCase
Variable ERC721UpgradeableMinimal.__gap (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#443) is not in mixedCase
Function HasContractURI.__HasContractURI_init_unchained(string) (contracts/HasContractURI.sol#16-19) is not in mixedCase
Parameter HasContractURI.__HasContractURI_init_unchained(string)._contractURI (contracts/HasContractURI.sol#16) is not in mixedCase
Variable HasContractURI.__gap (contracts/HasContractURI.sol#29) is not in mixedCase
Parameter LibURI.checkPrefix(string,string)._tokenURI (contracts/LibURI.sol#7) is not in mixedCase
Function Mint721Validator.__Mint721Validator_init_unchained() (contracts/Mint721Validator.sol#9-11) is not in mixedCase
Variable Mint721Validator.__gap (contracts/Mint721Validator.sol#16) is not in mixedCase
Parameter ERC721RaribleRevealWaveFactoryC2.createToken(string,string,string,string,uint256)._name (contracts/create-2/ERC721RaribleRevealWaveFactoryC2.sol#31) is not in mixedCase
Parameter ERC721RaribleRevealWaveFactoryC2.createToken(string,string,string,string,uint256)._symbol (contracts/create-2/ERC721RaribleRevealWaveFactoryC2.sol#31) is not in mixedCase
Parameter ERC721RaribleRevealWaveFactoryC2.createToken(string,string,string,string,uint256)._baseURI (contracts/create-2/ERC721RaribleRevealWaveFactoryC2.sol#31) is not in mixedCase
Parameter ERC721RaribleRevealWaveFactoryC2.createToken(string,string,string,string,address[],uint256)._symbol (contracts/create-2/ERC721RaribleRevealWaveFactoryC2.sol#39) is not in mixedCase
Parameter ERC721RaribleRevealWaveFactoryC2.createToken(string,string,string,string,address[],uint256)._symbol (contracts/create-2/ERC721RaribleRevealWaveFactoryC2.sol#39) is not in mixedCase
Parameter ERC721RaribleRevealWaveFactoryC2.getCreationBytecode(bytes)._data (contracts/create-2/ERC721RaribleRevealWaveFactoryC2.sol#59) is not in mixedCase
Parameter ERC721RaribleRevealWaveFactoryC2.getAddress(string,string,string,string,uint256)._name (contracts/create-2/ERC721RaribleRevealWaveFactoryC2.sol#64) is not in mixedCase
Parameter ERC721RaribleRevealWaveFactoryC2.getAddress(string,string,string,string,uint256)._symbol (contracts/create-2/ERC721RaribleRevealWaveFactoryC2.sol#64) is not in mixedCase
Parameter ERC721RaribleRevealWaveFactoryC2.getAddress(string,string,string,string,uint256)._salt (contracts/create-2/ERC721RaribleRevealWaveFactoryC2.sol#64) is not in mixedCase
Parameter ERC721RaribleRevealWaveFactoryC2.getData(string,string,string,string,string)._name (contracts/create-2/ERC721RaribleRevealWaveFactoryC2.sol#78) is not in mixedCase
Parameter ERC721RaribleRevealWaveFactoryC2.getData(string,string,string,string,string)._symbol (contracts/create-2/ERC721RaribleRevealWaveFactoryC2.sol#78) is not in mixedCase
Parameter ERC721RaribleRevealWaveFactoryC2.getAddress(string,string,string,string,string,address[],uint256)._name (contracts/create-2/ERC721RaribleRevealWaveFactoryC2.sol#83) is not in mixedCase
Parameter ERC721RaribleRevealWaveFactoryC2.getAddress(string,string,string,string,string,address[],uint256)._symbol (contracts/create-2/ERC721RaribleRevealWaveFactoryC2.sol#83) is not in mixedCase
Parameter ERC721RaribleRevealWaveFactoryC2.getAddress(string,string,string,string,string,address[],uint256)._salt (contracts/create-2/ERC721RaribleRevealWaveFactoryC2.sol#83) is not in mixedCase
Parameter ERC721RaribleRevealWaveFactoryC2.getData(string,string,string,string,string,address[])._name (contracts/create-2/ERC721RaribleRevealWaveFactoryC2.sol#97) is not in mixedCase
Parameter ERC721RaribleRevealWaveFactoryC2.getData(string,string,string,string,string,address[])._symbol (contracts/create-2/ERC721RaribleRevealWaveFactoryC2.sol#97) is not in mixedCase
Variable ERC1271Validator.__gap (contracts/erc-1271/ERC1271Validator.sol#94) is not in mixedCase
Variable ERC721BaseMinimal.__gap (contracts/erc-721-minimal/ERC721BaseMinimal.sol#38) is not in mixedCase
Function ERC721BurnableUpgradeableMinimal.__ERC721Burnable_init() (contracts/erc-721-minimal/ERC721BurnableUpgradeableMinimal.sol#14-18) is not in mixedCase
Function ERC721BurnableUpgradeableMinimal.__ERC721Burnable_init_unchained() (contracts/erc-721-minimal/ERC721BurnableUpgradeableMinimal.sol#20-21) is not in mixedCase
Variable ERC721BurnableUpgradeableMinimal.__gap (contracts/erc-721-minimal/ERC721BurnableUpgradeableMinimal.sol#41) is not in mixedCase
Variable ERC721DefaultApprovalMinimal.__gap (contracts/erc-721-minimal/ERC721DefaultApprovalMinimal.sol#24) is not in mixedCase
Function ERC721LazyMinimal.__ERC721Lazy_init_unchained() (contracts/erc-721-minimal/ERC721LazyMinimal.sol#25-27) is not in mixedCase
Parameter ERC721LazyMinimal.updateAccount(uint256,address,address)._id (contracts/erc-721-minimal/ERC721LazyMinimal.sol#95) is not in mixedCase
Parameter ERC721LazyMinimal.updateAccount(uint256,address,address)._from (contracts/erc-721-minimal/ERC721LazyMinimal.sol#95) is not in mixedCase
Parameter ERC721LazyMinimal.updateAccount(uint256,address,address)._to (contracts/erc-721-minimal/ERC721LazyMinimal.sol#95) is not in mixedCase
Parameter ERC721LazyMinimal.getCreators(uint256)._id (contracts/erc-721-minimal/ERC721LazyMinimal.sol#100) is not in mixedCase
Variable ERC721LazyMinimal.__gap (contracts/erc-721-minimal/ERC721LazyMinimal.sol#112) is not in mixedCase
Function ERC721RaribleMinimalRevealWave.__ERC721RaribleUser_init(string,string,string,string,address[],address,address) (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#29-30) is not in mixedCase
Parameter ERC721RaribleMinimalRevealWave.__ERC721RaribleUser_init(string,string,string,string,address[],address,address)._name (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#29) is not in mixedCase
Parameter ERC721RaribleMinimalRevealWave.__ERC721RaribleUser_init(string,string,string,string,address[],address,address)._symbol (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#29) is not in mixedCase
Function ERC721RaribleMinimalRevealWave.__ERC721Rarible_init(string,string,string,string,address,address) (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#40-45) is not in mixedCase
Parameter ERC721RaribleMinimalRevealWave.__ERC721Rarible_init(string,string,string,string,address,address)._name (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#40) is not in mixedCase
Parameter ERC721RaribleMinimalRevealWave.__ERC721Rarible_init(string,string,string,string,address,address)._symbol (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#40) is not in mixedCase
Function ERC721RaribleMinimalRevealWave.__ERC721Rarible_init_unchained(string,string,string,string,address,address) (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#47-62) is not in mixedCase
Parameter ERC721RaribleMinimalRevealWave.__ERC721Rarible_init_unchained(string,string,string,string,address,address)._name (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#47) is not in mixedCase
Parameter ERC721RaribleMinimalRevealWave.__ERC721Rarible_init_unchained(string,string,string,string,address,address)._symbol (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#47) is not in mixedCase
Parameter ERC721RaribleMinimalRevealWave.getRevealWaveIdByTokenId(uint256)._tokenId (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#64) is not in mixedCase
Parameter ERC721RaribleMinimalRevealWave.setRevealWave(uint256,string,string,bool,uint256)._waveId (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#71) is not in mixedCase
Parameter ERC721RaribleMinimalRevealWave.setRevealWave(uint256,string,string,bool,uint256)._name (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#72) is not in mixedCase
Parameter ERC721RaribleMinimalRevealWave.setRevealWave(uint256,string,string,bool,uint256)._hiddenBaseURI (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#73) is not in mixedCase
Parameter ERC721RaribleMinimalRevealWave.setRevealWave(uint256,string,string,bool,uint256)._addTokenURIToHiddenBaseURI (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#74) is not in mixedCase
Parameter ERC721RaribleMinimalRevealWave.setRevealWave(uint256,string,string,bool,uint256)._revealDate (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#75) is not in mixedCase
Parameter ERC721RaribleMinimalRevealWave.changeHiddenBaseURI(uint256,string)._waveId (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#86) is not in mixedCase
Parameter ERC721RaribleMinimalRevealWave.changeHiddenBaseURI(uint256,string)._baseURI (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#87) is not in mixedCase
Parameter ERC721RaribleMinimalRevealWave.changeAddTokenURIToHiddenBaseURI(uint256,bool)._waveId (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#95) is not in mixedCase
Parameter ERC721RaribleMinimalRevealWave.changeAddTokenURIToHiddenBaseURI(uint256,bool)._addTokenURIToHiddenBaseURI (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#96) is not in mixedCase
Parameter ERC721RaribleMinimalRevealWave.changeRevealBaseURI(uint256,string)._waveId (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#103) is not in mixedCase
Parameter ERC721RaribleMinimalRevealWave.changeRevealBaseURI(uint256,string)._revealBaseURI (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#104) is not in mixedCase
Parameter ERC721RaribleMinimalRevealWave.changeRevealDate(uint256,uint256)._waveId (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#114) is not in mixedCase
Parameter ERC721RaribleMinimalRevealWave.changeRevealDate(uint256,uint256)._revealDate (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#115) is not in mixedCase
Parameter ERC721RaribleMinimalRevealWave.resetIsRevealedAndRevealURI(uint256)._waveId (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#123) is not in mixedCase
Parameter ERC721RaribleMinimalRevealWave.changeName(uint256,string)._waveId (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#131) is not in mixedCase
Parameter ERC721RaribleMinimalRevealWave.changeName(uint256,string)._name (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#132) is not in mixedCase
Parameter ERC721RaribleMinimalRevealWave.assignRevealWaveIdToTokenId(uint256,uint256)._tokenId (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#140) is not in mixedCase
Parameter ERC721RaribleMinimalRevealWave.assignRevealWaveIdToTokenId(uint256,uint256)._waveId (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#140) is not in mixedCase
Parameter ERC721RaribleMinimalRevealWave.tokenURI(uint256)._tokenId (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#152) is not in mixedCase
Parameter ERC721RaribleMinimalRevealWave.mintAndTransferReveal(LibERC721LazyMint.Mint721Data,address,uint256)._waveId (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#202) is not in mixedCase
Variable ERC721RaribleMinimalRevealWave.__gap (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#211) is not in mixedCase
Variable ERC721URI.__gap (contracts/erc-721-minimal/ERC721URI.sol#92) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#94)" inEIP712Upgradeable (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#25-121)
Redundant expression "this (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#28)" inContextUpgradeable (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#16-32)
Redundant expression "this (node_modules/@openzeppelin/contracts/utils/Context.sol#21)" inContext (node_modules/@openzeppelin/contracts/utils/Context.sol#15-24)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

ERC721RaribleRevealWaveFactoryC2.getCreationBytecode(bytes) (contracts/create-2/ERC721RaribleRevealWaveFactoryC2.sol#59-61) uses literals with too many digits:
        - abi.encodePacked(type()(BeaconProxy).creationCode,abi.encode(beacon,_data)) (contracts/create-2/ERC721RaribleRevealWaveFactoryC2.sol#60)
ERC1271.slitherConstructorConstantVariables() (contracts/erc-1271/ERC1271.sol#5-26) uses literals with too many digits:
        - ERC1271_RETURN_INVALID_SIGNATURE = 0x00000000 (contracts/erc-1271/ERC1271.sol#9)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits

ERC1271 (contracts/erc-1271/ERC1271.sol#5-26) does not implement functions:
        - ERC1271.isValidSignature(bytes32,bytes) (contracts/erc-1271/ERC1271.sol#20)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unimplemented-functions

LibERC721LazyMint._INTERFACE_ID_MINT_AND_TRANSFER (node_modules/@rarible/lazy-mint/contracts/erc-721/LibERC721LazyMint.sol#9) is never used in LibERC721LazyMint (node_modules/@rarible/lazy-mint/contracts/erc-721/LibERC721LazyMint.sol#7-3
9)
ERC721RaribleMinimalRevealWave.__gap (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#211) is never used in ERC721RaribleMinimalRevealWave (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#8-213)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

renounceOwnership() should be declared external:
        - OwnableUpgradeable.renounceOwnership() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#60-63)
transferOwnership(address) should be declared external:
        - OwnableUpgradeable.transferOwnership(address) (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#69-73)
renounceOwnership() should be declared external:
        - Ownable.renounceOwnership() (node_modules/@openzeppelin/contracts/access/Ownable.sol#54-57)
transferOwnership(address) should be declared external:
        - Ownable.transferOwnership(address) (node_modules/@openzeppelin/contracts/access/Ownable.sol#63-67)
balanceOf(address) should be declared external:
        - ERC721UpgradeableMinimal.balanceOf(address) (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#89-92)
name() should be declared external:
        - ERC721UpgradeableMinimal.name() (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#106-108)
symbol() should be declared external:
        - ERC721UpgradeableMinimal.symbol() (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#113-115)
approve(address,uint256) should be declared external:
        - ERC721UpgradeableMinimal.approve(address,uint256) (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#126-136)
transferFrom(address,address,uint256) should be declared external:

        - ERC721UpgradeableMinimal.transferFrom(address,address,uint256) (node_modules/@rarible/tokens-minimal/contracts/erc-721/ERC721UpgradeableMinimal.sol#167-176)
getAddress(string,string,string,string,uint256) should be declared external:
        - ERC721RaribleRevealWaveFactoryC2.getAddress(string,string,string,string,uint256) (contracts/create-2/ERC721RaribleRevealWaveFactoryC2.sol#64-76)
getAddress(string,string,string,string,address[],uint256) should be declared external:
        - ERC721RaribleRevealWaveFactoryC2.getAddress(string,string,string,string,address[],uint256) (contracts/create-2/ERC721RaribleRevealWaveFactoryC2.sol#83-95)
isValidSignature(bytes32,bytes) should be declared external:
        - ERC1271.isValidSignature(bytes32,bytes) (contracts/erc-1271/ERC1271.sol#20)
burn(uint256) should be declared external:
        - ERC721BurnableUpgradeableMinimal.burn(uint256) (contracts/erc-721-minimal/ERC721BurnableUpgradeableMinimal.sol#29-39)
setRevealWave(uint256,string,string,bool,uint256) should be declared external:
        - ERC721RaribleMinimalRevealWave.setRevealWave(uint256,string,string,bool,uint256) (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#70-83)
changeHiddenBaseURI(uint256,string) should be declared external:
        - ERC721RaribleMinimalRevealWave.changeHiddenBaseURI(uint256,string) (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#85-92)
changeAddTokenURIToHiddenBaseURI(uint256,bool) should be declared external:
        - ERC721RaribleMinimalRevealWave.changeAddTokenURIToHiddenBaseURI(uint256,bool) (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#94-100)
changeRevealBaseURI(uint256,string) should be declared external:
        - ERC721RaribleMinimalRevealWave.changeRevealBaseURI(uint256,string) (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#102-111)
changeRevealDate(uint256,uint256) should be declared external:
        - ERC721RaribleMinimalRevealWave.changeRevealDate(uint256,uint256) (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#113-119)
resetIsRevealedAndRevealURI(uint256) should be declared external:
        - ERC721RaribleMinimalRevealWave.resetIsRevealedAndRevealURI(uint256) (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#121-127)
changeName(uint256,string) should be declared external:
        - ERC721RaribleMinimalRevealWave.changeName(uint256,string) (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#130-137)
mintAndTransferReveal(LibERC721LazyMint.Mint721Data,address,uint256) should be declared external:
        - ERC721RaribleMinimalRevealWave.mintAndTransferReveal(LibERC721LazyMint.Mint721Data,address,uint256) (contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol#199-209)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

ERC721WoonklyNFTMinimalBeacon.sol

```
Address.isContract(address) (node_modules/@openzeppelin/contracts/utils/Address.sol#26-35) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/utils/Address.sol#33)
Address._verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#171-188) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/utils/Address.sol#180-183)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Different versions of Solidity is used:
        - Version used: ['>=0.6.0<0.8.0', '>=0.6.2<0.8.0', '^0.7.0']
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/access/Ownable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/proxy/IBeacon.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/proxy/UpgradeableBeacon.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts/utils/Address.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#3)
        - ^0.7.0 (contracts/beacons/ERC721BarrbleMinimalBeacon.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

Address._verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#171-188) is never used and should be removed
Address.functionCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#79-81) is never used and should be removed
Address.functionCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#89-91) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#104-106) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#114-121) is never used and should be removed
Address.functionDelegateCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#153-155) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#163-169) is never used and should be removed
Address.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#129-131) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#139-145) is never used and should be removed
Address.sendValue(address,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#53-59) is never used and should be removed
Context._msgData() (node_modules/@openzeppelin/contracts/utils/Context.sol#20-23) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/access/Ownable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/proxy/IBeacon.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/proxy/UpgradeableBeacon.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts/utils/Address.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#3) is too complex
Pragma version^0.7.0 (contracts/beacons/ERC721BarrbleMinimalBeacon.sol#3) allows old versions
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

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

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

Variable UpgradeableBeacon._implementation (node_modules/@openzeppelin/contracts/proxy/UpgradeableBeacon.sol#16) is too similar to UpgradeableBeacon.constructor(address).implementation_ (node_modules/@openzeppelin/contracts/proxy/Upgrade
ableBeacon.sol#27)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-are-too-similar

renounceOwnership() should be declared external:
        - Ownable.renounceOwnership() (node_modules/@openzeppelin/contracts/access/Ownable.sol#54-57)
transferOwnership(address) should be declared external:
        - Ownable.transferOwnership(address) (node_modules/@openzeppelin/contracts/access/Ownable.sol#63-67)
implementation() should be declared external:
        - UpgradeableBeacon.implementation() (node_modules/@openzeppelin/contracts/proxy/UpgradeableBeacon.sol#34-36)
upgradeTo(address) should be declared external:
        - UpgradeableBeacon.upgradeTo(address) (node_modules/@openzeppelin/contracts/proxy/UpgradeableBeacon.sol#40-51)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

ERC721WoonklyNFTBeacon.sol

```
Address.isContract(address) (node_modules/@openzeppelin/contracts/utils/Address.sol#26-35) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/utils/Address.sol#33)
Address._verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#171-188) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/utils/Address.sol#180-183)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Different versions of Solidity is used:
        - Version used: ['>=0.6.0<0.8.0', '>=0.6.2<0.8.0', '^0.7.0']
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/access/Ownable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/proxy/IBeacon.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/proxy/UpgradeableBeacon.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts/utils/Address.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#3)
        - ^0.7.0 (contracts/beacons/ERC721BarrbleBeacon.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

Address._verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#171-188) is never used and should be removed
Address.functionCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#79-81) is never used and should be removed
Address.functionCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#89-91) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#104-106) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#114-121) is never used and should be removed
Address.functionDelegateCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#153-155) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#163-169) is never used and should be removed
Address.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#129-131) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#139-145) is never used and should be removed
Address.sendValue(address,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#53-59) is never used and should be removed
Context._msgData() (node_modules/@openzeppelin/contracts/utils/Context.sol#20-23) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/access/Ownable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/proxy/IBeacon.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/proxy/UpgradeableBeacon.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts/utils/Address.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#3) is too complex
Pragma version^0.7.0 (contracts/beacons/ERC721BarrbleBeacon.sol#3) allows old versions
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

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

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

Variable UpgradeableBeacon._implementation (node_modules/@openzeppelin/contracts/proxy/UpgradeableBeacon.sol#16) is too similar to UpgradeableBeacon.constructor(address).implementation_ (node_modules/@openzeppelin/contracts/proxy/Upgrade
ableBeacon.sol#27)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-are-too-similar

renounceOwnership() should be declared external:
        - Ownable.renounceOwnership() (node_modules/@openzeppelin/contracts/access/Ownable.sol#54-57)
transferOwnership(address) should be declared external:
        - Ownable.transferOwnership(address) (node_modules/@openzeppelin/contracts/access/Ownable.sol#63-67)
implementation() should be declared external:
        - UpgradeableBeacon.implementation() (node_modules/@openzeppelin/contracts/proxy/UpgradeableBeacon.sol#34-36)
upgradeTo(address) should be declared external:
        - UpgradeableBeacon.upgradeTo(address) (node_modules/@openzeppelin/contracts/proxy/UpgradeableBeacon.sol#40-51)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

ERC1155WoonklyNFT.sol

```
OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#74) shadows:
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
ERC1155ReceiverUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/ERC1155ReceiverUpgradeable.sol#24) shadows:
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
HasContractURI.__gap (contracts/HasContractURI.sol#28) shadows:
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
ERC1155Base.__gap (contracts/erc-1155/ERC1155Base.sol#82) shadows:
        - HasContractURI.__gap (contracts/HasContractURI.sol#28)
        - ERC1155Lazy.__gap (contracts/erc-1155/ERC1155Lazy.sol#148)
        - Mint1155Validator.__gap (contracts/erc-1155/Mint1155Validator.sol#16)
        - ERC1271Validator.__gap (contracts/erc-1271/ERC1271Validator.sol#34)
        - EIP712Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#120)
        - ERC1155BaseURI.__gap (contracts/erc-1155/ERC1155BaseURI.sol#69)
        - ERC1155BurnableUpgradeable.__gap (contracts/erc-1155/ERC1155BurnableUpgradeable.sol#40)
```

AUTOMATED TESTING

```
        - ERC1155DefaultApproval.__gap (contracts/erc-1155/ERC1155DefaultApproval.sol#20)
        - ERC1155Upgradeable.__gap (contracts/erc-1155/ERC1155Upgradeable.sol#421)
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#74)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
ERC1155BaseURI.__gap (contracts/erc-1155/ERC1155BaseURI.sol#69) shadows:
        - ERC1155Upgradeable.__gap (contracts/erc-1155/ERC1155Upgradeable.sol#421)
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
ERC1155BurnableUpgradeable.__gap (contracts/erc-1155/ERC1155BurnableUpgradeable.sol#40) shadows:
        - ERC1155Upgradeable.__gap (contracts/erc-1155/ERC1155Upgradeable.sol#421)
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
ERC1155DefaultApproval.__gap (contracts/erc-1155/ERC1155DefaultApproval.sol#20) shadows:
        - ERC1155Upgradeable.__gap (contracts/erc-1155/ERC1155Upgradeable.sol#421)
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
ERC1155Lazy.__gap (contracts/erc-1155/ERC1155Lazy.sol#148) shadows:
        - Mint1155Validator.__gap (contracts/erc-1155/Mint1155Validator.sol#16)
        - ERC1271Validator.__gap (contracts/erc-1271/ERC1271Validator.sol#34)
        - EIP712Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#120)
        - ERC1155BaseURI.__gap (contracts/erc-1155/ERC1155BaseURI.sol#69)
        - ERC1155Upgradeable.__gap (contracts/erc-1155/ERC1155Upgradeable.sol#421)
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
ERC1155Lazy._INTERFACE_ID_ERC165 (contracts/erc-1155/ERC1155Lazy.sol#16) shadows:
        - ERC165Upgradeable._INTERFACE_ID_ERC165 (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#18)
ERC1155Lazy._INTERFACE_ID_ERC1155 (contracts/erc-1155/ERC1155Lazy.sol#17) shadows:
        - ERC1155Upgradeable._INTERFACE_ID_ERC1155 (contracts/erc-1155/ERC1155Upgradeable.sol#46)
ERC1155Lazy._INTERFACE_ID_ERC1155_METADATA_URI (contracts/erc-1155/ERC1155Lazy.sol#18) shadows:
        - ERC1155Upgradeable._INTERFACE_ID_ERC1155_METADATA_URI (contracts/erc-1155/ERC1155Upgradeable.sol#51)
ERC1155Razible.__gap (contracts/erc-1155/ERC1155Razible.sol#57) shadows:
        - ERC1155Base.__gap (contracts/erc-1155/ERC1155Base.sol#82)
        - HasContractURI.__gap (contracts/HasContractURI.sol#29)
        - ERC1155Lazy.__gap (contracts/erc-1155/ERC1155Lazy.sol#148)
        - Mint1155Validator.__gap (contracts/erc-1155/Mint1155Validator.sol#16)
        - ERC1271Validator.__gap (contracts/erc-1271/ERC1271Validator.sol#34)
        - EIP712Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#120)
        - ERC1155BaseURI.__gap (contracts/erc-1155/ERC1155BaseURI.sol#69)
        - ERC1155BurnableUpgradeable.__gap (contracts/erc-1155/ERC1155BurnableUpgradeable.sol#40)
        - ERC1155DefaultApproval.__gap (contracts/erc-1155/ERC1155DefaultApproval.sol#20)
        - ERC1155Upgradeable.__gap (contracts/erc-1155/ERC1155Upgradeable.sol#421)
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#74)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
ERC1155Upgradeable.__gap (contracts/erc-1155/ERC1155Upgradeable.sol#421) shadows:
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
Mint1155Validator.__gap (contracts/erc-1155/Mint1155Validator.sol#16) shadows:
        - ERC1271Validator.__gap (contracts/erc-1271/ERC1271Validator.sol#34)
        - EIP712Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#120)
ERC1271Validator.__gap (contracts/erc-1271/ERC1271Validator.sol#34) shadows:
        - EIP712Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#120)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variable-shadowing

Reentrancy in ERC1155Lazy.transferFromOrMint(LibERC1155LazyMint.Mint1155Data,address,address,uint256) (contracts/erc-1155/ERC1155Lazy.sol#36-55):
        External calls:
        - safeTransferFrom(from,to,data.tokenId,transfer,) (contracts/erc-1155/ERC1155Lazy.sol#49)
                - IERC1155ReceiverUpgradeable(to).onERC1155Received(operator,from,id,amount,data) (contracts/erc-1155/ERC1155Upgradeable.sol#380-388)
        - mintAndTransfer(data,to,left) (contracts/erc-1155/ERC1155Lazy.sol#53)
                - IERC1155ReceiverUpgradeable(to).onERC1155Received(operator,from,id,amount,data) (contracts/erc-1155/ERC1155Upgradeable.sol#380-388)
        State variables written after the call(s):
        - mintAndTransfer(data,to,left) (contracts/erc-1155/ERC1155Lazy.sol#53)
                - _balances[id][account] = _balances[id][account].add(amount) (contracts/erc-1155/ERC1155Lazy.sol#103)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-1

ERC1155Upgradeable._doSafeTransferAcceptanceCheck(address,address,address,uint256,uint256,bytes).response (contracts/erc-1155/ERC1155Upgradeable.sol#380) is a local variable never initialized
ERC1271Validator.validate1271(address,bytes32,bytes).signerFromSig (contracts/erc-1271/ERC1271Validator.sol#19) is a local variable never initialized
ERC1155Upgradeable._doSafeBatchTransferAcceptanceCheck(address,address,address,uint256[],uint256[],bytes).reason (contracts/erc-1155/ERC1155Upgradeable.sol#407) is a local variable never initialized
ERC1155Upgradeable._doSafeTransferAcceptanceCheck(address,address,address,uint256,uint256,bytes).reason (contracts/erc-1155/ERC1155Upgradeable.sol#384) is a local variable never initialized
ERC1155Upgradeable._doSafeBatchTransferAcceptanceCheck(address,address,address,uint256[],uint256[],bytes).response (contracts/erc-1155/ERC1155Upgradeable.sol#403) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

ERC1155Upgradeable._doSafeTransferAcceptanceCheck(address,address,address,uint256,uint256,bytes) (contracts/erc-1155/ERC1155Upgradeable.sol#369-390) ignores return value by IERC1155ReceiverUpgradeable(to).onERC1155Received(operator,from,
id,amount,data) (contracts/erc-1155/ERC1155Upgradeable.sol#380-388)
ERC1155Upgradeable._doSafeBatchTransferAcceptanceCheck(address,address,address,uint256[],uint256[],bytes) (contracts/erc-1155/ERC1155Upgradeable.sol#392-413) ignores return value by IERC1155ReceiverUpgradeable(to).onERC1155BatchReceived(
operator,from,ids,amounts,data) (contracts/erc-1155/ERC1155Upgradeable.sol#403-411)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

ERC1155Base.isApprovedForAll(address,address)._owner (contracts/erc-1155/ERC1155Base.sol#19) shadows:
        - OwnableUpgradeable._owner (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#20) (state variable)
ERC1155Base._burnLazy(uint256,uint256).supply (contracts/erc-1155/ERC1155Base.sol#56) shadows:
        - ERC1155Lazy.supply (contracts/erc-1155/ERC1155Lazy.sol#21) (state variable)
ERC1155BaseURI._tokenURI(uint256)._tokenURI (contracts/erc-1155/ERC1155BaseURI.sol#33) shadows:
        - ERC1155BaseURI._tokenURI(uint256) (contracts/erc-1155/ERC1155BaseURI.sol#32-46) (function)
ERC1155BaseURI._setTokenURI(uint256,string)._uri (contracts/erc-1155/ERC1155BaseURI.sol#55) shadows:
        - ERC1155Upgradeable._uri (contracts/erc-1155/ERC1155Upgradeable.sol#33) (state variable)
ERC1155Razible.__ERC1155RazibleUser_init(string,string,string,string,address[],address,address).baseURI (contracts/erc-1155/ERC1155Razible.sol#15) shadows:
        - ERC1155BaseURI.baseURI() (contracts/erc-1155/ERC1155BaseURI.sol#24-26) (function)
ERC1155Razible.__ERC1155RazibleUser_init(string,string,string,string,address[],address,address).contractURI (contracts/erc-1155/ERC1155Razible.sol#15) shadows:
        - HasContractURI.contractURI (contracts/HasContractURI.sol#9) (state variable)
ERC1155Razible.__ERC1155Razible_init(string,string,string,string,address,address).baseURI (contracts/erc-1155/ERC1155Razible.sol#25) shadows:
        - ERC1155BaseURI.baseURI() (contracts/erc-1155/ERC1155BaseURI.sol#24-26) (function)
ERC1155Razible.__ERC1155Razible_init(string,string,string,string,address,address).contractURI (contracts/erc-1155/ERC1155Razible.sol#25) shadows:
        - HasContractURI.contractURI (contracts/HasContractURI.sol#9) (state variable)
ERC1155Razible.__ERC1155Razible_init_unchained(string,string,string,string,address,address).baseURI (contracts/erc-1155/ERC1155Razible.sol#32) shadows:
        - ERC1155BaseURI.baseURI() (contracts/erc-1155/ERC1155BaseURI.sol#24-26) (function)
ERC1155Razible.__ERC1155Razible_init_unchained(string,string,string,string,address,address).contractURI (contracts/erc-1155/ERC1155Razible.sol#32) shadows:
        - HasContractURI.contractURI (contracts/HasContractURI.sol#9) (state variable)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

Variable 'ERC1155Upgradeable._doSafeTransferAcceptanceCheck(address,address,address,uint256,uint256,bytes).response (contracts/erc-1155/ERC1155Upgradeable.sol#380)' in ERC1155Upgradeable._doSafeTransferAcceptanceCheck(address,address,add
ress,uint256,uint256,bytes) (contracts/erc-1155/ERC1155Upgradeable.sol#369-390) potentially used before declaration: response != IERC1155ReceiverUpgradeable(to).onERC1155Received.selector (contracts/erc-1155/ERC1155Upgradeable.sol#381)
Variable 'ERC1155Upgradeable._doSafeTransferAcceptanceCheck(address,address,address,uint256,uint256,bytes).reason (contracts/erc-1155/ERC1155Upgradeable.sol#384)' in ERC1155Upgradeable._doSafeTransferAcceptanceCheck(address,address,addre
ss,uint256,uint256,bytes) (contracts/erc-1155/ERC1155Upgradeable.sol#369-390) potentially used before declaration: revert(string)(reason) (contracts/erc-1155/ERC1155Upgradeable.sol#385)
Variable 'ERC1155Upgradeable._doSafeBatchTransferAcceptanceCheck(address,address,address,uint256[],uint256[],bytes).response (contracts/erc-1155/ERC1155Upgradeable.sol#403)' in ERC1155Upgradeable._doSafeBatchTransferAcceptanceCheck(addre
ss,address,address,uint256[],uint256[],bytes) (contracts/erc-1155/ERC1155Upgradeable.sol#392-413) potentially used before declaration: response != IERC1155ReceiverUpgradeable(to).onERC1155BatchReceived.selector (contracts/erc-1155/ERC115
5Upgradeable.sol#404)
Variable 'ERC1155Upgradeable._doSafeBatchTransferAcceptanceCheck(address,address,address,uint256[],uint256[],bytes).reason (contracts/erc-1155/ERC1155Upgradeable.sol#407)' in ERC1155Upgradeable._doSafeBatchTransferAcceptanceCheck(address
,address,address,uint256[],uint256[],bytes) (contracts/erc-1155/ERC1155Upgradeable.sol#392-413) potentially used before declaration: revert(string)(reason) (contracts/erc-1155/ERC1155Upgradeable.sol#408)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables

Reentrancy in ERC1155Lazy.mintAndTransfer(LibERC1155LazyMint.Mint1155Data,address,uint256) (contracts/erc-1155/ERC1155Lazy.sol#57-90):
        External calls:
        - _mint(to,data.tokenId,_amount,) (contracts/erc-1155/ERC1155Lazy.sol#83)
                - IERC1155ReceiverUpgradeable(to).onERC1155Received(operator,from,id,amount,data) (contracts/erc-1155/ERC1155Upgradeable.sol#380-388)
        Event emitted after the call(s):
        - TransferSingle(sender,address(0),minter,data.tokenId,_amount) (contracts/erc-1155/ERC1155Lazy.sol#85)
        - TransferSingle(sender,minter,to,data.tokenId,_amount) (contracts/erc-1155/ERC1155Lazy.sol#86)
        - TransferSingle(sender,address(0),to,data.tokenId,_amount) (contracts/erc-1155/ERC1155Lazy.sol#88)
Reentrancy in ERC1155Lazy.transferFromOrMint(LibERC1155LazyMint.Mint1155Data,address,address,uint256) (contracts/erc-1155/ERC1155Lazy.sol#36-55):
        External calls:
        - safeTransferFrom(from,to,data.tokenId,transfer,) (contracts/erc-1155/ERC1155Lazy.sol#49)
                - IERC1155ReceiverUpgradeable(to).onERC1155Received(operator,from,id,amount,data) (contracts/erc-1155/ERC1155Upgradeable.sol#380-388)
        - mintAndTransfer(data,to,left) (contracts/erc-1155/ERC1155Lazy.sol#53)
                - IERC1155ReceiverUpgradeable(to).onERC1155Received(operator,from,id,amount,data) (contracts/erc-1155/ERC1155Upgradeable.sol#380-388)

        Event emitted after the call(s):
        - Creators(tokenId,_creators) (contracts/erc-1155/ERC1155Lazy.sol#124)
                - mintAndTransfer(data,to,left) (contracts/erc-1155/ERC1155Lazy.sol#53)
        - RoyaltiesSet(id,_royalties) (node_modules/@rarible/royalties/contracts/impl/RoyaltiesV2Impl.sol#16)
                - mintAndTransfer(data,to,left) (contracts/erc-1155/ERC1155Lazy.sol#53)
        - Supply(tokenId,_supply) (contracts/erc-1155/ERC1155Lazy.sol#111)
                - mintAndTransfer(data,to,left) (contracts/erc-1155/ERC1155Lazy.sol#53)
        - TransferSingle(sender,address(0),minter,data.tokenId,_amount) (contracts/erc-1155/ERC1155Lazy.sol#85)
                - mintAndTransfer(data,to,left) (contracts/erc-1155/ERC1155Lazy.sol#53)
        - TransferSingle(sender,minter,to,data.tokenId,_amount) (contracts/erc-1155/ERC1155Lazy.sol#86)
                - mintAndTransfer(data,to,left) (contracts/erc-1155/ERC1155Lazy.sol#53)
        - TransferSingle(sender,address(0),to,data.tokenId,_amount) (contracts/erc-1155/ERC1155Lazy.sol#88)
                - mintAndTransfer(data,to,left) (contracts/erc-1155/ERC1155Lazy.sol#53)
        - URI(_tokenURI(tokenId),tokenId) (contracts/erc-1155/ERC1155BaseURI.sol#57)
                - mintAndTransfer(data,to,left) (contracts/erc-1155/ERC1155Lazy.sol#53)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

EIP712Upgradeable._getChainId() (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#93-98) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#96-98)
AddressUpgradeable.isContract(address) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#26-35) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#33)
AddressUpgradeable._verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#147-164) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#156-159)
LibSignature.recover(bytes32,bytes) (node_modules/@rarible/libraries/contracts/LibSignature.sol#20-45) uses assembly
        - INLINE ASM (node_modules/@rarible/libraries/contracts/LibSignature.sol#38-42)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Different versions of Solidity is used:
        - Version used: ['0.7.6', '>=0.4.24<0.8.0', '>=0.6.0<0.8.0', '>=0.6.2<0.8.0', '^0.7.0']
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/introspection/IERC165Upgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#3)
        - >=0.4.24<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/proxy/Initializable.sol#4)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/ERC1155ReceiverUpgradeable.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/IERC1155MetadataURIUpgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/IERC1155ReceiverUpgradeable.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/IERC1155Upgradeable.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/StringsUpgradeable.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@rarible/lazy-mint/contracts/erc-1155/IERC1155LazyMint.sol#3)
        - v2 (node_modules/@rarible/lazy-mint/contracts/erc-1155/IERC1155LazyMint.sol#4)
        - >=0.6.2<0.8.0 (node_modules/@rarible/lazy-mint/contracts/erc-1155/LibERC1155LazyMint.sol#3)
        - ^0.7.0 (node_modules/@rarible/libraries/contracts/LibSignature.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@rarible/royalties-upgradeable/contracts/RoyaltiesV2Upgradeable.sol#3)
        - v2 (node_modules/@rarible/royalties-upgradeable/contracts/RoyaltiesV2Upgradeable.sol#4)
        - >=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/LibPart.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/LibRoyaltiesV2.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/RoyaltiesV2.sol#3)
        - v2 (node_modules/@rarible/royalties/contracts/RoyaltiesV2.sol#4)
```

AUTOMATED TESTING

```
        - >=0.6.2<0.8.0 (node_modules/@zarible/royalties/contracts/impl/AbstractRoyalties.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@zarible/royalties/contracts/impl/RoyaltiesV2Impl.sol#3)
        - v2 (node_modules/@zarible/royalties/contracts/impl/RoyaltiesV2Impl.sol#4)
        - 0.7.6 (contracts/HasContractURI.sol#3)
        - 0.7.6 (contracts/LibURI.sol#3)
        - 0.7.6 (contracts/erc-1155/ERC1155Base.sol#3)
        - v2 (contracts/erc-1155/ERC1155Base.sol#4)
        - 0.7.6 (contracts/erc-1155/ERC1155BaseURI.sol#3)
        - >=0.6.0<0.8.0 (contracts/erc-1155/ERC1155BurnableUpgradeable.sol#3)
        - 0.7.6 (contracts/erc-1155/ERC1155DefaultApproval.sol#3)
        - 0.7.6 (contracts/erc-1155/ERC1155Lazy.sol#3)
        - v2 (contracts/erc-1155/ERC1155Lazy.sol#4)
        - 0.7.6 (contracts/erc-1155/ERC1155Rarible.sol#3)
        - v2 (contracts/erc-1155/ERC1155Rarible.sol#4)
        - >=0.6.0<0.8.0 (contracts/erc-1155/ERC1155Upgradeable.sol#3)
        - 0.7.6 (contracts/erc-1155/MintI155Validator.sol#3)
        - 0.7.6 (contracts/erc-1271/ERC1271.sol#3)
        - 0.7.6 (contracts/erc-1271/ERC1271Validator.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

AbstractRoyalties._onRoyaltiesSet(uint256,LibPart.Part[]) (node_modules/@zarible/royalties/contracts/impl/AbstractRoyalties.sol#31) is never used and should be removed
AddressUpgradeable._verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#147-164) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#79-81) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#89-91) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#104-106) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#114-121) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#129-131) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#139-145) is never used and should be removed
AddressUpgradeable.sendValue(address,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#53-59) is never used and should be removed
ContextUpgradeable.__Context_init() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#17-19) is never used and should be removed
ContextUpgradeable._msgData() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#27-30) is never used and should be removed
EIP712Upgradeable.__EIP712_init(string,string) (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#44-46) is never used and should be removed
ERC1155BurnableUpgradeable.__ERC1155Burnable_init() (contracts/erc-1155/ERC1155BurnableUpgradeable.sol#15-19) is never used and should be removed
ERC1155ReceiverUpgradeable.__ERC1155Receiver_init() (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/ERC1155ReceiverUpgradeable.sol#13-16) is never used and should be removed
ERC1155ReceiverUpgradeable.__ERC1155Receiver_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/ERC1155ReceiverUpgradeable.sol#18-23) is never used and should be removed
ERC1155Upgradeable.__ERC1155_init(string) (contracts/erc-1155/ERC1155Upgradeable.sol#56-60) is never used and should be removed
ERC1155Upgradeable._mint(address,uint256,uint256,bytes) (contracts/erc-1155/ERC1155Upgradeable.sol#250-261) is never used and should be removed
ERC1155Upgradeable._mintBatch(address,uint256[],uint256[],bytes) (contracts/erc-1155/ERC1155Upgradeable.sol#272-287) is never used and should be removed
ERC1271.returnIsValidSignatureMagicNumber(bool) (contracts/erc-1271/ERC1271.sol#22-24) is never used and should be removed
ERC165Upgradeable.__ERC165_init() (node_modules/@openzeppelin/contracts-upgradeable/utils/introspection/ERC165Upgradeable.sol#25-27) is never used and should be removed
HasContractURI._setContractURI(string) (contracts/HasContractURI.sol#25-27) is never used and should be removed
OwnableUpgradeable.__Ownable_init() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#27-30) is never used and should be removed
SafeMathUpgradeable.div(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#135-138) is never used and should be removed
SafeMathUpgradeable.div(uint256,uint256,string) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#190-193) is never used and should be removed
SafeMathUpgradeable.mod(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#152-155) is never used and should be removed
SafeMathUpgradeable.mod(uint256,uint256,string) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#210-213) is never used and should be removed
SafeMathUpgradeable.mul(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#116-121) is never used and should be removed
SafeMathUpgradeable.sub(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#101-104) is never used and should be removed
SafeMathUpgradeable.tryAdd(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#24-28) is never used and should be removed
SafeMathUpgradeable.tryDiv(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#60-63) is never used and should be removed
SafeMathUpgradeable.tryMod(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#70-73) is never used and should be removed
SafeMathUpgradeable.tryMul(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#45-53) is never used and should be removed
SafeMathUpgradeable.trySub(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#35-38) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/introspection/IERC165Upgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#3) is too complex
Pragma version>=0.4.24<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/proxy/Initializable.sol#4) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/ERC1155ReceiverUpgradeable.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/IERC1155MetadataURIUpgradeable.sol#3) is too complex

Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/IERC1155ReceiverUpgradeable.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/IERC1155Upgradeable.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/StringsUpgradeable.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@zarible/lazy-mint/contracts/erc-1155/IERC1155LazyMint.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@zarible/lazy-mint/contracts/erc-1155/LibERC1155LazyMint.sol#3) is too complex
Pragma version^0.7.0 (node_modules/@zarible/libraries/contracts/LibSignature.sol#3) allows old versions
Pragma version>=0.6.2<0.8.0 (node_modules/@zarible/royalties-upgradeable/contracts/RoyaltiesV2Upgradeable.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@zarible/royalties/contracts/LibPart.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@zarible/royalties/contracts/LibRoyaltiesV2.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@zarible/royalties/contracts/RoyaltiesV2.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@zarible/royalties/contracts/impl/AbstractRoyalties.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@zarible/royalties/contracts/impl/RoyaltiesV2Impl.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (contracts/erc-1155/ERC1155BurnableUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (contracts/erc-1155/ERC1155Upgradeable.sol#3) is too complex
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

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

Function OwnableUpgradeable.__Ownable_init() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#27-30) is not in mixedCase
Function OwnableUpgradeable.__Ownable_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#32-36) is not in mixedCase
Variable OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#76) is not in mixedCase
Function EIP712Upgradeable.__EIP712_init(string,string) (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#44-46) is not in mixedCase
Function EIP712Upgradeable.__EIP712_init_unchained(string,string) (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#48-53) is not in mixedCase
Function EIP712Upgradeable._EIP712NameHash() (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#107-109) is not in mixedCase
Function EIP712Upgradeable._EIP712VersionHash() (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#117-119) is not in mixedCase
Variable EIP712Upgradeable._HASHED_NAME (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#27) is not in mixedCase
Variable EIP712Upgradeable._HASHED_VERSION (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#28) is not in mixedCase
Variable EIP712Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#120) is not in mixedCase
Function ERC165Upgradeable.__ERC165_init() (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#25-27) is not in mixedCase
Function ERC165Upgradeable.__ERC165_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#29-33) is not in mixedCase
Variable ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59) is not in mixedCase
Function ERC1155ReceiverUpgradeable.__ERC1155Receiver_init() (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/ERC1155ReceiverUpgradeable.sol#13-16) is not in mixedCase
Function ERC1155ReceiverUpgradeable.__ERC1155Receiver_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/ERC1155ReceiverUpgradeable.sol#18-23) is not in mixedCase
Variable ERC1155ReceiverUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/ERC1155ReceiverUpgradeable.sol#24) is not in mixedCase
Function ContextUpgradeable.__Context_init() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#17-19) is not in mixedCase
Function ContextUpgradeable.__Context_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#21-22) is not in mixedCase
Variable ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31) is not in mixedCase
Function RoyaltiesV2Upgradeable.__RoyaltiesV2Upgradeable_init_unchained() (node_modules/@zarible/royalties-upgradeable/contracts/RoyaltiesV2Upgradeable.sol#11-13) is not in mixedCase
Function HasContractURI.__HasContractURI_init_unchained(string) (contracts/HasContractURI.sol#16-19) is not in mixedCase
Parameter HasContractURI.__HasContractURI_init_unchained(string)._contractURI (contracts/HasContractURI.sol#16) is not in mixedCase
Variable HasContractURI.__gap (contracts/HasContractURI.sol#29) is not in mixedCase
Parameter LibURI.checkPrefix(string,string)._tokenURI (contracts/LibURI.sol#7) is not in mixedCase
Parameter ERC1155Base.isApprovedForAll(address,address)._owner (contracts/erc-1155/ERC1155Base.sol#19) is not in mixedCase
Parameter ERC1155Base.isApprovedForAll(address,address)._operator (contracts/erc-1155/ERC1155Base.sol#19) is not in mixedCase
Function ERC1155Base.__ERC1155Base_init_unchained(string,string) (contracts/erc-1155/ERC1155Base.sol#73-76) is not in mixedCase
Parameter ERC1155Base.__ERC1155Base_init_unchained(string,string)._name (contracts/erc-1155/ERC1155Base.sol#73) is not in mixedCase
Parameter ERC1155Base.__ERC1155Base_init_unchained(string,string)._symbol (contracts/erc-1155/ERC1155Base.sol#73) is not in mixedCase
Variable ERC1155Base.__gap (contracts/erc-1155/ERC1155Base.sol#82) is not in mixedCase
Variable ERC1155BaseURI.__gap (contracts/erc-1155/ERC1155BaseURI.sol#69) is not in mixedCase
Function ERC1155BurnableUpgradeable.__ERC1155Burnable_init() (contracts/erc-1155/ERC1155BurnableUpgradeable.sol#15-19) is not in mixedCase
Function ERC1155BurnableUpgradeable.__ERC1155Burnable_init_unchained() (contracts/erc-1155/ERC1155BurnableUpgradeable.sol#21-22) is not in mixedCase
Variable ERC1155BurnableUpgradeable.__gap (contracts/erc-1155/ERC1155BurnableUpgradeable.sol#40) is not in mixedCase
Parameter ERC1155DefaultApproval.isApprovedForAll(address,address)._owner (contracts/erc-1155/ERC1155DefaultApproval.sol#17) is not in mixedCase
Parameter ERC1155DefaultApproval.isApprovedForAll(address,address)._operator (contracts/erc-1155/ERC1155DefaultApproval.sol#17) is not in mixedCase
Variable ERC1155DefaultApproval.__gap (contracts/erc-1155/ERC1155DefaultApproval.sol#20) is not in mixedCase

Variable ERC1155DefaultApproval.__gap (contracts/erc-1155/ERC1155DefaultApproval.sol#20) is not in mixedCase
Function ERC1155Lazy.__ERC1155Lazy_init_unchained() (contracts/erc-1155/ERC1155Lazy.sol#24-26) is not in mixedCase
Parameter ERC1155Lazy.mintAndTransfer(LibERC1155LazyMint.Mint1155Data,address,uint256)._amount (contracts/erc-1155/ERC1155Lazy.sol#57) is not in mixedCase
Parameter ERC1155Lazy.updateAccount(uint256,address,address)._id (contracts/erc-1155/ERC1155Lazy.sol#127) is not in mixedCase
Parameter ERC1155Lazy.updateAccount(uint256,address,address)._from (contracts/erc-1155/ERC1155Lazy.sol#127) is not in mixedCase
Parameter ERC1155Lazy.updateAccount(uint256,address,address)._to (contracts/erc-1155/ERC1155Lazy.sol#127) is not in mixedCase
Parameter ERC1155Lazy.getCreators(uint256)._id (contracts/erc-1155/ERC1155Lazy.sol#132) is not in mixedCase
Variable ERC1155Lazy.__gap (contracts/erc-1155/ERC1155Lazy.sol#140) is not in mixedCase
Function ERC1155Rarible.__ERC1155Rarible_init(string,string) (contracts/erc-1155/ERC1155Rarible.sol#15-23) is not in mixedCase
Parameter ERC1155Rarible.__ERC1155RaribleUser_init(string,string,string,string,address[],address,address)._name (contracts/erc-1155/ERC1155Rarible.sol#15) is not in mixedCase
Parameter ERC1155Rarible.__ERC1155RaribleUser_init(string,string,string,string,address[],address,address)._symbol (contracts/erc-1155/ERC1155Rarible.sol#15) is not in mixedCase
Function ERC1155Rarible.__ERC1155Rarible_init(string,string,address,address) (contracts/erc-1155/ERC1155Rarible.sol#25-30) is not in mixedCase
Parameter ERC1155Rarible.__ERC1155Rarible_init(string,string,string,string,address,address)._name (contracts/erc-1155/ERC1155Rarible.sol#25) is not in mixedCase
Parameter ERC1155Rarible.__ERC1155Rarible_init(string,string,string,string,address,address)._symbol (contracts/erc-1155/ERC1155Rarible.sol#25) is not in mixedCase
Function ERC1155Rarible.__ERC1155Rarible_init_unchained(string,string,address,address) (contracts/erc-1155/ERC1155Rarible.sol#32-48) is not in mixedCase
Parameter ERC1155Rarible.__ERC1155Rarible_init_unchained(string,string,string,string,address,address)._name (contracts/erc-1155/ERC1155Rarible.sol#32) is not in mixedCase
Parameter ERC1155Rarible.__ERC1155Rarible_init_unchained(string,string,string,string,address,address)._symbol (contracts/erc-1155/ERC1155Rarible.sol#32) is not in mixedCase
Parameter ERC1155Rarible.mintAndTransfer(ERC1155LazyMint.Mint1155Data,address,uint256)._amount (contracts/erc-1155/ERC1155Rarible.sol#50) is not in mixedCase
Variable ERC1155Rarible.__gap (contracts/erc-1155/ERC1155Rarible.sol#57) is not in mixedCase
Function ERC1155Upgradeable.__ERC1155_init(string) (contracts/erc-1155/ERC1155Upgradeable.sol#56-60) is not in mixedCase
Function ERC1155Upgradeable.__ERC1155_init_unchained(string) (contracts/erc-1155/ERC1155Upgradeable.sol#62-70) is not in mixedCase
Variable ERC1155Upgradeable.__gap (contracts/erc-1155/ERC1155Upgradeable.sol#421) is not in mixedCase
Variable ERC1155Upgradeable._balances (contracts/erc-1155/ERC1155Upgradeable.sol#27) is not in mixedCase
Function MintI155Validator.__MintI155Validator_init_unchained() (contracts/erc-1155/MintI155Validator.sol#9-11) is not in mixedCase
Variable MintI155Validator.__gap (contracts/erc-1155/MintI155Validator.sol#16) is not in mixedCase
Variable ERC1271Validator.__gap (contracts/erc-1271/ERC1271Validator.sol#34) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#94)" inEIP712Upgradeable (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#25-121)
Redundant expression "this (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#28)" inContextUpgradeable (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#16-32)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

ERC1271.slitherConstructorConstantVariables() (contracts/erc-1271/ERC1271.sol#5-26) uses literals with too many digits:
        - ERC1271_RETURN_INVALID_SIGNATURE = 0x00000000 (contracts/erc-1271/ERC1271.sol#9)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits
```

```
ERC1155ReceiverUpgradeable (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/ERC1155ReceiverUpgradeable.sol#12-25) does not implement functions:
        - IERC1155ReceiverUpgradeable.onERC1155BatchReceived(address,address,uint256[],uint256[],bytes) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/IERC1155ReceiverUpgradeable.sol#48-56)
        - IERC1155ReceiverUpgradeable.onERC1155Received(address,address,uint256,uint256,bytes) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/IERC1155ReceiverUpgradeable.sol#28-33)
ERC1271 (contracts/erc-1271/ERC1271.sol#5-26) does not implement functions:
        - ERC1271.isValidSignature(bytes32,bytes) (contracts/erc-1271/ERC1271.sol#20)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unimplemented-functions

ERC1155ReceiverUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/ERC1155ReceiverUpgradeable.sol#24) is never used in ERC1155ReceiverUpgradeable (node_modules/@openzeppelin/contracts-upgradeable/token/ERC11
55/ERC1155ReceiverUpgradeable.sol#12-25)
LibERC1155LazyMint._INTERFACE_ID_MINT_AND_TRANSFER (node_modules/@rarible/lazy-mint/contracts/erc-1155/LibERC1155LazyMint.sol#9) is never used in LibERC1155LazyMint (node_modules/@rarible/lazy-mint/contracts/erc-1155/LibERC1155LazyMint.s
ol#7-40)
ERC1155Rarible.__gap (contracts/erc-1155/ERC1155Rarible.sol#57) is never used in ERC1155Rarible (contracts/erc-1155/ERC1155Rarible.sol#8-58)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

renounceOwnership() should be declared external:
        - OwnableUpgradeable.renounceOwnership() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#60-63)
transferOwnership(address) should be declared external:
        - OwnableUpgradeable.transferOwnership(address) (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#65-73)
balanceOfBatch(address[],uint256[]) should be declared external:
        - ERC1155Upgradeable.balanceOfBatch(address[],uint256[]) (contracts/erc-1155/ERC1155Upgradeable.sol#105-124)
safeBatchTransferFrom(address,address,uint256[],uint256[],bytes) should be declared external:
        - ERC1155Upgradeable.safeBatchTransferFrom(address,address,uint256[],uint256[],bytes) (contracts/erc-1155/ERC1155Upgradeable.sol#178-214)
isValidSignature(bytes32,bytes) should be declared external:
        - ERC1271.isValidSignature(bytes32,bytes) (contracts/erc-1271/ERC1271.sol#20)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

ERC1155WoonklyNFTRevealWave.sol

```
OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#74) shadows:
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
ERC1155ReceiverUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/ERC1155ReceiverUpgradeable.sol#24) shadows:
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
HasContractURI.__gap (contracts/HasContractURI.sol#29) shadows:
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
ERC1155Base.__gap (contracts/erc-1155/ERC1155Base.sol#82) shadows:
        - HasContractURI.__gap (contracts/HasContractURI.sol#29)
        - ERC1155Lazy.__gap (contracts/erc-1155/ERC1155Lazy.sol#145)
        - Mint1155Validator.__gap (contracts/erc-1155/Mint1155Validator.sol#16)
        - ERC1271Validator.__gap (contracts/erc-1271/ERC1271Validator.sol#34)
        - EIP712Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#120)
        - ERC1155BaseURI.__gap (contracts/erc-1155/ERC1155BaseURI.sol#69)
        - ERC1155BurnableUpgradeable.__gap (contracts/erc-1155/ERC1155BurnableUpgradeable.sol#40)
        - ERC1155DefaultApproval.__gap (contracts/erc-1155/ERC1155DefaultApproval.sol#20)
        - ERC1155Upgradeable.__gap (contracts/erc-1155/ERC1155Upgradeable.sol#421)
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#74)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
ERC1155BaseURI.__gap (contracts/erc-1155/ERC1155BaseURI.sol#69) shadows:
        - ERC1155Upgradeable.__gap (contracts/erc-1155/ERC1155Upgradeable.sol#421)
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
ERC1155BurnableUpgradeable.__gap (contracts/erc-1155/ERC1155BurnableUpgradeable.sol#40) shadows:
        - ERC1155Upgradeable.__gap (contracts/erc-1155/ERC1155Upgradeable.sol#421)
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
ERC1155DefaultApproval.__gap (contracts/erc-1155/ERC1155DefaultApproval.sol#20) shadows:
        - ERC1155Upgradeable.__gap (contracts/erc-1155/ERC1155Upgradeable.sol#421)
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
ERC1155Lazy.__gap (contracts/erc-1155/ERC1155Lazy.sol#148) shadows:
        - Mint1155Validator.__gap (contracts/erc-1155/Mint1155Validator.sol#16)
        - ERC1271Validator.__gap (contracts/erc-1271/ERC1271Validator.sol#34)
        - EIP712Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#120)
        - ERC1155BaseURI.__gap (contracts/erc-1155/ERC1155BaseURI.sol#69)
        - ERC1155BurnableUpgradeable.__gap (contracts/erc-1155/ERC1155BurnableUpgradeable.sol#40)
        - ERC1155DefaultApproval.__gap (contracts/erc-1155/ERC1155DefaultApproval.sol#20)
        - ERC1155Upgradeable.__gap (contracts/erc-1155/ERC1155Upgradeable.sol#421)
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#74)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
ERC1155Upgradeable.__gap (contracts/erc-1155/ERC1155Upgradeable.sol#421) shadows:
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
Mint1155Validator.__gap (contracts/erc-1155/Mint1155Validator.sol#16) shadows:

        - ERC1271Validator.__gap (contracts/erc-1271/ERC1271Validator.sol#34)
        - EIP712Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#120)
ERC1271Validator.__gap (contracts/erc-1271/ERC1271Validator.sol#34) shadows:
        - EIP712Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#120)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variable-shadowing

Reentrancy in ERC1155Lazy.transferFromOrMint(LibERC1155LazyMint.Mint1155Data,address,address,uint256) (contracts/erc-1155/ERC1155Lazy.sol#36-55):
        External calls:
        - safeTransferFrom(from,to,data.tokenId,transfer,) (contracts/erc-1155/ERC1155Lazy.sol#49)
                - IERC1155ReceiverUpgradeable(to).onERC1155Received(operator,from,id,amount,data) (contracts/erc-1155/ERC1155Upgradeable.sol#380-388)
        - mintAndTransfer(data,to,left) (contracts/erc-1155/ERC1155Lazy.sol#53)
                - IERC1155ReceiverUpgradeable(to).onERC1155Received(operator,from,id,amount,data) (contracts/erc-1155/ERC1155Upgradeable.sol#380-388)
        State variables written after the call(s):
        - mintAndTransfer(data,to,left) (contracts/erc-1155/ERC1155Lazy.sol#53)
                - _balances[id][account] += _balances[id][account].add(amount) (contracts/erc-1155/ERC1155Lazy.sol#103)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-1

ERC1155Upgradeable._doSafeTransferAcceptanceCheck(address,address,address,uint256,uint256,bytes).reason (contracts/erc-1155/ERC1155Upgradeable.sol#384) is a local variable never initialized
ERC1155Upgradeable._doSafeTransferAcceptanceCheck(address,address,address,uint256,uint256,bytes).response (contracts/erc-1155/ERC1155Upgradeable.sol#380) is a local variable never initialized
ERC1155RaribleRevealWave._tokenURI(uint256)._tokenURI (contracts/erc-1155/ERC1155RaribleRevealWave.sol#153) is a local variable never initialized
ERC1271Validator.validate1271(address,bytes32,bytes).signerSlot (contracts/erc-1271/ERC1271Validator.sol#19) is a local variable never initialized
ERC1155Upgradeable._doSafeBatchTransferAcceptanceCheck(address,address,address,uint256[],uint256[],bytes).response (contracts/erc-1155/ERC1155Upgradeable.sol#403) is a local variable never initialized
ERC1155Upgradeable._doSafeBatchTransferAcceptanceCheck(address,address,address,uint256[],uint256[],bytes).reason (contracts/erc-1155/ERC1155Upgradeable.sol#407) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

ERC1155Upgradeable._doSafeTransferAcceptanceCheck(address,address,address,uint256,uint256,bytes) (contracts/erc-1155/ERC1155Upgradeable.sol#369-390) ignores return value by IERC1155ReceiverUpgradeable(to).onERC1155Received(operator,from,
id,amount,data) (contracts/erc-1155/ERC1155Upgradeable.sol#380-388)
ERC1155Upgradeable._doSafeBatchTransferAcceptanceCheck(address,address,address,uint256[],uint256[],bytes) (contracts/erc-1155/ERC1155Upgradeable.sol#392-413) ignores return value by IERC1155ReceiverUpgradeable(to).onERC1155BatchReceived(
operator,from,ids,amounts,data) (contracts/erc-1155/ERC1155Upgradeable.sol#403-411)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

ERC1155Base.isApprovedForAll(address,address)._owner (contracts/erc-1155/ERC1155Base.sol#19) shadows:
        - OwnableUpgradeable._owner (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#20) (state variable)
ERC1155Base._burnLazy(uint256,uint256).supply (contracts/erc-1155/ERC1155Base.sol#56) shadows:
        - ERC1155Lazy.supply (contracts/erc-1155/ERC1155Lazy.sol#21) (state variable)
ERC1155BaseURI._tokenURI(uint256)._tokenURI (contracts/erc-1155/ERC1155BaseURI.sol#33) shadows:
        - ERC1155BaseURI._tokenURI(uint256) (contracts/erc-1155/ERC1155BaseURI.sol#32-46) (function)
ERC1155BaseURI._setTokenURI(uint256,string)._uri (contracts/erc-1155/ERC1155BaseURI.sol#55) shadows:
        - ERC1155Upgradeable._uri (contracts/erc-1155/ERC1155Upgradeable.sol#33) (state variable)
ERC1155RaribleRevealWave.__ERC1155RaribleUser_init(string,string,string,string,address[],address,address).baseURI (contracts/erc-1155/ERC1155RaribleRevealWave.sol#29) shadows:
        - ERC1155BaseURI.baseURI() (contracts/erc-1155/ERC1155BaseURI.sol#24-26) (function)
ERC1155RaribleRevealWave.__ERC1155RaribleUser_init(string,string,string,string,address[],address,address).contractURI (contracts/erc-1155/ERC1155RaribleRevealWave.sol#29) shadows:
        - HasContractURI.contractURI (contracts/HasContractURI.sol#9) (state variable)
ERC1155RaribleRevealWave.__ERC1155Rarible_init(string,string,string,string,address,address).baseURI (contracts/erc-1155/ERC1155RaribleRevealWave.sol#39) shadows:
        - ERC1155BaseURI.baseURI() (contracts/erc-1155/ERC1155BaseURI.sol#24-26) (function)
ERC1155RaribleRevealWave.__ERC1155Rarible_init(string,string,string,string,address,address).contractURI (contracts/erc-1155/ERC1155RaribleRevealWave.sol#39) shadows:
        - HasContractURI.contractURI (contracts/HasContractURI.sol#9) (state variable)
ERC1155RaribleRevealWave.__ERC1155Rarible_init_unchained(string,string,string,string,address,address).baseURI (contracts/erc-1155/ERC1155RaribleRevealWave.sol#46) shadows:
        - ERC1155BaseURI.baseURI() (contracts/erc-1155/ERC1155BaseURI.sol#24-26) (function)
ERC1155RaribleRevealWave.__ERC1155Rarible_init_unchained(string,string,string,string,address,address).contractURI (contracts/erc-1155/ERC1155RaribleRevealWave.sol#46) shadows:
        - HasContractURI.contractURI (contracts/HasContractURI.sol#9) (state variable)
ERC1155RaribleRevealWave.changeHiddenBaseURI(uint256,string)._baseURI (contracts/erc-1155/ERC1155RaribleRevealWave.sol#89) shadows:
        - ERC1155BaseURI._baseURI (contracts/erc-1155/ERC1155BaseURI.sol#17) (state variable)
ERC1155RaribleRevealWave._tokenURI(uint256)._tokenURI (contracts/erc-1155/ERC1155RaribleRevealWave.sol#153) shadows:
        - ERC1155RaribleRevealWave._tokenURI(uint256) (contracts/erc-1155/ERC1155RaribleRevealWave.sol#147-169) (function)
        - ERC1155BaseURI._tokenURI(uint256) (contracts/erc-1155/ERC1155BaseURI.sol#32-46) (function)
ERC1155RaribleRevealWave._tokenURI(uint256)._baseURI (contracts/erc-1155/ERC1155RaribleRevealWave.sol#154) shadows:
        - ERC1155BaseURI._baseURI (contracts/erc-1155/ERC1155BaseURI.sol#17) (state variable)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

Variable 'ERC1155Upgradeable._doSafeTransferAcceptanceCheck(address,address,address,uint256,uint256,bytes).response (contracts/erc-1155/ERC1155Upgradeable.sol#380)' in ERC1155Upgradeable._doSafeTransferAcceptanceCheck(address,address,add
ress,uint256,uint256,bytes) (contracts/erc-1155/ERC1155Upgradeable.sol#369-390) potentially used before declaration: response != IERC1155ReceiverUpgradeable(to).onERC1155Received.selector (contracts/erc-1155/ERC1155Upgradeable.sol#383)
Variable 'ERC1155Upgradeable._doSafeTransferAcceptanceCheck(address,address,address,uint256,uint256,bytes).reason (contracts/erc-1155/ERC1155Upgradeable.sol#384)' in ERC1155Upgradeable._doSafeTransferAcceptanceCheck(address,address,addre
```

AUTOMATED TESTING

```
ss,uint256,uint256,bytes) (contracts/erc-1155/ERC1155Upgradeable.sol#369-390) potentially used before declaration: revert(string)(reason) (contracts/erc-1155/ERC1155Upgradeable.sol#385)
Variable 'ERC1155Upgradeable._doSafeBatchTransferAcceptanceCheck(address,address,address,uint256[],uint256[],bytes).response (contracts/erc-1155/ERC1155Upgradeable.sol#403)' in ERC1155Upgradeable._doSafeBatchTransferAcceptanceCheck(addre
ss,address,address,uint256[],uint256[],bytes) (contracts/erc-1155/ERC1155Upgradeable.sol#392-413) potentially used before declaration: response != IERC1155ReceiverUpgradeable(to).onERC1155BatchReceived.selector (contracts/erc-1155/ERC115
5Upgradeable.sol#404)
Variable 'ERC1155Upgradeable._doSafeBatchTransferAcceptanceCheck(address,address,address,uint256[],uint256[],bytes).reason (contracts/erc-1155/ERC1155Upgradeable.sol#407)' in ERC1155Upgradeable._doSafeBatchTransferAcceptanceCheck(address
,address,address,uint256[],uint256[],bytes) (contracts/erc-1155/ERC1155Upgradeable.sol#392-413) potentially used before declaration: revert(string)(reason) (contracts/erc-1155/ERC1155Upgradeable.sol#408)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables

Reentrancy in ERC1155Lazy.mintAndTransfer(LibERC1155LazyMint.Mint1155Data,address,uint256) (contracts/erc-1155/ERC1155Lazy.sol#57-90):
        External calls:
        - _mint(to,data.tokenId,_amount,) (contracts/erc-1155/ERC1155Lazy.sol#83)
            - IERC1155ReceiverUpgradeable(to).onERC1155Received(operator,from,id,amount,data) (contracts/erc-1155/ERC1155Upgradeable.sol#380-388)
        Event emitted after the call(s):
        - TransferSingle(sender,address(0),minter,data.tokenId,_amount) (contracts/erc-1155/ERC1155Lazy.sol#85)
        - TransferSingle(sender,minter,to,data.tokenId,_amount) (contracts/erc-1155/ERC1155Lazy.sol#86)
        - TransferSingle(sender,address(0),to,data.tokenId,_amount) (contracts/erc-1155/ERC1155Lazy.sol#88)
Reentrancy in ERC1155Lazy.transferFromOrMint(LibERC1155LazyMint.Mint1155Data,address,address,uint256) (contracts/erc-1155/ERC1155Lazy.sol#36-55):
        External calls:
        - safeTransferFrom(from,to,data.tokenId,transfer,) (contracts/erc-1155/ERC1155Lazy.sol#49)
            - IERC1155ReceiverUpgradeable(to).onERC1155Received(operator,from,id,amount,data) (contracts/erc-1155/ERC1155Upgradeable.sol#380-388)
        - mintAndTransfer(data,to,left) (contracts/erc-1155/ERC1155Lazy.sol#53)
            - IERC1155ReceiverUpgradeable(to).onERC1155Received(operator,from,id,amount,data) (contracts/erc-1155/ERC1155Upgradeable.sol#380-388)
        Event emitted after the call(s):
        - Creators(tokenId,_creators) (contracts/erc-1155/ERC1155Lazy.sol#124)
            - mintAndTransfer(data,to,left) (contracts/erc-1155/ERC1155Lazy.sol#53)
        - RoyaltiesSet(id,_royalties) (node_modules/@rarible/royalties/contracts/impl/RoyaltiesV2Impl.sol#16)
            - mintAndTransfer(data,to,left) (contracts/erc-1155/ERC1155Lazy.sol#53)
        - Supply(tokenId,_supply) (contracts/erc-1155/ERC1155Lazy.sol#111)
            - mintAndTransfer(data,to,left) (contracts/erc-1155/ERC1155Lazy.sol#53)
        - TransferSingle(sender,address(0),minter,data.tokenId,_amount) (contracts/erc-1155/ERC1155Lazy.sol#85)
            - mintAndTransfer(data,to,left) (contracts/erc-1155/ERC1155Lazy.sol#53)
        - TransferSingle(sender,minter,to,data.tokenId,_amount) (contracts/erc-1155/ERC1155Lazy.sol#86)
            - mintAndTransfer(data,to,left) (contracts/erc-1155/ERC1155Lazy.sol#53)
        - TransferSingle(sender,address(0),to,data.tokenId,_amount) (contracts/erc-1155/ERC1155Lazy.sol#88)
            - mintAndTransfer(data,to,left) (contracts/erc-1155/ERC1155Lazy.sol#53)
        - URI(_tokenURI(tokenId),tokenId) (contracts/erc-1155/ERC1155BaseURI.sol#57)
            - mintAndTransfer(data,to,left) (contracts/erc-1155/ERC1155Lazy.sol#53)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

EIP712Upgradeable.__getChainId() (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#93-99) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#96-98)
AddressUpgradeable.isContract(address) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#26-35) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#33)
AddressUpgradeable._verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#147-164) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#156-159)
LibSignature.recover(bytes32,bytes) (node_modules/@rarible/libraries/contracts/LibSignature.sol#20-45) uses assembly
        - INLINE ASM (node_modules/@rarible/libraries/contracts/LibSignature.sol#38-42)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

ERC1155RaribleRevealWave._tokenURI(uint256) (contracts/erc-1155/ERC1155RaribleRevealWave.sol#147-169) compares to a boolean constant:
        -revealWave.isRevealed == false (contracts/erc-1155/ERC1155RaribleRevealWave.sol#161)
ERC1155RaribleRevealWave._tokenURI(uint256) (contracts/erc-1155/ERC1155RaribleRevealWave.sol#147-169) compares to a boolean constant:
        -revealWave.isRevealed == true || revealWave.addToNftURITOHiddenBaseURI == true (contracts/erc-1155/ERC1155RaribleRevealWave.sol#156)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#boolean-equality

Different versions of Solidity is used:
        - Version used: ['0.7.6', '>=0.4.24<0.8.0', '>=0.6.0<0.8.0', '>=0.6.2<0.8.0', '^0.7.0']
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/introspection/IERC165Upgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#3)
        - >=0.4.24<0.8.0 (node_modules/@openzeppelin/proxy/Initializable.sol#4)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/ERC1155ReceiverUpgradeable.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/IERC1155MetadataURIUpgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/IERC1155ReceiverUpgradeable.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/IERC1155Upgradeable.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/StringsUpgradeable.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@rarible/lazy-mint/contracts/erc-1155/IERC1155LazyMint.sol#3)
        - v2 (node_modules/@rarible/lazy-mint/contracts/erc-1155/IERC1155LazyMint.sol#4)
        - >=0.6.2<0.8.0 (node_modules/@rarible/lazy-mint/contracts/erc-1155/LibERC1155LazyMint.sol#3)
        - ^0.7.0 (node_modules/@rarible/libraries/contracts/LibSignature.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@rarible/royalties-upgradeable/contracts/RoyaltiesV2Upgradeable.sol#3)
        - v2 (node_modules/@rarible/royalties-upgradeable/contracts/RoyaltiesV2Upgradeable.sol#4)
        - >=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/LibPart.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/LibRoyaltiesV2.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/RoyaltiesV2.sol#3)
        - v2 (node_modules/@rarible/royalties/contracts/RoyaltiesV2.sol#4)
        - >=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/impl/AbstractRoyalties.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/impl/RoyaltiesV2Impl.sol#3)
        - v2 (node_modules/@rarible/royalties/contracts/impl/RoyaltiesV2Impl.sol#4)
        - 0.7.6 (contracts/HasContractURI.sol#3)
        - 0.7.6 (contracts/LibURI.sol#3)
        - 0.7.6 (contracts/erc-1155/ERC1155Base.sol#3)
        - v2 (contracts/erc-1155/ERC1155Base.sol#4)
        - 0.7.6 (contracts/erc-1155/ERC1155BaseURI.sol#3)
        - >=0.6.0<0.8.0 (contracts/erc-1155/ERC1155BurnableUpgradeable.sol#3)
        - 0.7.6 (contracts/erc-1155/ERC1155DefaultApproval.sol#3)
        - 0.7.6 (contracts/erc-1155/ERC1155Lazy.sol#3)
        - v2 (contracts/erc-1155/ERC1155Lazy.sol#4)
        - 0.7.6 (contracts/erc-1155/ERC1155RaribleRevealWave.sol#3)
        - v2 (contracts/erc-1155/ERC1155RaribleRevealWave.sol#4)
        - >=0.6.0<0.8.0 (contracts/erc-1155/ERC1155Upgradeable.sol#3)
        - 0.7.6 (contracts/erc-1155/Mint1155Validator.sol#3)
        - 0.7.6 (contracts/erc-1271/ERC1271.sol#3)
        - 0.7.6 (contracts/erc-1271/ERC1271Validator.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

AbstractRoyalties._onRoyaltiesSet(uint256,LibPart.Part[]) (node_modules/@rarible/royalties/contracts/impl/AbstractRoyalties.sol#31) is never used and should be removed
AddressUpgradeable._verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#147-164) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#79-81) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#89-91) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#104-106) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#114-121) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#129-131) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#139-145) is never used and should be removed
AddressUpgradeable.sendValue(address,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#53-59) is never used and should be removed
ContextUpgradeable.__Context_init() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#17-19) is never used and should be removed
ContextUpgradeable._msgData() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#27-30) is never used and should be removed
EIP712Upgradeable.__EIP712_init(string,string) (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#44-46) is never used and should be removed
ERC1155BurnableUpgradeable.__ERC1155Burnable_init() (contracts/erc-1155/ERC1155BurnableUpgradeable.sol#15-19) is never used and should be removed
ERC1155ReceiverUpgradeable.__ERC1155Receiver_init() (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/ERC1155ReceiverUpgradeable.sol#23-16) is never used and should be removed
ERC1155ReceiverUpgradeable.__ERC1155Receiver_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/ERC1155ReceiverUpgradeable.sol#18-23) is never used and should be removed
ERC1155Upgradeable.__ERC1155_init(string) (contracts/erc-1155/ERC1155Upgradeable.sol#56-60) is never used and should be removed
ERC1155Upgradeable._mint(address,uint256,uint256,bytes) (contracts/erc-1155/ERC1155Upgradeable.sol#250-261) is never used and should be removed
ERC1155Upgradeable._mintBatch(address,uint256[],uint256[],bytes) (contracts/erc-1155/ERC1155Upgradeable.sol#272-287) is never used and should be removed
ERC1271.returnIsValidSignatureMagicNumber(bool) (contracts/erc-1271/ERC1271.sol#22-24) is never used and should be removed
ERC165Upgradeable.__ERC165_init() (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#25-27) is never used and should be removed
HasContractURI._setContractURI(string) (contracts/HasContractURI.sol#25-27) is never used and should be removed
OwnableUpgradeable.__Ownable_init() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#27-30) is never used and should be removed
SafeMathUpgradeable.div(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#135-138) is never used and should be removed
SafeMathUpgradeable.div(uint256,uint256,string) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#190-193) is never used and should be removed
SafeMathUpgradeable.mod(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#152-155) is never used and should be removed
SafeMathUpgradeable.mod(uint256,uint256,string) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#210-213) is never used and should be removed
SafeMathUpgradeable.mul(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#116-121) is never used and should be removed
SafeMathUpgradeable.sub(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#101-104) is never used and should be removed
SafeMathUpgradeable.tryAdd(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#24-28) is never used and should be removed
SafeMathUpgradeable.tryDiv(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#60-63) is never used and should be removed
SafeMathUpgradeable.tryMod(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#70-73) is never used and should be removed
SafeMathUpgradeable.tryMul(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#45-53) is never used and should be removed
SafeMathUpgradeable.trySub(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#35-38) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/introspection/IERC165Upgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#3) is too complex
Pragma version>=0.4.24<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/proxy/Initializable.sol#4) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/ERC1155ReceiverUpgradeable.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/IERC1155MetadataURIUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/IERC1155ReceiverUpgradeable.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/IERC1155Upgradeable.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/StringsUpgradeable.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/lazy-mint/contracts/erc-1155/IERC1155LazyMint.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/lazy-mint/contracts/erc-1155/LibERC1155LazyMint.sol#3) is too complex
Pragma version^0.7.0 (node_modules/@rarible/libraries/contracts/LibSignature.sol#3) allows old versions
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties-upgradeable/contracts/RoyaltiesV2Upgradeable.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/LibPart.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/LibRoyaltiesV2.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/RoyaltiesV2.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/impl/AbstractRoyalties.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/impl/RoyaltiesV2Impl.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (contracts/erc-1155/ERC1155BurnableUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (contracts/erc-1155/ERC1155Upgradeable.sol#3) is too complex
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

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

```
Function OwnableUpgradeable.__Ownable_init() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#27-30) is not in mixedCase
Function OwnableUpgradeable.__Ownable_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#32-36) is not in mixedCase
Variable OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#74) is not in mixedCase
Function EIP712Upgradeable.__EIP712_init(string,string) (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#44-46) is not in mixedCase
Function EIP712Upgradeable.__EIP712_init_unchained(string,string) (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#48-53) is not in mixedCase
Function EIP712Upgradeable._EIP712NameHash() (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#107-109) is not in mixedCase
Function EIP712Upgradeable._EIP712VersionHash() (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#117-119) is not in mixedCase
Variable EIP712Upgradeable._HASHED_NAME (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#27) is not in mixedCase
Variable EIP712Upgradeable._HASHED_VERSION (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#28) is not in mixedCase
Variable EIP712Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#120) is not in mixedCase
Function ERC165Upgradeable.__ERC165_init() (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#25-27) is not in mixedCase
Function ERC165Upgradeable.__ERC165_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#29-33) is not in mixedCase
Variable ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59) is not in mixedCase
Function ERC1155ReceiverUpgradeable.__ERC1155Receiver_init() (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/ERC1155ReceiverUpgradeable.sol#13-16) is not in mixedCase
Function ERC1155ReceiverUpgradeable.__ERC1155Receiver_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/ERC1155ReceiverUpgradeable.sol#18-23) is not in mixedCase
Variable ERC1155ReceiverUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/ERC1155ReceiverUpgradeable.sol#24) is not in mixedCase
Function ContextUpgradeable.__Context_init() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#17-18) is not in mixedCase
Function ContextUpgradeable.__Context_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#21-22) is not in mixedCase
Variable ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31) is not in mixedCase
Function RoyaltiesV2Upgradeable.__RoyaltiesV2Upgradeable_init_unchained() (node_modules/@rarible/royalties-upgradeable/contracts/RoyaltiesV2Upgradeable.sol#11-13) is not in mixedCase
Function HasContractURI.__HasContractURI_init_unchained(string) (contracts/HasContractURI.sol#16-19) is not in mixedCase
Parameter HasContractURI.__HasContractURI_init_unchained(string)._contractURI (contracts/HasContractURI.sol#16) is not in mixedCase
Variable HasContractURI.__gap (contracts/HasContractURI.sol#29) is not in mixedCase
Parameter LibURI.checkPrefix(string,string)._tokenURI (contracts/LibURI.sol#7) is not in mixedCase
Parameter ERC1155Base.isApprovedForAll(address,address)._owner (contracts/erc-1155/ERC1155Base.sol#19) is not in mixedCase
Parameter ERC1155Base.isApprovedForAll(address,address)._operator (contracts/erc-1155/ERC1155Base.sol#19) is not in mixedCase
Function ERC1155Base.__ERC1155Base_init_unchained(uint256,string) (contracts/erc-1155/ERC1155Base.sol#73-76) is not in mixedCase
Parameter ERC1155Base.__ERC1155Base_init_unchained(uint256,string)._name (contracts/erc-1155/ERC1155Base.sol#73) is not in mixedCase
Parameter ERC1155Base.__ERC1155Base_init_unchained(uint256,string)._symbol (contracts/erc-1155/ERC1155Base.sol#73) is not in mixedCase
Variable ERC1155Base.__gap (contracts/erc-1155/ERC1155Base.sol#83) is not in mixedCase
Variable ERC1155BaseURI.__gap (contracts/erc-1155/ERC1155BaseURI.sol#69) is not in mixedCase
Function ERC1155BurnableUpgradeable.__ERC1155Burnable_init() (contracts/erc-1155/ERC1155BurnableUpgradeable.sol#15-19) is not in mixedCase
Function ERC1155BurnableUpgradeable.__ERC1155Burnable_init_unchained() (contracts/erc-1155/ERC1155BurnableUpgradeable.sol#21-22) is not in mixedCase
Variable ERC1155BurnableUpgradeable.__gap (contracts/erc-1155/ERC1155BurnableUpgradeable.sol#40) is not in mixedCase
Parameter ERC1155DefaultApproval.isApprovedForAll(address,address)._owner (contracts/erc-1155/ERC1155DefaultApproval.sol#17) is not in mixedCase
Parameter ERC1155DefaultApproval.isApprovedForAll(address,address)._operator (contracts/erc-1155/ERC1155DefaultApproval.sol#17) is not in mixedCase
Variable ERC1155DefaultApproval.__gap (contracts/erc-1155/ERC1155DefaultApproval.sol#20) is not in mixedCase
Function ERC1155Lazy.__ERC1155Lazy_init_unchained() (contracts/erc-1155/ERC1155Lazy.sol#24-26) is not in mixedCase
Parameter ERC1155Lazy.mintAndTransfer(LibERC1155LazyMint.Mint1155Data,address,uint256)._amount (contracts/erc-1155/ERC1155Lazy.sol#57) is not in mixedCase
Parameter ERC1155Lazy.updateAccount(uint256,address,address)._id (contracts/erc-1155/ERC1155Lazy.sol#127) is not in mixedCase
Parameter ERC1155Lazy.updateAccount(uint256,address,address)._from (contracts/erc-1155/ERC1155Lazy.sol#127) is not in mixedCase
Parameter ERC1155Lazy.updateAccount(uint256,address,address)._to (contracts/erc-1155/ERC1155Lazy.sol#127) is not in mixedCase
Parameter ERC1155Lazy.getCreators(uint256)._id (contracts/erc-1155/ERC1155Lazy.sol#132) is not in mixedCase
Variable ERC1155Lazy.__gap (contracts/erc-1155/ERC1155Lazy.sol#148) is not in mixedCase
Function ERC1155RaribleRevealWave.__ERC1155RaribleUser_init(string,string,string,string,address[],address,address) (contracts/erc-1155/ERC1155RaribleRevealWave.sol#29-37) is not in mixedCase
Parameter ERC1155RaribleRevealWave.__ERC1155RaribleUser_init(string,string,string,string,address[],address,address)._name (contracts/erc-1155/ERC1155RaribleRevealWave.sol#29) is not in mixedCase
Parameter ERC1155RaribleRevealWave.__ERC1155RaribleUser_init(string,string,string,string,address[],address,address)._symbol (contracts/erc-1155/ERC1155RaribleRevealWave.sol#29) is not in mixedCase
Function ERC1155RaribleRevealWave.__ERC1155Rarible_init(string,string,string,string,address,address) (contracts/erc-1155/ERC1155RaribleRevealWave.sol#39-44) is not in mixedCase
Parameter ERC1155RaribleRevealWave.__ERC1155Rarible_init(string,string,string,string,address,address)._name (contracts/erc-1155/ERC1155RaribleRevealWave.sol#39) is not in mixedCase
Parameter ERC1155RaribleRevealWave.__ERC1155Rarible_init(string,string,string,string,address,address)._symbol (contracts/erc-1155/ERC1155RaribleRevealWave.sol#39) is not in mixedCase
Function ERC1155RaribleRevealWave.__ERC1155Rarible_init_unchained(string,string,string,string,address,address) (contracts/erc-1155/ERC1155RaribleRevealWave.sol#46-62) is not in mixedCase
Parameter ERC1155RaribleRevealWave.__ERC1155Rarible_init_unchained(string,string,string,string,address,address)._name (contracts/erc-1155/ERC1155RaribleRevealWave.sol#46) is not in mixedCase
Parameter ERC1155RaribleRevealWave.__ERC1155Rarible_init_unchained(string,string,string,string,address,address)._symbol (contracts/erc-1155/ERC1155RaribleRevealWave.sol#46) is not in mixedCase
Parameter ERC1155RaribleRevealWave.getRevealWaveIdByTokenId(uint256)._tokenId (contracts/erc-1155/ERC1155RaribleRevealWave.sol#66) is not in mixedCase
Parameter ERC1155RaribleRevealWave.setRevealWave(uint256,string,string,bool,uint256)._waveId (contracts/erc-1155/ERC1155RaribleRevealWave.sol#73) is not in mixedCase
Parameter ERC1155RaribleRevealWave.setRevealWave(uint256,string,string,bool,uint256)._name (contracts/erc-1155/ERC1155RaribleRevealWave.sol#74) is not in mixedCase
Parameter ERC1155RaribleRevealWave.setRevealWave(uint256,string,string,bool,uint256)._hiddenBaseURI (contracts/erc-1155/ERC1155RaribleRevealWave.sol#75) is not in mixedCase
Parameter ERC1155RaribleRevealWave.setRevealWave(uint256,string,string,bool,uint256)._addTokenURIToHiddenBaseURI (contracts/erc-1155/ERC1155RaribleRevealWave.sol#76) is not in mixedCase
Parameter ERC1155RaribleRevealWave.setRevealWave(uint256,string,string,bool,uint256)._revealDate (contracts/erc-1155/ERC1155RaribleRevealWave.sol#77) is not in mixedCase
Parameter ERC1155RaribleRevealWave.changeHiddenBaseURI(uint256,string)._waveId (contracts/erc-1155/ERC1155RaribleRevealWave.sol#88) is not in mixedCase
Parameter ERC1155RaribleRevealWave.changeHiddenBaseURI(uint256,string)._baseURI (contracts/erc-1155/ERC1155RaribleRevealWave.sol#88) is not in mixedCase
Parameter ERC1155RaribleRevealWave.changeAddTokenURITOHiddenBaseURI(uint256,bool)._waveId (contracts/erc-1155/ERC1155RaribleRevealWave.sol#97) is not in mixedCase

Parameter ERC1155RaribleRevealWave.changeAddTokenURITOHiddenBaseURI(uint256,bool)._addTokenURITOHiddenBaseURI (contracts/erc-1155/ERC1155RaribleRevealWave.sol#98) is not in mixedCase
Parameter ERC1155RaribleRevealWave.changeRevealBaseURI(uint256,string)._waveId (contracts/erc-1155/ERC1155RaribleRevealWave.sol#105) is not in mixedCase
Parameter ERC1155RaribleRevealWave.changeRevealBaseURI(uint256,string)._revealBaseURI (contracts/erc-1155/ERC1155RaribleRevealWave.sol#106) is not in mixedCase
Parameter ERC1155RaribleRevealWave.changeRevealDate(uint256,uint256)._waveId (contracts/erc-1155/ERC1155RaribleRevealWave.sol#116) is not in mixedCase
Parameter ERC1155RaribleRevealWave.changeRevealDate(uint256,uint256)._revealDate (contracts/erc-1155/ERC1155RaribleRevealWave.sol#117) is not in mixedCase
Parameter ERC1155RaribleRevealWave.resetIsRevealedAndRevealURI(uint256)._waveId (contracts/erc-1155/ERC1155RaribleRevealWave.sol#124) is not in mixedCase
Parameter ERC1155RaribleRevealWave.changeName(uint256,string)._waveId (contracts/erc-1155/ERC1155RaribleRevealWave.sol#133) is not in mixedCase
Parameter ERC1155RaribleRevealWave.changeName(uint256,string)._name (contracts/erc-1155/ERC1155RaribleRevealWave.sol#134) is not in mixedCase
Parameter ERC1155RaribleRevealWave.assignRevealWaveIdToTokenId(uint256,uint256)._tokenId (contracts/erc-1155/ERC1155RaribleRevealWave.sol#142) is not in mixedCase
Parameter ERC1155RaribleRevealWave.assignRevealWaveIdToTokenId(uint256,uint256)._waveId (contracts/erc-1155/ERC1155RaribleRevealWave.sol#142) is not in mixedCase
Parameter ERC1155RaribleRevealWave.mintAndTransfer(LibERC1155LazyMint.Mint1155Data,address,uint256)._amount (contracts/erc-1155/ERC1155RaribleRevealWave.sol#175) is not in mixedCase
Variable ERC1155RaribleRevealWave.__gap (contracts/erc-1155/ERC1155RaribleRevealWave.sol#196) is not in mixedCase
Function ERC1155Upgradeable.__ERC1155_init(string) (contracts/erc-1155/ERC1155Upgradeable.sol#56-60) is not in mixedCase
Function ERC1155Upgradeable.__ERC1155_init_unchained(string) (contracts/erc-1155/ERC1155Upgradeable.sol#62-70) is not in mixedCase
Variable ERC1155Upgradeable.__gap (contracts/erc-1155/ERC1155Upgradeable.sol#421) is not in mixedCase
Variable ERC1155Upgradeable._balances (contracts/erc-1155/ERC1155Upgradeable.sol#27) is not in mixedCase
Function Mint1155Validator.__Mint1155Validator_init_unchained() (contracts/erc-1155/Mint1155Validator.sol#9-11) is not in mixedCase
Variable Mint1155Validator.__gap (contracts/erc-1155/Mint1155Validator.sol#16) is not in mixedCase
Variable ERC1271Validator.__gap (contracts/erc-1271/ERC1271Validator.sol#34) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#94)" inEIP712Upgradeable (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#25-121)
Redundant expression "this (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#28)" inContextUpgradeable (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#16-32)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

ERC1271.slitherConstructorConstantVariables() (contracts/erc-1271/ERC1271.sol#5-26) uses literals with too many digits:
        - ERC1271_RETURN_INVALID_SIGNATURE = 0x00000000 (contracts/erc-1271/ERC1271.sol#9)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits

ERC1155ReceiverUpgradeable (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/ERC1155ReceiverUpgradeable.sol#12-25) does not implement functions:
        - IERC1155ReceiverUpgradeable.onERC1155BatchReceived(address,address,uint256[],uint256[],bytes) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/IERC1155ReceiverUpgradeable.sol#48-56)
        - IERC1155ReceiverUpgradeable.onERC1155Received(address,address,uint256,uint256,bytes) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/IERC1155ReceiverUpgradeable.sol#25-33)
ERC1271 (contracts/erc-1271/ERC1271.sol#5-26) does not implement functions:
        - ERC1271.isValidSignature(bytes32,bytes) (contracts/erc-1271/ERC1271.sol#20)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unimplemented-functions

ERC1155ReceiverUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/ERC1155ReceiverUpgradeable.sol#24) is never used in ERC1155ReceiverUpgradeable (node_modules/@openzeppelin/contracts-upgradeable/token/ERC11
55/ERC1155ReceiverUpgradeable.sol#12-25)
LibERC1155LazyMint._INTERFACE_ID_MINT_AND_TRANSFER (node_modules/@rarible/lazy-mint/contracts/erc-1155/LibERC1155LazyMint.sol#9) is never used in LibERC1155LazyMint (node_modules/@rarible/lazy-mint/contracts/erc-1155/LibERC1155LazyMint.s
ol#7-40)
ERC1155RaribleRevealWave.__gap (contracts/erc-1155/ERC1155RaribleRevealWave.sol#196) is never used in ERC1155RaribleRevealWave (contracts/erc-1155/ERC1155RaribleRevealWave.sol#8-197)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

renounceOwnership() should be declared external:
        - OwnableUpgradeable.renounceOwnership() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#60-63)
transferOwnership(address) should be declared external:
        - OwnableUpgradeable.transferOwnership(address) (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#69-73)
setRevealWave(uint256,string,string,bool,uint256) should be declared external:
        - ERC1155RaribleRevealWave.setRevealWave(uint256,string,string,bool,uint256) (contracts/erc-1155/ERC1155RaribleRevealWave.sol#72-85)
changeHiddenBaseURI(uint256,string) should be declared external:
        - ERC1155RaribleRevealWave.changeHiddenBaseURI(uint256,string) (contracts/erc-1155/ERC1155RaribleRevealWave.sol#87-94)
changeAddTokenURITOHiddenBaseURI(uint256,bool) should be declared external:
        - ERC1155RaribleRevealWave.changeAddTokenURITOHiddenBaseURI(uint256,bool) (contracts/erc-1155/ERC1155RaribleRevealWave.sol#96-102)
changeRevealBaseURI(uint256,string) should be declared external:
        - ERC1155RaribleRevealWave.changeRevealBaseURI(uint256,string) (contracts/erc-1155/ERC1155RaribleRevealWave.sol#104-113)
changeRevealDate(uint256,uint256) should be declared external:
        - ERC1155RaribleRevealWave.changeRevealDate(uint256,uint256) (contracts/erc-1155/ERC1155RaribleRevealWave.sol#115-121)
resetIsRevealedAndRevealURI(uint256) should be declared external:
        - ERC1155RaribleRevealWave.resetIsRevealedAndRevealURI(uint256) (contracts/erc-1155/ERC1155RaribleRevealWave.sol#123-129)
changeName(uint256,string) should be declared external:
        - ERC1155RaribleRevealWave.changeName(uint256,string) (contracts/erc-1155/ERC1155RaribleRevealWave.sol#132-135)
assignRevealWaveIdToTokenId(uint256,uint256) should be declared external:
        - ERC1155RaribleRevealWave.assignRevealWaveIdToTokenId(uint256,uint256) (contracts/erc-1155/ERC1155RaribleRevealWave.sol#142-145)
balanceOfBatch(address[],uint256[]) should be declared external:
        - ERC1155Upgradeable.balanceOfBatch(address[],uint256[]) (contracts/erc-1155/ERC1155Upgradeable.sol#105-124)
safeBatchTransferFrom(address,address,uint256[],uint256[],bytes) should be declared external:
        - ERC1155Upgradeable.safeBatchTransferFrom(address,address,uint256[],uint256[],bytes) (contracts/erc-1155/ERC1155Upgradeable.sol#178-214)
isValidSignature(bytes32,bytes4) should be declared external:
        - ERC1271.isValidSignature(bytes32,bytes) (contracts/erc-1271/ERC1271.sol#20)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

ERC1155WoonklyNFTFactoryC2.sol

```
OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#74) shadows:
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
ERC1155ReceiverUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/ERC1155ReceiverUpgradeable.sol#24) shadows:
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
HasContractURI.__gap (contracts/HasContractURI.sol#29) shadows:
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
ERC1155Base.__gap (contracts/erc-1155/ERC1155Base.sol#82) shadows:
        - HasContractURI.__gap (contracts/HasContractURI.sol#29)
        - ERC1155Lazy.__gap (contracts/erc-1155/ERC1155Lazy.sol#148)
        - Mint1155Validator.__gap (contracts/erc-1155/Mint1155Validator.sol#16)
        - ERC1271Validator.__gap (contracts/erc-1271/ERC1271Validator.sol#34)
        - EIP712Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#120)
        - ERC1155BaseURI.__gap (contracts/erc-1155/ERC1155BaseURI.sol#69)
        - ERC1155BurnableUpgradeable.__gap (contracts/erc-1155/ERC1155BurnableUpgradeable.sol#40)
        - ERC1155DefaultApproval.__gap (contracts/erc-1155/ERC1155DefaultApproval.sol#20)
        - ERC1155Upgradeable.__gap (contracts/erc-1155/ERC1155Upgradeable.sol#421)
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#74)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
ERC1155BaseURI.__gap (contracts/erc-1155/ERC1155BaseURI.sol#69) shadows:
        - ERC1155Upgradeable.__gap (contracts/erc-1155/ERC1155Upgradeable.sol#421)
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
```

AUTOMATED TESTING

```
ERC1155BurnableUpgradeable.__gap (contracts/erc-1155/ERC1155BurnableUpgradeable.sol#40) shadows:
        - ERC1155Upgradeable.__gap (contracts/erc-1155/ERC1155Upgradeable.sol#421)
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
ERC1155DefaultApproval.__gap (contracts/erc-1155/ERC1155DefaultApproval.sol#20) shadows:
        - ERC1155Upgradeable.__gap (contracts/erc-1155/ERC1155Upgradeable.sol#421)
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
ERC1155Lazy.__gap (contracts/erc-1155/ERC1155Lazy.sol#148) shadows:
        - Mint1155Validator.__gap (contracts/erc-1155/Mint1155Validator.sol#16)
        - ERC1271Validator.__gap (contracts/erc-1271/ERC1271Validator.sol#34)
        - EIP712Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#120)
        - ERC1155BaseURI.__gap (contracts/erc-1155/ERC1155BaseURI.sol#69)
        - ERC1155Upgradeable.__gap (contracts/erc-1155/ERC1155Upgradeable.sol#421)
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
ERC1155Lazy._INTERFACE_ID_ERC165 (contracts/erc-1155/ERC1155Lazy.sol#16) shadows:
        - ERC165Upgradeable._INTERFACE_ID_ERC165 (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#18)
ERC1155Lazy._INTERFACE_ID_ERC1155 (contracts/erc-1155/ERC1155Lazy.sol#17) shadows:
        - ERC1155Upgradeable._INTERFACE_ID_ERC1155 (contracts/erc-1155/ERC1155Upgradeable.sol#46)
ERC1155Lazy._INTERFACE_ID_ERC1155_METADATA_URI (contracts/erc-1155/ERC1155Lazy.sol#18) shadows:
        - ERC1155Upgradeable._INTERFACE_ID_ERC1155_METADATA_URI (contracts/erc-1155/ERC1155Upgradeable.sol#51)
ERC1155Rarible.__gap (contracts/erc-1155/ERC1155Rarible.sol#57) shadows:
        - ERC1155Base.__gap (contracts/erc-1155/ERC1155Base.sol#82)
        - HasContractURI.__gap (contracts/HasContractURI.sol#29)
        - ERC1155Lazy.__gap (contracts/erc-1155/ERC1155Lazy.sol#148)
        - Mint1155Validator.__gap (contracts/erc-1155/Mint1155Validator.sol#16)
        - ERC1271Validator.__gap (contracts/erc-1271/ERC1271Validator.sol#34)
        - EIP712Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#120)
        - ERC1155BaseURI.__gap (contracts/erc-1155/ERC1155BaseURI.sol#69)
        - ERC1155BurnableUpgradeable.__gap (contracts/erc-1155/ERC1155BurnableUpgradeable.sol#40)
        - ERC1155DefaultApproval.__gap (contracts/erc-1155/ERC1155DefaultApproval.sol#20)
        - ERC1155Upgradeable.__gap (contracts/erc-1155/ERC1155Upgradeable.sol#421)
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#74)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
ERC1155Upgradeable.__gap (contracts/erc-1155/ERC1155Upgradeable.sol#421) shadows:
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
Mint1155Validator.__gap (contracts/erc-1155/Mint1155Validator.sol#16) shadows:
        - ERC1271Validator.__gap (contracts/erc-1271/ERC1271Validator.sol#34)
        - EIP712Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#120)
ERC1271Validator.__gap (contracts/erc-1271/ERC1271Validator.sol#34) shadows:
        - EIP712Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#120)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variable-shadowing

Reentrancy in ERC1155Lazy.transferFromOrMint(LibERC1155LazyMint.Mint1155Data,address,address,uint256) (contracts/erc-1155/ERC1155Lazy.sol#36-55):
        External calls:
        - safeTransferFrom(from,to,data.tokenId,transfer,) (contracts/erc-1155/ERC1155Lazy.sol#49)
                - IERC1155ReceiverUpgradeable(to).onERC1155Received(operator,from,id,amount,data) (contracts/erc-1155/ERC1155Upgradeable.sol#380-388)
        - mintAndTransfer(data,to,left) (contracts/erc-1155/ERC1155Lazy.sol#53)
                - IERC1155ReceiverUpgradeable(to).onERC1155Received(operator,from,id,amount,data) (contracts/erc-1155/ERC1155Upgradeable.sol#380-388)
        State variables written after the call(s):
        - mintAndTransfer(data,to,left) (contracts/erc-1155/ERC1155Lazy.sol#53)
                - _balances[id][account] = _balances[id][account].add(amount) (contracts/erc-1155/ERC1155Lazy.sol#103)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-1

ERC1271Validator.validate1271(address,bytes32,bytes).signerFromSig (contracts/erc-1271/ERC1271Validator.sol#19) is a local variable never initialized
ERC1155Upgradeable._doSafeBatchTransferAcceptanceCheck(address,address,address,uint256,uint256,bytes) (contracts/erc-1155/ERC1155Upgradeable.sol#403) is a local variable never initialized
ERC1155Upgradeable._doSafeTransferAcceptanceCheck(address,address,address,uint256,uint256,bytes).response (contracts/erc-1155/ERC1155Upgradeable.sol#384) is a local variable never initialized
ERC1155Upgradeable._doSafeBatchTransferAcceptanceCheck(address,address,address,uint256[],uint256[],bytes).response (contracts/erc-1155/ERC1155Upgradeable.sol#407) is a local variable never initialized
ERC1155Upgradeable._doSafeTransferAcceptanceCheck(address,address,address,uint256,uint256,bytes).response (contracts/erc-1155/ERC1155Upgradeable.sol#380) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

BeaconProxy._setBeacon(address,bytes) (node_modules/@openzeppelin/contracts/proxy/BeaconProxy.sol#68-87) ignores return value by Address.functionDelegateCall(_implementation(),data,BeaconProxy: function call failed) (node_modules/@openze
ppelin/contracts/proxy/BeaconProxy.sol#85)
ERC1155Upgradeable._doSafeTransferAcceptanceCheck(address,address,address,uint256,uint256,bytes) (contracts/erc-1155/ERC1155Upgradeable.sol#369-390) ignores return value by IERC1155ReceiverUpgradeable(to).onERC1155Received(operator,from,
id,amount,data) (contracts/erc-1155/ERC1155Upgradeable.sol#380-388)
ERC1155Upgradeable._doSafeBatchTransferAcceptanceCheck(address,address,address,uint256[],uint256[],bytes) (contracts/erc-1155/ERC1155Upgradeable.sol#392-413) ignores return value by IERC1155ReceiverUpgradeable(to).onERC1155BatchReceived(
operator,from,ids,amounts,data) (contracts/erc-1155/ERC1155Upgradeable.sol#403-411)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

ERC1155Base.isApprovedForAll(address,address)._owner (contracts/erc-1155/ERC1155Base.sol#19) shadows:
        - OwnableUpgradeable._owner (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#20) (state variable)
ERC1155Base._burnLazy(uint256,uint256).supply (contracts/erc-1155/ERC1155Base.sol#56) shadows:
        - ERC1155Lazy.supply (contracts/erc-1155/ERC1155Lazy.sol#21) (state variable)
ERC1155BaseURI._tokenURI(uint256)._tokenURI (contracts/erc-1155/ERC1155BaseURI.sol#33) shadows:
        - ERC1155BaseURI._tokenURI(uint256) (contracts/erc-1155/ERC1155BaseURI.sol#32-46) (function)
ERC1155BaseURI._setTokenURI(uint256,string)._uri (contracts/erc-1155/ERC1155BaseURI.sol#55) shadows:
        - ERC1155Upgradeable._uri (contracts/erc-1155/ERC1155Upgradeable.sol#33) (state variable)
ERC1155Rarible.__ERC1155RaribleUser_init(string,string,string,string,address[],address,address).baseURI (contracts/erc-1155/ERC1155Rarible.sol#15) shadows:
        - ERC1155BaseURI.baseURI() (contracts/erc-1155/ERC1155BaseURI.sol#24-26) (function)
ERC1155Rarible.__ERC1155RaribleUser_init(string,string,string,string,address[],address,address).contractURI (contracts/erc-1155/ERC1155Rarible.sol#15) shadows:
        - HasContractURI.contractURI (contracts/HasContractURI.sol#9) (state variable)
ERC1155Rarible._ERC1155Rarible_init(string,string,string,string,address,address).baseURI (contracts/erc-1155/ERC1155Rarible.sol#25) shadows:
        - ERC1155BaseURI.baseURI() (contracts/erc-1155/ERC1155BaseURI.sol#24-26) (function)
ERC1155Rarible.__ERC1155Rarible_init(string,string,string,string,address,address).contractURI (contracts/erc-1155/ERC1155Rarible.sol#25) shadows:
        - HasContractURI.contractURI (contracts/HasContractURI.sol#9) (state variable)
ERC1155Rarible._ERC1155Rarible_init_unchained(string,string,string,string,address,address).baseURI (contracts/erc-1155/ERC1155Rarible.sol#32) shadows:
        - ERC1155BaseURI.baseURI() (contracts/erc-1155/ERC1155BaseURI.sol#24-26) (function)
ERC1155Rarible.__ERC1155Rarible_init_unchained(string,string,string,string,address,address).contractURI (contracts/erc-1155/ERC1155Rarible.sol#32) shadows:
        - HasContractURI.contractURI (contracts/HasContractURI.sol#9) (state variable)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

ERC1155RaribleFactoryC2.constructor(address,address,address)._beacon (contracts/create-2/ERC1155RaribleFactoryC2.sol#24) lacks a zero-check on :
                - beacon = _beacon (contracts/create-2/ERC1155RaribleFactoryC2.sol#25)
ERC1155RaribleFactoryC2.constructor(address,address,address)._transferProxy (contracts/create-2/ERC1155RaribleFactoryC2.sol#24) lacks a zero-check on :
                - transferProxy = _transferProxy (contracts/create-2/ERC1155RaribleFactoryC2.sol#26)
ERC1155RaribleFactoryC2.constructor(address,address,address)._lazyTransferProxy (contracts/create-2/ERC1155RaribleFactoryC2.sol#24) lacks a zero-check on :
                - lazyTransferProxy = _lazyTransferProxy (contracts/create-2/ERC1155RaribleFactoryC2.sol#27)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation

Variable 'ERC1155Upgradeable._doSafeTransferAcceptanceCheck(address,address,address,uint256,uint256,bytes).response (contracts/erc-1155/ERC1155Upgradeable.sol#380)' in ERC1155Upgradeable._doSafeTransferAcceptanceCheck(address,address,add
ress,uint256,uint256,bytes) (contracts/erc-1155/ERC1155Upgradeable.sol#369-390) potentially used before declaration: response != IERC1155ReceiverUpgradeable(to).onERC1155Received.selector (contracts/erc-1155/ERC1155Upgradeable.sol#381)
Variable 'ERC1155Upgradeable._doSafeTransferAcceptanceCheck(address,address,address,uint256,uint256,bytes).reason (contracts/erc-1155/ERC1155Upgradeable.sol#384)' in ERC1155Upgradeable._doSafeTransferAcceptanceCheck(address,address,addre
ss,uint256,uint256,bytes) (contracts/erc-1155/ERC1155Upgradeable.sol#369-390) potentially used before declaration: revert(string)(reason) (contracts/erc-1155/ERC1155Upgradeable.sol#385)
Variable 'ERC1155Upgradeable._doSafeBatchTransferAcceptanceCheck(address,address,address,uint256[],uint256[],bytes).response (contracts/erc-1155/ERC1155Upgradeable.sol#403)' in ERC1155Upgradeable._doSafeBatchTransferAcceptanceCheck(addre
ss,address,address,uint256[],uint256[],bytes) (contracts/erc-1155/ERC1155Upgradeable.sol#392-413) potentially used before declaration: response != IERC1155ReceiverUpgradeable(to).onERC1155BatchReceived.selector (contracts/erc-1155/ERC1155
Upgradeable.sol#404)
Variable 'ERC1155Upgradeable._doSafeBatchTransferAcceptanceCheck(address,address,address,uint256[],uint256[],bytes).reason (contracts/erc-1155/ERC1155Upgradeable.sol#407)' in ERC1155Upgradeable._doSafeBatchTransferAcceptanceCheck(address
,address,address,uint256[],uint256[],bytes) (contracts/erc-1155/ERC1155Upgradeable.sol#392-413) potentially used before declaration: revert(string)(reason) (contracts/erc-1155/ERC1155Upgradeable.sol#408)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables

Reentrancy in ERC1155RaribleFactoryC2.createToken(string,string,string,string,uint256) (contracts/create-2/ERC1155RaribleFactoryC2.sol#30-36):
        External calls:
        - token.transferOwnership(_msgSender()) (contracts/create-2/ERC1155RaribleFactoryC2.sol#34)
        Event emitted after the call(s):
        - Create1155RaribleProxy(beaconProxy) (contracts/create-2/ERC1155RaribleFactoryC2.sol#35)
Reentrancy in ERC1155RaribleFactoryC2.createToken(string,string,string,string,address[],uint256) (contracts/create-2/ERC1155RaribleFactoryC2.sol#38-46):
        External calls:
        - token.transferOwnership(_msgSender()) (contracts/create-2/ERC1155RaribleFactoryC2.sol#42)
        Event emitted after the call(s):
        - Create1155RaribleUserProxy(beaconProxy) (contracts/create-2/ERC1155RaribleFactoryC2.sol#43)
Reentrancy in ERC1155Lazy.mintAndTransfer(LibERC1155LazyMint.Mint1155Data,address,uint256) (contracts/erc-1155/ERC1155Lazy.sol#57-90):
        External calls:
        - _mint(to,data.tokenId,_amount,) (contracts/erc-1155/ERC1155Lazy.sol#83)
                - IERC1155ReceiverUpgradeable(to).onERC1155Received(operator,from,id,amount,data) (contracts/erc-1155/ERC1155Upgradeable.sol#380-388)
        Event emitted after the call(s):
        - TransferSingle(sender,address(0),minter,data.tokenId,_amount) (contracts/erc-1155/ERC1155Lazy.sol#85)
        - TransferSingle(sender,minter,to,data.tokenId,_amount) (contracts/erc-1155/ERC1155Lazy.sol#86)
        - TransferSingle(sender,address(0),to,data.tokenId,_amount) (contracts/erc-1155/ERC1155Lazy.sol#88)
Reentrancy in ERC1155Lazy.transferFromOrMint(LibERC1155LazyMint.Mint1155Data,address,address,uint256) (contracts/erc-1155/ERC1155Lazy.sol#36-55):
        External calls:
        - safeTransferFrom(from,to,data.tokenId,transfer,) (contracts/erc-1155/ERC1155Lazy.sol#49)
                - IERC1155ReceiverUpgradeable(to).onERC1155Received(operator,from,id,amount,data) (contracts/erc-1155/ERC1155Upgradeable.sol#380-388)
        - mintAndTransfer(data,to,left) (contracts/erc-1155/ERC1155Lazy.sol#53)
                - IERC1155ReceiverUpgradeable(to).onERC1155Received(operator,from,id,amount,data) (contracts/erc-1155/ERC1155Upgradeable.sol#380-388)
        Event emitted after the call(s):
        - Creators(tokenId,_creators) (contracts/erc-1155/ERC1155Lazy.sol#124)
                - mintAndTransfer(data,to,left) (contracts/erc-1155/ERC1155Lazy.sol#53)
        - RoyaltiesSet(id,_royalties) (node_modules/@rarible/royalties/contracts/impl/RoyaltiesV2Impl.sol#16)
                - mintAndTransfer(data,to,left) (contracts/erc-1155/ERC1155Lazy.sol#53)
        - Supply(tokenId,_supply) (contracts/erc-1155/ERC1155Lazy.sol#111)
                - mintAndTransfer(data,to,left) (contracts/erc-1155/ERC1155Lazy.sol#53)
        - TransferSingle(sender,address(0),minter,data.tokenId,_amount) (contracts/erc-1155/ERC1155Lazy.sol#85)
                - mintAndTransfer(data,to,left) (contracts/erc-1155/ERC1155Lazy.sol#53)
        - TransferSingle(sender,minter,to,data.tokenId,_amount) (contracts/erc-1155/ERC1155Lazy.sol#86)
                - mintAndTransfer(data,to,left) (contracts/erc-1155/ERC1155Lazy.sol#53)
        - TransferSingle(sender,address(0),to,data.tokenId,_amount) (contracts/erc-1155/ERC1155Lazy.sol#88)
                - mintAndTransfer(data,to,left) (contracts/erc-1155/ERC1155Lazy.sol#53)
        - URI(_tokenURI(tokenId),tokenId) (contracts/erc-1155/ERC1155BaseURI.sol#57)
                - mintAndTransfer(data,to,left) (contracts/erc-1155/ERC1155Lazy.sol#53)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

EIP712Upgradeable._getChainId() (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#93-99) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#96-98)
AddressUpgradeable.isContract(address) (node_modules/@openzeppelin/contracts-upgradeable/utils/Address Upgradeable.sol#26-35) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#33)
AddressUpgradeable._verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#147-164) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#156-159)
BeaconProxy._beacon() (node_modules/@openzeppelin/contracts/proxy/BeaconProxy.sol#43-49) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/proxy/BeaconProxy.sol#46-48)
BeaconProxy._setBeacon(address,bytes) (node_modules/@openzeppelin/contracts/proxy/BeaconProxy.sol#68-87) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/proxy/BeaconProxy.sol#80-82)
Proxy._delegate(address) (node_modules/@openzeppelin/contracts/proxy/Proxy.sol#21-41) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/proxy/Proxy.sol#23-40)
Address.isContract(address) (node_modules/@openzeppelin/contracts/utils/Address.sol#26-35) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/utils/Address.sol#33)
Address._verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#171-188) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/utils/Address.sol#180-183)
LibSignature.recover(bytes32,bytes) (node_modules/@rarible/libraries/contracts/LibSignature.sol#20-45) uses assembly
        - INLINE ASM (node_modules/@rarible/libraries/contracts/LibSignature.sol#38-42)
ERC1155RaribleFactoryC2.deployProxy(bytes,uint256) (contracts/create-2/ERC1155RaribleFactoryC2.sol#47-55) uses assembly
        - INLINE ASM (contracts/create-2/ERC1155RaribleFactoryC2.sol#49-54)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
```

```
Different versions of Solidity is used:
        - Version used: ['0.7.6', '>=0.4.24<0.8.0', '>=0.6.0<0.8.0', '>=0.6.2<0.8.0', '^0.7.0']
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/introspection/IERC165Upgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#3)
        - >=0.4.24<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/proxy/Initializable.sol#4)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/ERC1155ReceiverUpgradeable.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/ERC1155MetadataURIUpgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/ERC1155ReceiverUpgradeable.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/ERC1155Upgradeable.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/StringsUpgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/access/Ownable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/proxy/BeaconProxy.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/proxy/IBeacon.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/proxy/Proxy.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts/utils/Address.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@rarible/lazy-mint/contracts/erc-1155/IERC1155LazyMint.sol#3)
        - v2 (node_modules/@rarible/lazy-mint/contracts/erc-1155/IERC1155LazyMint.sol#4)
        - >=0.6.2<0.8.0 (node_modules/@rarible/lazy-mint/contracts/erc-1155/LibERC1155LazyMint.sol#3)
        - ^0.7.0 (node_modules/@rarible/libraries/contracts/LibSignature.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@rarible/royalties-upgradeable/contracts/RoyaltiesV2Upgradeable.sol#3)
        - v2 (node_modules/@rarible/royalties-upgradeable/contracts/RoyaltiesV2Upgradeable.sol#4)
        - >=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/LibPart.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/LibRoyaltiesV2.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/RoyaltiesV2.sol#3)
        - v2 (node_modules/@rarible/royalties/contracts/RoyaltiesV2.sol#4)
        - >=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/impl/AbstractRoyalties.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/impl/RoyaltiesV2Impl.sol#3)
        - v2 (node_modules/@rarible/royalties/contracts/impl/RoyaltiesV2Impl.sol#4)
        - 0.7.6 (contracts/HasContractURI.sol#3)
        - 0.7.6 (contracts/LibURI.sol#3)
        - >=0.6.2<0.8.0 (contracts/create-2/ERC1155RaribleFactoryC2.sol#3)
        - v2 (contracts/create-2/ERC1155RaribleFactoryC2.sol#4)
        - 0.7.6 (contracts/erc-1155/ERC1155Base.sol#3)
        - v2 (contracts/erc-1155/ERC1155Base.sol#4)
        - 0.7.6 (contracts/erc-1155/ERC1155BaseURI.sol#3)
        - >=0.6.0<0.8.0 (contracts/erc-1155/ERC1155BurnableUpgradeable.sol#3)
        - 0.7.6 (contracts/erc-1155/ERC1155DefaultApproval.sol#3)
        - 0.7.6 (contracts/erc-1155/ERC1155Lazy.sol#3)
        - v2 (contracts/erc-1155/ERC1155Lazy.sol#4)
        - 0.7.6 (contracts/erc-1155/ERC1155Rarible.sol#3)
        - v2 (contracts/erc-1155/ERC1155Rarible.sol#4)
        - >=0.6.0<0.8.0 (contracts/erc-1155/ERC1155Upgradeable.sol#3)
        - 0.7.6 (contracts/erc-1155/Mint1155Validator.sol#3)
        - 0.7.6 (contracts/erc-1271/ERC1271.sol#3)
        - 0.7.6 (contracts/erc-1271/ERC1271Validator.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

AbstractRoyalties._onRoyaltiesSet(uint256,LibPart.Part[]) (node_modules/@rarible/royalties/contracts/impl/AbstractRoyalties.sol#31) is never used and should be removed
Address.functionCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#79-81) is never used and should be removed
Address.functionCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#89-91) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#104-106) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#114-121) is never used and should be removed
Address.functionDelegateCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#153-155) is never used and should be removed
Address.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#129-131) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#139-145) is never used and should be removed
Address.sendValue(address,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#53-59) is never used and should be removed

AddressUpgradeable._verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#147-164) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#79-81) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#89-91) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#104-106) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#114-121) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#129-131) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#139-145) is never used and should be removed
AddressUpgradeable.sendValue(address,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#53-59) is never used and should be removed
Context._msgData() (node_modules/@openzeppelin/contracts/utils/Context.sol#20-23) is never used and should be removed
ContextUpgradeable.__Context_init() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#17-19) is never used and should be removed
ContextUpgradeable._msgData() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#27-30) is never used and should be removed
EIP712Upgradeable.__EIP712_init(string,string) (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#44-46) is never used and should be removed
ERC1155BurnableUpgradeable.__ERC1155Burnable_init() (contracts/erc-1155/ERC1155BurnableUpgradeable.sol#15-19) is never used and should be removed
ERC1155ReceiverUpgradeable.__ERC1155Receiver_init() (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/ERC1155ReceiverUpgradeable.sol#13-16) is never used and should be removed
ERC1155ReceiverUpgradeable.__ERC1155Receiver_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/ERC1155ReceiverUpgradeable.sol#18-23) is never used and should be removed
ERC1155Upgradeable.__ERC1155_init(string) (contracts/erc-1155/ERC1155Upgradeable.sol#56-60) is never used and should be removed
ERC1155Upgradeable._mint(address,uint256,uint256,bytes) (contracts/erc-1155/ERC1155Upgradeable.sol#250-261) is never used and should be removed
ERC1155Upgradeable._mintBatch(address,uint256[],uint256[],bytes) (contracts/erc-1155/ERC1155Upgradeable.sol#272-287) is never used and should be removed
ERC1271.returnIsValidSignatureMagicNumber(bool) (contracts/erc-1271/ERC1271.sol#22-24) is never used and should be removed
ERC165Upgradeable.__ERC165_init() (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#25-27) is never used and should be removed
HasContractURI._setContractURI(string) (contracts/HasContractURI.sol#25-27) is never used and should be removed
OwnableUpgradeable.__Ownable_init() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#27-30) is never used and should be removed
Proxy._implementation() (node_modules/@openzeppelin/contracts/proxy/Proxy.sol#47) is never used and should be removed
SafeMathUpgradeable.div(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#135-138) is never used and should be removed
SafeMathUpgradeable.div(uint256,uint256,string) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#190-193) is never used and should be removed
SafeMathUpgradeable.mod(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#152-155) is never used and should be removed
SafeMathUpgradeable.mod(uint256,uint256,string) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#210-213) is never used and should be removed
SafeMathUpgradeable.mul(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#116-121) is never used and should be removed
SafeMathUpgradeable.sub(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#101-104) is never used and should be removed
SafeMathUpgradeable.tryAdd(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#24-28) is never used and should be removed
SafeMathUpgradeable.tryDiv(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#60-63) is never used and should be removed
SafeMathUpgradeable.tryMod(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#70-73) is never used and should be removed
SafeMathUpgradeable.tryMul(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#45-53) is never used and should be removed
SafeMathUpgradeable.trySub(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#35-38) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/introspection/IERC165Upgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#3) is too complex
Pragma version>=0.4.24<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/proxy/Initializable.sol#4) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/ERC1155ReceiverUpgradeable.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/IERC1155MetadataURIUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/ERC1155ReceiverUpgradeable.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/IERC1155Upgradeable.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/StringsUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/access/Ownable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/proxy/BeaconProxy.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/proxy/IBeacon.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/proxy/Proxy.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts/utils/Address.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/lazy-mint/contracts/erc-1155/IERC1155LazyMint.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/lazy-mint/contracts/erc-1155/LibERC1155LazyMint.sol#3) is too complex
Pragma version^0.7.0 (node_modules/@rarible/libraries/contracts/LibSignature.sol#3) allows old versions
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties-upgradeable/contracts/RoyaltiesV2Upgradeable.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/LibPart.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/LibRoyaltiesV2.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/RoyaltiesV2.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/impl/AbstractRoyalties.sol#3) is too complex

Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/impl/RoyaltiesV2Impl.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (contracts/create-2/ERC1155RaribleFactoryC2.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (contracts/erc-1155/ERC1155BurnableUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (contracts/erc-1155/ERC1155Upgradeable.sol#3) is too complex
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

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

Function OwnableUpgradeable.__Ownable_init() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#27-30) is not in mixedCase
Function OwnableUpgradeable.__Ownable_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#32-36) is not in mixedCase
Variable OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#74) is not in mixedCase
Function EIP712Upgradeable.__EIP712_init(string,string) (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#44-46) is not in mixedCase
Function EIP712Upgradeable.__EIP712_init_unchained(string,string) (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#48-53) is not in mixedCase
Function EIP712Upgradeable.__EIP712NameHash() (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#107-109) is not in mixedCase
Function EIP712Upgradeable.__EIP712VersionHash() (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#117-119) is not in mixedCase
Variable EIP712Upgradeable._HASHED_NAME (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#27) is not in mixedCase
Variable EIP712Upgradeable._HASHED_VERSION (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#28) is not in mixedCase
Variable EIP712Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#120) is not in mixedCase
Function ERC165Upgradeable.__ERC165_init() (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#25-27) is not in mixedCase
Function ERC165Upgradeable.__ERC165_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#29-32) is not in mixedCase
Variable ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59) is not in mixedCase
Function ERC1155ReceiverUpgradeable.__ERC1155Receiver_init() (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/ERC1155ReceiverUpgradeable.sol#13-16) is not in mixedCase
Function ERC1155ReceiverUpgradeable.__ERC1155Receiver_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/ERC1155ReceiverUpgradeable.sol#18-23) is not in mixedCase
Variable ERC1155ReceiverUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/ERC1155ReceiverUpgradeable.sol#24) is not in mixedCase
Function ContextUpgradeable.__Context_init() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#17-19) is not in mixedCase
Function ContextUpgradeable.__Context_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#21-22) is not in mixedCase
Variable ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31) is not in mixedCase
Function RoyaltiesV2Upgradeable.__RoyaltiesV2Upgradeable_init_unchained() (node_modules/@rarible/royalties-upgradeable/contracts/RoyaltiesV2Upgradeable.sol#11-13) is not in mixedCase
Function HasContractURI.__HasContractURI_init_unchained(string) (contracts/HasContractURI.sol#14-19) is not in mixedCase
Parameter: HasContractURI.__HasContractURI_init_unchained(string)._contractURI (contracts/HasContractURI.sol#16) is not in mixedCase
```

```
Variable HasContractURI.__gap (contracts/HasContractURI.sol#29) is not in mixedCase
Parameter LibURI.checkPrefix(string,string). tokenURI (contracts/LibURI.sol#57) is not in mixedCase
Parameter ERC1155RaribleFactoryC2.createToken(string,string,string,string,uint256)._name (contracts/create-2/ERC1155RaribleFactoryC2.sol#30) is not in mixedCase
Parameter ERC1155RaribleFactoryC2.createToken(string,string,string,string,uint256)._symbol (contracts/create-2/ERC1155RaribleFactoryC2.sol#30) is not in mixedCase
Parameter ERC1155RaribleFactoryC2.createToken(string,string,string,string,address[],uint256)._name (contracts/create-2/ERC1155RaribleFactoryC2.sol#38) is not in mixedCase
Parameter ERC1155RaribleFactoryC2.createToken(string,string,string,string,address[],uint256)._symbol (contracts/create-2/ERC1155RaribleFactoryC2.sol#38) is not in mixedCase
Parameter ERC1155RaribleFactoryC2.getCreationBytecode(bytes)._data (contracts/create-2/ERC1155RaribleFactoryC2.sol#58) is not in mixedCase
Parameter ERC1155RaribleFactoryC2.getAddress(string,string,string,string,uint256)._name (contracts/create-2/ERC1155RaribleFactoryC2.sol#63) is not in mixedCase
Parameter ERC1155RaribleFactoryC2.getAddress(string,string,string,string,uint256)._symbol (contracts/create-2/ERC1155RaribleFactoryC2.sol#63) is not in mixedCase
Parameter ERC1155RaribleFactoryC2.getAddress(string,string,string,string,uint256)._salt (contracts/create-2/ERC1155RaribleFactoryC2.sol#63) is not in mixedCase
Parameter ERC1155RaribleFactoryC2.getData(string,string,string,string)._name (contracts/create-2/ERC1155RaribleFactoryC2.sol#77) is not in mixedCase
Parameter ERC1155RaribleFactoryC2.getData(string,string,string,string)._symbol (contracts/create-2/ERC1155RaribleFactoryC2.sol#77) is not in mixedCase
Parameter ERC1155RaribleFactoryC2.getAddress(string,string,string,string,address[],uint256)._name (contracts/create-2/ERC1155RaribleFactoryC2.sol#82) is not in mixedCase
Parameter ERC1155RaribleFactoryC2.getAddress(string,string,string,string,address[],uint256)._symbol (contracts/create-2/ERC1155RaribleFactoryC2.sol#82) is not in mixedCase
Parameter ERC1155RaribleFactoryC2.getAddress(string,string,string,string,address[],uint256)._salt (contracts/create-2/ERC1155RaribleFactoryC2.sol#82) is not in mixedCase
Parameter ERC1155RaribleFactoryC2.getData(string,string,string,string,address[])._name (contracts/create-2/ERC1155RaribleFactoryC2.sol#96) is not in mixedCase
Parameter ERC1155RaribleFactoryC2.getData(string,string,string,string,address[])._symbol (contracts/create-2/ERC1155RaribleFactoryC2.sol#96) is not in mixedCase
Parameter ERC1155Base.isApprovedForAll(address,address)._owner (contracts/erc-1155/ERC1155Base.sol#19) is not in mixedCase
Parameter ERC1155Base.isApprovedForAll(address,address)._operator (contracts/erc-1155/ERC1155Base.sol#19) is not in mixedCase
Function ERC1155Base.__ERC1155Base_init_unchained(string,string) (contracts/erc-1155/ERC1155Base.sol#73-76) is not in mixedCase
Parameter ERC1155Base.__ERC1155Base_init_unchained(string,string)._name (contracts/erc-1155/ERC1155Base.sol#73) is not in mixedCase
Parameter ERC1155Base.__ERC1155Base_init_unchained(string,string)._symbol (contracts/erc-1155/ERC1155Base.sol#73) is not in mixedCase
Variable ERC1155Base.__gap (contracts/erc-1155/ERC1155Base.sol#82) is not in mixedCase
Variable ERC1155BaseURI.__gap (contracts/erc-1155/ERC1155BaseURI.sol#69) is not in mixedCase
Function ERC1155BurnableUpgradeable.__ERC1155Burnable_init() (contracts/erc-1155/ERC1155BurnableUpgradeable.sol#15-19) is not in mixedCase
Function ERC1155BurnableUpgradeable.__ERC1155Burnable_init_unchained() (contracts/erc-1155/ERC1155BurnableUpgradeable.sol#21-22) is not in mixedCase
Variable ERC1155BurnableUpgradeable.__gap (contracts/erc-1155/ERC1155BurnableUpgradeable.sol#40) is not in mixedCase
Parameter ERC1155DefaultApproval.isApprovedForAll(address,address)._owner (contracts/erc-1155/ERC1155DefaultApproval.sol#17) is not in mixedCase
Parameter ERC1155DefaultApproval.isApprovedForAll(address,address)._operator (contracts/erc-1155/ERC1155DefaultApproval.sol#17) is not in mixedCase
Variable ERC1155DefaultApproval.__gap (contracts/erc-1155/ERC1155DefaultApproval.sol#20) is not in mixedCase
Function ERC1155Lazy.__ERC1155Lazy_init_unchained() (contracts/erc-1155/ERC1155Lazy.sol#24-26) is not in mixedCase
Parameter ERC1155Lazy.mintAndTransfer(LibERC1155LazyMint.Mint1155Data,address,uint256)._amount (contracts/erc-1155/ERC1155Lazy.sol#57) is not in mixedCase
Parameter ERC1155Lazy.updateAccount(uint256,address,address)._id (contracts/erc-1155/ERC1155Lazy.sol#127) is not in mixedCase
Parameter ERC1155Lazy.updateAccount(uint256,address,address)._from (contracts/erc-1155/ERC1155Lazy.sol#127) is not in mixedCase
Parameter ERC1155Lazy.updateAccount(uint256,address,address)._to (contracts/erc-1155/ERC1155Lazy.sol#127) is not in mixedCase
Parameter ERC1155Lazy.getCreators(uint256)._id (contracts/erc-1155/ERC1155Lazy.sol#132) is not in mixedCase
Variable ERC1155Lazy.__gap (contracts/erc-1155/ERC1155Lazy.sol#148) is not in mixedCase
Function ERC1155Rarible.__ERC1155RaribleUser_init(string,string,string,string,address[],address,address) (contracts/erc-1155/ERC1155Rarible.sol#15-23) is not in mixedCase
Parameter ERC1155Rarible.__ERC1155RaribleUser_init(string,string,string,string,address[],address,address)._name (contracts/erc-1155/ERC1155Rarible.sol#15) is not in mixedCase
Parameter ERC1155Rarible.__ERC1155RaribleUser_init(string,string,string,string,address[],address,address)._symbol (contracts/erc-1155/ERC1155Rarible.sol#15) is not in mixedCase
Function ERC1155Rarible.__ERC1155Rarible_init(string,string,string,string,address,address) (contracts/erc-1155/ERC1155Rarible.sol#25-30) is not in mixedCase
Parameter ERC1155Rarible.__ERC1155Rarible_init(string,string,string,string,address,address)._name (contracts/erc-1155/ERC1155Rarible.sol#25) is not in mixedCase
Parameter ERC1155Rarible.__ERC1155Rarible_init(string,string,string,string,address,address)._symbol (contracts/erc-1155/ERC1155Rarible.sol#25) is not in mixedCase
Function ERC1155Rarible.__ERC1155Rarible_init_unchained(string,string,string,address,address) (contracts/erc-1155/ERC1155Rarible.sol#32-48) is not in mixedCase
Parameter ERC1155Rarible.__ERC1155Rarible_init_unchained(string,string,string,address,address)._name (contracts/erc-1155/ERC1155Rarible.sol#32) is not in mixedCase
Parameter ERC1155Rarible.__ERC1155Rarible_init_unchained(string,string,string,address,address)._symbol (contracts/erc-1155/ERC1155Rarible.sol#32) is not in mixedCase
Parameter ERC1155Rarible.mintAndTransfer(LibERC1155LazyMint.Mint1155Data,address,uint256)._amount (contracts/erc-1155/ERC1155Rarible.sol#50) is not in mixedCase
Variable ERC1155Rarible.__gap (contracts/erc-1155/ERC1155Rarible.sol#57) is not in mixedCase
Function ERC1155Upgradeable.__ERC1155_init(string) (contracts/erc-1155/ERC1155Upgradeable.sol#56-60) is not in mixedCase
Function ERC1155Upgradeable.__ERC1155_init_unchained(string) (contracts/erc-1155/ERC1155Upgradeable.sol#62-70) is not in mixedCase
Variable ERC1155Upgradeable.__gap (contracts/erc-1155/ERC1155Upgradeable.sol#421) is not in mixedCase
Variable ERC1155Upgradeable._balances (contracts/erc-1155/ERC1155Upgradeable.sol#27) is not in mixedCase
Function Mint1155Validator.__Mint1155Validator_init_unchained() (contracts/erc-1155/Mint1155Validator.sol#9-11) is not in mixedCase
Variable Mint1155Validator.__gap (contracts/erc-1155/Mint1155Validator.sol#16) is not in mixedCase
Variable ERC1271Validator.__gap (contracts/erc-1271/ERC1271Validator.sol#34) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#94)" inEIP712Upgradeable (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#25-121)
Redundant expression "this (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#20)" inContextUpgradeable (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#16-32)
Redundant expression "this (node_modules/@openzeppelin/contracts/utils/Context.sol#21)" inContext (node_modules/@openzeppelin/contracts/utils/Context.sol#15-24)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

ERC1155RaribleFactoryC2.getCreationBytecode(bytes) (contracts/create-2/ERC1155RaribleFactoryC2.sol#58-60) uses literals with too many digits:
        - abi.encodePacked(type()(BeaconProxy).creationCode,abi.encode(beacon,_data)) (contracts/create-2/ERC1155RaribleFactoryC2.sol#59)
ERC1271.slitherConstructorConstantVariables() (contracts/erc-1271/ERC1271.sol#5-26) uses literals with too many digits:
        - ERC1271_RETURN_INVALID_SIGNATURE = 0x00000000 (contracts/erc-1271/ERC1271.sol#9)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits

ERC1155ReceiverUpgradeable (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/ERC1155ReceiverUpgradeable.sol#12-25) does not implement functions:
        - IERC1155ReceiverUpgradeable.onERC1155BatchReceived(address,address,uint256[],uint256[],bytes) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/IERC1155ReceiverUpgradeable.sol#48-56)
        - IERC1155ReceiverUpgradeable.onERC1155Received(address,address,uint256,uint256,bytes) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/IERC1155ReceiverUpgradeable.sol#25-33)
ERC1271 (contracts/erc-1271/ERC1271.sol#5-26) does not implement functions:
        - ERC1271.isValidSignature(bytes32,bytes) (contracts/erc-1271/ERC1271.sol#20)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unimplemented-functions

ERC1155ReceiverUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/ERC1155ReceiverUpgradeable.sol#24) is never used in ERC1155ReceiverUpgradeable (node_modules/@openzeppelin/contracts-upgradeable/token/ERC11
55/ERC1155ReceiverUpgradeable.sol#12-25)
LibERC1155LazyMint._INTERFACE_ID_MINT_AND_TRANSFER (node_modules/@rarible/lazy-mint/contracts/erc-1155/LibERC1155LazyMint.sol#9) is never used in LibERC1155LazyMint (node_modules/@rarible/lazy-mint/contracts/erc-1155/LibERC1155LazyMint.s
ol#7-40)
ERC1155Rarible.__gap (contracts/erc-1155/ERC1155Rarible.sol#57) is never used in ERC1155Rarible (contracts/erc-1155/ERC1155Rarible.sol#5-58)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

renounceOwnership() should be declared external:
        - OwnableUpgradeable.renounceOwnership() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#60-63)
transferOwnership(address) should be declared external:
        - OwnableUpgradeable.transferOwnership(address) (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#69-73)
renounceOwnership() should be declared external:
        - Ownable.renounceOwnership() (node_modules/@openzeppelin/contracts/access/Ownable.sol#54-57)
transferOwnership(address) should be declared external:
        - Ownable.transferOwnership(address) (node_modules/@openzeppelin/contracts/access/Ownable.sol#63-67)
getAddress(string,string,string,string,uint256) should be declared external:
        - ERC1155RaribleFactoryC2.getAddress(string,string,string,string,uint256) (contracts/create-2/ERC1155RaribleFactoryC2.sol#63-75)
getAddress(string,string,string,string,address[],uint256) should be declared external:
        - ERC1155RaribleFactoryC2.getAddress(string,string,string,string,address[],uint256) (contracts/create-2/ERC1155RaribleFactoryC2.sol#82-94)
balanceOfBatch(address[],uint256[]) should be declared external:
        - ERC1155Upgradeable.balanceOfBatch(address[],uint256[]) (contracts/erc-1155/ERC1155Upgradeable.sol#105-124)
safeBatchTransferFrom(address,address,uint256[],uint256[],bytes) should be declared external:
        - ERC1155Upgradeable.safeBatchTransferFrom(address,address,uint256[],uint256[],bytes) (contracts/erc-1155/ERC1155Upgradeable.sol#178-214)
isValidSignature(bytes32,bytes) should be declared external:
        - ERC1271.isValidSignature(bytes32,bytes) (contracts/erc-1271/ERC1271.sol#20)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

ERC1155WoonklyNFTRevealWaveFactoryC2.sol

```
OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#74) shadows:
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#51)
ERC1155ReceiverUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/ERC1155ReceiverUpgradeable.sol#24) shadows:
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
HasContractURI.__gap (contracts/HasContractURI.sol#29) shadows:
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
ERC1155Base.__gap (contracts/erc-1155/ERC1155Base.sol#82) shadows:
        - HasContractURI.__gap (contracts/HasContractURI.sol#29)
        - ERC1155Lazy.__gap (contracts/erc-1155/ERC1155Lazy.sol#148)
        - Mint1155Validator.__gap (contracts/erc-1155/Mint1155Validator.sol#16)
        - ERC1271Validator.__gap (contracts/erc-1271/ERC1271Validator.sol#34)
        - EIP712Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#120)
        - ERC1155BaseURI.__gap (contracts/erc-1155/ERC1155BaseURI.sol#69)
        - ERC1155BurnableUpgradeable.__gap (contracts/erc-1155/ERC1155BurnableUpgradeable.sol#40)
        - ERC1155DefaultApproval.__gap (contracts/erc-1155/ERC1155DefaultApproval.sol#20)
        - ERC1155Upgradeable.__gap (contracts/erc-1155/ERC1155Upgradeable.sol#421)
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#74)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
ERC1155BaseURI.__gap (contracts/erc-1155/ERC1155BaseURI.sol#69) shadows:
        - ERC1155Upgradeable.__gap (contracts/erc-1155/ERC1155Upgradeable.sol#421)
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
ERC1155BurnableUpgradeable.__gap (contracts/erc-1155/ERC1155BurnableUpgradeable.sol#40) shadows:
        - ERC1155Upgradeable.__gap (contracts/erc-1155/ERC1155Upgradeable.sol#421)
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
ERC1155DefaultApproval.__gap (contracts/erc-1155/ERC1155DefaultApproval.sol#20) shadows:
        - ERC1155Upgradeable.__gap (contracts/erc-1155/ERC1155Upgradeable.sol#421)
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
ERC1155Lazy.__gap (contracts/erc-1155/ERC1155Lazy.sol#148) shadows:
        - Mint1155Validator.__gap (contracts/erc-1155/Mint1155Validator.sol#16)
        - ERC1271Validator.__gap (contracts/erc-1271/ERC1271Validator.sol#34)
        - EIP712Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#120)
        - ERC1155BaseURI.__gap (contracts/erc-1155/ERC1155BaseURI.sol#69)
        - ERC1155Upgradeable.__gap (contracts/erc-1155/ERC1155Upgradeable.sol#421)
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
ERC1155Lazy._INTERFACE_ID_ERC165 (contracts/erc-1155/ERC1155Lazy.sol#16) shadows:
        - ERC165Upgradeable._INTERFACE_ID_ERC165 (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#18)
ERC1155Lazy._INTERFACE_ID_ERC1155 (contracts/erc-1155/ERC1155Lazy.sol#17) shadows:
        - ERC1155Upgradeable._INTERFACE_ID_ERC1155 (contracts/erc-1155/ERC1155Upgradeable.sol#66)
ERC1155Lazy._INTERFACE_ID_ERC1155_METADATA_URI (contracts/erc-1155/ERC1155Lazy.sol#18) shadows:
        - ERC1155Upgradeable._INTERFACE_ID_ERC1155_METADATA_URI (contracts/erc-1155/ERC1155Upgradeable.sol#51)
ERC1155RaribleRevealWave.__gap (contracts/erc-1155/ERC1155RaribleRevealWave.sol#196) shadows:
        - ERC1155Base.__gap (contracts/erc-1155/ERC1155Base.sol#82)
        - HasContractURI.__gap (contracts/HasContractURI.sol#29)
        - ERC1155Lazy.__gap (contracts/erc-1155/ERC1155Lazy.sol#148)
        - Mint1155Validator.__gap (contracts/erc-1155/Mint1155Validator.sol#16)
        - ERC1271Validator.__gap (contracts/erc-1271/ERC1271Validator.sol#34)
        - EIP712Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#120)
        - ERC1155BaseURI.__gap (contracts/erc-1155/ERC1155BaseURI.sol#69)
        - ERC1155BurnableUpgradeable.__gap (contracts/erc-1155/ERC1155BurnableUpgradeable.sol#40)
        - ERC1155DefaultApproval.__gap (contracts/erc-1155/ERC1155DefaultApproval.sol#20)
        - ERC1155Upgradeable.__gap (contracts/erc-1155/ERC1155Upgradeable.sol#421)
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#74)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
ERC1155Upgradeable.__gap (contracts/erc-1155/ERC1155Upgradeable.sol#421) shadows:
        - ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59)
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
Mint1155Validator.__gap (contracts/erc-1155/Mint1155Validator.sol#16) shadows:
```

```
        - ERC1271Validator.__gap (contracts/erc-1271/ERC1271Validator.sol#34)
        - EIP712Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#120)
ERC1271Validator.__gap (contracts/erc-1271/ERC1271Validator.sol#34) shadows:
        - EIP712Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#120)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variable-shadowing

Reentrancy in ERC1155Lazy.transferFromOrMint(LibERC1155LazyMint.Mint1155Data,address,address,uint256) (contracts/erc-1155/ERC1155Lazy.sol#36-55):
        External calls:
        - safeTransferFrom(from,to,data.tokenId,transfer,) (contracts/erc-1155/ERC1155Lazy.sol#49)
                - IERC1155ReceiverUpgradeable(to).onERC1155Received(operator,from,id,amount,data) (contracts/erc-1155/ERC1155Upgradeable.sol#380-388)
        - mintAndTransfer(data,to,left) (contracts/erc-1155/ERC1155Lazy.sol#53)
                - IERC1155ReceiverUpgradeable(to).onERC1155Received(operator,from,id,amount,data) (contracts/erc-1155/ERC1155Upgradeable.sol#380-388)
        State variables written after the call(s):
        - mintAndTransfer(data,to,left) (contracts/erc-1155/ERC1155Lazy.sol#53)
                - _balances[id][account] += _balances[id][account].add(amount) (contracts/erc-1155/ERC1155Lazy.sol#103)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-1

ERC1155Upgradeable._doSafeBatchTransferAcceptanceCheck(address,address,address,uint256[],uint256[],bytes) (contracts/erc-1155/ERC1155Upgradeable.sol#403) is a local variable never initialized
ERC1155Upgradeable._doSafeBatchTransferAcceptanceCheck(address,address,address,uint256[],uint256[],bytes).reason (contracts/erc-1155/ERC1155Upgradeable.sol#407) is a local variable never initialized
ERC1155Upgradeable._doSafeTransferAcceptanceCheck(address,address,address,uint256,uint256,bytes).response (contracts/erc-1155/ERC1155Upgradeable.sol#380) is a local variable never initialized
ERC1271Validator.validate1271(address,bytes32,bytes).signerFromSig (contracts/erc-1271/ERC1271Validator.sol#19) is a local variable never initialized
ERC1155Upgradeable._doSafeTransferAcceptanceCheck(address,address,address,uint256,uint256,bytes).reason (contracts/erc-1155/ERC1155Upgradeable.sol#384) is a local variable never initialized
ERC1155RaribleRevealWave._tokenURI(uint256).._tokenURI (contracts/erc-1155/ERC1155RaribleRevealWave.sol#153) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

BeaconProxy._setBeacon(address,bytes) (node_modules/@openzeppelin/contracts/proxy/BeaconProxy.sol#68-87) ignores return value by Address.functionDelegateCall(_implementation(),data,BeaconProxy: function call failed) (node_modules/@openze
ppelin/contracts/proxy/BeaconProxy.sol#85)
ERC1155Upgradeable._doSafeTransferAcceptanceCheck(address,address,address,uint256,uint256,bytes) (contracts/erc-1155/ERC1155Upgradeable.sol#369-390) ignores return value by IERC1155ReceiverUpgradeable(to).onERC1155Received(operator,from,
id,amount,data) (contracts/erc-1155/ERC1155Upgradeable.sol#380-388)
ERC1155Upgradeable._doSafeBatchTransferAcceptanceCheck(address,address,address,uint256[],uint256[],bytes) (contracts/erc-1155/ERC1155Upgradeable.sol#392-413) ignores return value by IERC1155ReceiverUpgradeable(to).onERC1155BatchReceived(
operator,from,ids,amounts,data) (contracts/erc-1155/ERC1155Upgradeable.sol#403-411)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

ERC1155Base.isApprovedForAll(address,address)._owner (contracts/erc-1155/ERC1155Base.sol#19) shadows:
        - OwnableUpgradeable._owner (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#20) (state variable)
ERC1155Base._burnLazy(uint256,uint256).supply (contracts/erc-1155/ERC1155Base.sol#56) shadows:
        - ERC1155Lazy.supply (contracts/erc-1155/ERC1155Lazy.sol#21) (state variable)
ERC1155BaseURI._tokenURI(uint256)._tokenURI (contracts/erc-1155/ERC1155BaseURI.sol#33) shadows:
        - ERC1155BaseURI._tokenURI(uint256) (contracts/erc-1155/ERC1155BaseURI.sol#32-46) (function)
ERC1155BaseURI._setTokenURI(uint256,string)._uri (contracts/erc-1155/ERC1155BaseURI.sol#55) shadows:
        - ERC1155Upgradeable._uri (contracts/erc-1155/ERC1155Upgradeable.sol#33) (state variable)
ERC1155RaribleRevealWave.__ERC1155RaribleUser_init(string,string,string,string,address[],address,address).baseURI (contracts/erc-1155/ERC1155RaribleRevealWave.sol#29) shadows:
        - ERC1155BaseURI.baseURI() (contracts/erc-1155/ERC1155BaseURI.sol#24-26) (function)
ERC1155RaribleRevealWave.__ERC1155RaribleUser_init(string,string,string,string,address[],address,address).contractURI (contracts/erc-1155/ERC1155RaribleRevealWave.sol#29) shadows:
        - HasContractURI.contractURI (contracts/HasContractURI.sol#9) (state variable)
ERC1155RaribleRevealWave.__ERC1155Rarible_init(string,string,string,string,address,address).baseURI (contracts/erc-1155/ERC1155RaribleRevealWave.sol#39) shadows:
        - ERC1155BaseURI.baseURI() (contracts/erc-1155/ERC1155BaseURI.sol#24-26) (function)
ERC1155RaribleRevealWave.__ERC1155Rarible_init(string,string,string,string,address,address).contractURI (contracts/erc-1155/ERC1155RaribleRevealWave.sol#39) shadows:
        - HasContractURI.contractURI (contracts/HasContractURI.sol#9) (state variable)
ERC1155RaribleRevealWave.__ERC1155Rarible_init_unchained(string,string,string,string,address,address).baseURI (contracts/erc-1155/ERC1155RaribleRevealWave.sol#46) shadows:
        - ERC1155BaseURI.baseURI() (contracts/erc-1155/ERC1155BaseURI.sol#24-26) (function)
ERC1155RaribleRevealWave.__ERC1155Rarible_init_unchained(string,string,string,string,address,address).contractURI (contracts/erc-1155/ERC1155RaribleRevealWave.sol#46) shadows:
        - HasContractURI.contractURI (contracts/HasContractURI.sol#9) (state variable)
ERC1155RaribleRevealWave.changeHiddenBaseURI(uint256,string)._baseURI (contracts/erc-1155/ERC1155RaribleRevealWave.sol#89) shadows:
        - ERC1155BaseURI._baseURI (contracts/erc-1155/ERC1155BaseURI.sol#17) (state variable)
ERC1155RaribleRevealWave._tokenURI(uint256).._tokenURI (contracts/erc-1155/ERC1155RaribleRevealWave.sol#153) shadows:
        - ERC1155RaribleRevealWave._tokenURI(uint256) (contracts/erc-1155/ERC1155RaribleRevealWave.sol#147-169) (function)
        - ERC1155BaseURI._tokenURI(uint256) (contracts/erc-1155/ERC1155BaseURI.sol#32-46) (function)
ERC1155RaribleRevealWave._tokenURI(uint256)._baseURI (contracts/erc-1155/ERC1155RaribleRevealWave.sol#154) shadows:
        - ERC1155BaseURI._baseURI (contracts/erc-1155/ERC1155BaseURI.sol#17) (state variable)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

ERC1155RaribleRevealWaveFactoryC2.constructor(address,address,address)._beacon (contracts/create-2/ERC1155RaribleRevealWaveFactoryC2.sol#24) lacks a zero-check on :
                - beacon = _beacon (contracts/create-2/ERC1155RaribleRevealWaveFactoryC2.sol#25)
ERC1155RaribleRevealWaveFactoryC2.constructor(address,address,address)._transferProxy (contracts/create-2/ERC1155RaribleRevealWaveFactoryC2.sol#24) lacks a zero-check on :
                - transferProxy = _transferProxy (contracts/create-2/ERC1155RaribleRevealWaveFactoryC2.sol#26)
ERC1155RaribleRevealWaveFactoryC2.constructor(address,address,address)._lazyTransferProxy (contracts/create-2/ERC1155RaribleRevealWaveFactoryC2.sol#24) lacks a zero-check on :
                - lazyTransferProxy = _lazyTransferProxy (contracts/create-2/ERC1155RaribleRevealWaveFactoryC2.sol#27)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation

Variable 'ERC1155Upgradeable._doSafeTransferAcceptanceCheck(address,address,address,uint256,uint256,bytes).response (contracts/erc-1155/ERC1155Upgradeable.sol#380)' in ERC1155Upgradeable._doSafeTransferAcceptanceCheck(address,address,add
ress,uint256,uint256,bytes) (contracts/erc-1155/ERC1155Upgradeable.sol#369-390) potentially used before declaration: response != IERC1155ReceiverUpgradeable(to).onERC1155Received.selector (contracts/erc-1155/ERC1155Upgradeable.sol#381)
Variable 'ERC1155Upgradeable._doSafeTransferAcceptanceCheck(address,address,address,uint256,uint256,bytes).reason (contracts/erc-1155/ERC1155Upgradeable.sol#384)' in ERC1155Upgradeable._doSafeTransferAcceptanceCheck(address,address,addre
ss,uint256,uint256,bytes) (contracts/erc-1155/ERC1155Upgradeable.sol#369-390) potentially used before declaration: revert(string)(reason) (contracts/erc-1155/ERC1155Upgradeable.sol#385)
Variable 'ERC1155Upgradeable._doSafeBatchTransferAcceptanceCheck(address,address,address,uint256[],uint256[],bytes).response (contracts/erc-1155/ERC1155Upgradeable.sol#403)' in ERC1155Upgradeable._doSafeBatchTransferAcceptanceCheck(addre
ss,address,address,uint256[],uint256[],bytes) (contracts/erc-1155/ERC1155Upgradeable.sol#392-413) potentially used before declaration: response != IERC1155ReceiverUpgradeable(to).onERC1155BatchReceived.selector (contracts/erc-1155/ERC1155
Upgradeable.sol#404)
Variable 'ERC1155Upgradeable._doSafeBatchTransferAcceptanceCheck(address,address,address,uint256[],uint256[],bytes).reason (contracts/erc-1155/ERC1155Upgradeable.sol#407)' in ERC1155Upgradeable._doSafeBatchTransferAcceptanceCheck(address
,address,address,uint256[],uint256[],bytes) (contracts/erc-1155/ERC1155Upgradeable.sol#392-413) potentially used before declaration: revert(string)(reason) (contracts/erc-1155/ERC1155Upgradeable.sol#408)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables

Reentrancy in ERC1155RaribleRevealWaveFactoryC2.createToken(string,string,string,string,uint256) (contracts/create-2/ERC1155RaribleRevealWaveFactoryC2.sol#30-36):
        External calls:
        - token.transferOwnership(_msgSender()) (contracts/create-2/ERC1155RaribleRevealWaveFactoryC2.sol#34)
        Event emitted after the call(s):
        - CreateERC1155RaribleProxy(beaconProxy) (contracts/create-2/ERC1155RaribleRevealWaveFactoryC2.sol#35)
Reentrancy in ERC1155RaribleRevealWaveFactoryC2.createToken(string,string,string,string,address[],uint256) (contracts/create-2/ERC1155RaribleRevealWaveFactoryC2.sol#38-44):
        External calls:
        - token.transferOwnership(_msgSender()) (contracts/create-2/ERC1155RaribleRevealWaveFactoryC2.sol#42)
        Event emitted after the call(s):
        - CreateERC1155RaribleUserProxy(beaconProxy) (contracts/create-2/ERC1155RaribleRevealWaveFactoryC2.sol#43)
Reentrancy in ERC1155Lazy.mintAndTransfer(LibERC1155LazyMint.Mint1155Data,address,uint256) (contracts/erc-1155/ERC1155Lazy.sol#57-90):
        External calls:
        - _mint(to,data.tokenId,_amount,) (contracts/erc-1155/ERC1155Lazy.sol#83)
                - IERC1155ReceiverUpgradeable(to).onERC1155Received(operator,from,id,amount,data) (contracts/erc-1155/ERC1155Upgradeable.sol#380-388)
        Event emitted after the call(s):
        - TransferSingle(sender,address(0),minter,data.tokenId,_amount) (contracts/erc-1155/ERC1155Lazy.sol#85)
        - TransferSingle(sender,minter,to,data.tokenId,_amount) (contracts/erc-1155/ERC1155Lazy.sol#86)
        - TransferSingle(sender,address(0),to,data.tokenId,_amount) (contracts/erc-1155/ERC1155Lazy.sol#88)
Reentrancy in ERC1155Lazy.transferFromOrMint(LibERC1155LazyMint.Mint1155Data,address,address,uint256) (contracts/erc-1155/ERC1155Lazy.sol#36-55):
        External calls:
        - safeTransferFrom(from,to,data.tokenId,transfer,) (contracts/erc-1155/ERC1155Lazy.sol#49)
                - IERC1155ReceiverUpgradeable(to).onERC1155Received(operator,from,id,amount,data) (contracts/erc-1155/ERC1155Upgradeable.sol#380-388)
        - mintAndTransfer(data,to,left) (contracts/erc-1155/ERC1155Lazy.sol#53)
                - IERC1155ReceiverUpgradeable(to).onERC1155Received(operator,from,id,amount,data) (contracts/erc-1155/ERC1155Upgradeable.sol#380-388)
        Event emitted after the call(s):
        - Creators(tokenId,_creators) (contracts/erc-1155/ERC1155Lazy.sol#124)
                - mintAndTransfer(data,to,left) (contracts/erc-1155/ERC1155Lazy.sol#53)
        - RoyaltiesSet(id,_royalties) (node_modules/@rarible/royalties/contracts/impl/RoyaltiesV2Impl.sol#16)
                - mintAndTransfer(data,to,left) (contracts/erc-1155/ERC1155Lazy.sol#53)
        - Supply(tokenId,_supply) (contracts/erc-1155/ERC1155Lazy.sol#111)
                - mintAndTransfer(data,to,left) (contracts/erc-1155/ERC1155Lazy.sol#53)
        - TransferSingle(sender,address(0),minter,data.tokenId,_amount) (contracts/erc-1155/ERC1155Lazy.sol#85)
                - mintAndTransfer(data,to,left) (contracts/erc-1155/ERC1155Lazy.sol#53)
        - TransferSingle(sender,minter,to,data.tokenId,_amount) (contracts/erc-1155/ERC1155Lazy.sol#86)
                - mintAndTransfer(data,to,left) (contracts/erc-1155/ERC1155Lazy.sol#53)
        - TransferSingle(sender,address(0),to,data.tokenId,_amount) (contracts/erc-1155/ERC1155Lazy.sol#88)
                - mintAndTransfer(data,to,lefc) (contracts/erc-1155/ERC1155Lazy.sol#53)
        - URI(_tokenURI(tokenId),tokenId) (contracts/erc-1155/ERC1155BaseURI.sol#57)
                - mintAndTransfer(data,to,left) (contracts/erc-1155/ERC1155Lazy.sol#53)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

EIP712Upgradeable._getChainId() (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#93-99) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#96-98)
AddressUpgradeable.isContract(address) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#26-35) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#33)
AddressUpgradeable._verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#147-164) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#156-159)
BeaconProxy._beacon() (node_modules/@openzeppelin/contracts/proxy/BeaconProxy.sol#43-49) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/proxy/BeaconProxy.sol#46-48)
BeaconProxy._setBeacon(address,bytes) (node_modules/@openzeppelin/contracts/proxy/BeaconProxy.sol#68-87) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/proxy/BeaconProxy.sol#80-82)
Proxy._delegate(address) (node_modules/@openzeppelin/contracts/proxy/Proxy.sol#21-41) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/proxy/Proxy.sol#23-40)
Address.isContract(address) (node_modules/@openzeppelin/contracts/utils/Address.sol#26-35) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/utils/Address.sol#33)
Address._verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#171-188) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/utils/Address.sol#180-183)
LibSignature.recover(bytes32,bytes) (node_modules/@rarible/libraries/contracts/LibSignature.sol#20-45) uses assembly
        - INLINE ASM (node_modules/@rarible/libraries/contracts/LibSignature.sol#38-42)
ERC1155RaribleRevealWaveFactoryC2.deployProxy(bytes,bytes256) (contracts/create-2/ERC1155RaribleRevealWaveFactoryC2.sol#47-55) uses assembly
        - INLINE ASM (contracts/create-2/ERC1155RaribleRevealWaveFactoryC2.sol#49-54)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

ERC1155RaribleRevealWave._tokenURI(uint256) (contracts/erc-1155/ERC1155RaribleRevealWave.sol#147-169) compares to a boolean constant:
        -revealWave.isRevealed == false (contracts/erc-1155/ERC1155RaribleRevealWave.sol#161)
ERC1155RaribleRevealWave._tokenURI(uint256) (contracts/erc-1155/ERC1155RaribleRevealWave.sol#147-169) compares to a boolean constant:
        -revealWave.isRevealed == true || revealWave.addTokenURIToHiddenBaseURI == true (contracts/erc-1155/ERC1155RaribleRevealWave.sol#156)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#boolean-equality

Different versions of Solidity is used:
        - Version used: ['0.7.6', '>=0.4.24<0.8.0', '>=0.6.0<0.8.0', '>=0.6.2<0.8.0', '^0.7.0']
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/introspection/IERC165Upgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#3)
        - >=0.4.24<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/proxy/Initializable.sol#4)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/ERC1155ReceiverUpgradeable.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/IERC1155MetadataURIUpgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/IERC1155ReceiverUpgradeable.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/IERC1155Upgradeable.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/StringsUpgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/access/Ownable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/proxy/BeaconProxy.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/proxy/IBeacon.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/proxy/Proxy.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts/utils/Address.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@rarible/lazy-mint/contracts/erc-1155/IERC1155LazyMint.sol#3)
```

```
            - v2 (node_modules/@rarible/lazy-mint/contracts/erc-1155/IERC1155LazyMint.sol#4)
            - >=0.6.2<0.8.0 (node_modules/@rarible/lazy-mint/contracts/erc-1155/LibERC1155LazyMint.sol#3)
            - ^0.7.0 (node_modules/@rarible/libraries/contracts/LibSignature.sol#3)
            - >=0.6.2<0.8.0 (node_modules/@rarible/royalties-upgradeable/contracts/RoyaltiesV2Upgradeable.sol#3)
            - v2 (node_modules/@rarible/royalties-upgradeable/contracts/RoyaltiesV2Upgradeable.sol#4)
            - >=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/Lib.sol#3)
            - >=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/LibRoyaltiesV2.sol#3)
            - >=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/RoyaltiesV2.sol#3)
            - v2 (node_modules/@rarible/royalties/contracts/RoyaltiesV2.sol#4)
            - >=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/impl/AbstractRoyalties.sol#3)
            - >=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/impl/RoyaltiesV2Impl.sol#3)
            - v2 (node_modules/@rarible/royalties/contracts/impl/RoyaltiesV2Impl.sol#4)
            - 0.7.6 (contracts/HasContractURI.sol#3)
            - 0.7.6 (contracts/LibURI.sol#3)
            - >=0.6.2<0.8.0 (contracts/create-2/ERC1155RaribleRevealWaveFactoryC2.sol#3)
            - v2 (contracts/create-2/ERC1155RaribleRevealWaveFactoryC2.sol#4)
            - 0.7.6 (contracts/erc-1155/ERC1155Base.sol#3)
            - v2 (contracts/erc-1155/ERC1155Base.sol#4)
            - 0.7.6 (contracts/erc-1155/ERC1155BaseURI.sol#3)
            - >=0.6.0<0.8.0 (contracts/erc-1155/ERC1155BurnableUpgradeable.sol#3)
            - 0.7.6 (contracts/erc-1155/ERC1155DefaultApproval.sol#3)
            - 0.7.6 (contracts/erc-1155/ERC1155Lazy.sol#3)
            - v2 (contracts/erc-1155/ERC1155Lazy.sol#4)
            - 0.7.6 (contracts/erc-1155/ERC1155RaribleRevealWave.sol#3)
            - v2 (contracts/erc-1155/ERC1155RaribleRevealWave.sol#4)
            - >=0.6.0<0.8.0 (contracts/erc-1155/ERC1155Upgradeable.sol#3)
            - 0.7.6 (contracts/erc-1155/HintI155Validator.sol#3)
            - 0.7.6 (contracts/erc-1271/ERC1271.sol#3)
            - 0.7.6 (contracts/erc-1271/ERC1271Validator.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

AbstractRoyalties._onRoyaltiesSet(uint256,LibPart.Part[]) (node_modules/@rarible/royalties/contracts/impl/AbstractRoyalties.sol#31) is never used and should be removed
Address.functionCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#79-81) is never used and should be removed
Address.functionCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#88-91) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#104-106) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#114-121) is never used and should be removed
Address.functionDelegateCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#153-155) is never used and should be removed
Address.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#129-131) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#139-145) is never used and should be removed
Address.sendValue(address,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#53-59) is never used and should be removed
AddressUpgradeable._verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#147-164) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#79-81) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#88-91) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#104-106) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#114-121) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#129-131) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#139-145) is never used and should be removed
AddressUpgradeable.sendValue(address,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#53-59) is never used and should be removed
Context._msgData() (node_modules/@openzeppelin/contracts/utils/Context.sol#20-23) is never used and should be removed
ContextUpgradeable.__Context_init() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#17-19) is never used and should be removed
ContextUpgradeable._msgData() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#27-30) is never used and should be removed
EIP712Upgradeable.__EIP712_init(string,string) (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#44-46) is never used and should be removed
ERC1155BurnableUpgradeable.__ERC1155Burnable_init() (contracts/erc-1155/ERC1155BurnableUpgradeable.sol#15-18) is never used and should be removed
ERC1155ReceiverUpgradeable.__ERC1155Receiver_init() (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/ERC1155ReceiverUpgradeable.sol#13-16) is never used and should be removed
ERC1155ReceiverUpgradeable.__ERC1155Receiver_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/ERC1155ReceiverUpgradeable.sol#18-23) is never used and should be removed
ERC1155Upgradeable.__ERC1155_init(string) (contracts/erc-1155/ERC1155Upgradeable.sol#56-60) is never used and should be removed
ERC1155Upgradeable._mint(address,uint256,uint256,bytes) (contracts/erc-1155/ERC1155Upgradeable.sol#250-261) is never used and should be removed
ERC1155Upgradeable._mintBatch(address,uint256[],uint256[],bytes) (contracts/erc-1155/ERC1155Upgradeable.sol#272-287) is never used and should be removed
ERC1271.returnIsValidSignatureMagicNumber(bool) (contracts/erc-1271/ERC1271.sol#22-24) is never used and should be removed
ERC165Upgradeable.__ERC165_init() (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#25-27) is never used and should be removed
HasContractURI._setContractURI(string) (contracts/HasContractURI.sol#25-27) is never used and should be removed
OwnableUpgradeable.__Ownable_init() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#27-30) is never used and should be removed
Proxy._implementation() (node_modules/@openzeppelin/contracts/proxy/Proxy.sol#47) is never used and should be removed
SafeMathUpgradeable.div(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#135-138) is never used and should be removed
SafeMathUpgradeable.div(uint256,uint256,string) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#190-193) is never used and should be removed
SafeMathUpgradeable.mod(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#152-155) is never used and should be removed
SafeMathUpgradeable.mod(uint256,uint256,string) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#210-213) is never used and should be removed
SafeMathUpgradeable.mul(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#116-121) is never used and should be removed
SafeMathUpgradeable.sub(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#101-104) is never used and should be removed
SafeMathUpgradeable.tryAdd(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#24-28) is never used and should be removed
SafeMathUpgradeable.tryDiv(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#60-63) is never used and should be removed
SafeMathUpgradeable.tryMod(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#70-73) is never used and should be removed
SafeMathUpgradeable.tryMul(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#45-53) is never used and should be removed
SafeMathUpgradeable.trySub(uint256,uint256) (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#35-38) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/introspection/IERC165Upgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/math/SafeMathUpgradeable.sol#3) is too complex
Pragma version>=0.4.24<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/proxy/Initializable.sol#4) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/ERC1155ReceiverUpgradeable.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/IERC1155MetadataURIUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/IERC1155ReceiverUpgradeable.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/IERC1155Upgradeable.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/StringsUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/access/Ownable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/proxy/BeaconProxy.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/proxy/IBeacon.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/proxy/Proxy.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts/utils/Address.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/lazy-mint/contracts/erc-1155/IERC1155LazyMint.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/lazy-mint/contracts/erc-1155/LibERC1155LazyMint.sol#3) is too complex
Pragma version^0.7.0 (node_modules/@rarible/libraries/contracts/LibSignature.sol#3) allows old versions
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties-upgradeable/contracts/RoyaltiesV2Upgradeable.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/LibPart.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/LibRoyaltiesV2.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/RoyaltiesV2.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/impl/AbstractRoyalties.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/impl/RoyaltiesV2Impl.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (contracts/create-2/ERC1155RaribleRevealWaveFactoryC2.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (contracts/erc-1155/ERC1155BurnableUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (contracts/erc-1155/ERC1155Upgradeable.sol#3) is too complex
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

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

Function OwnableUpgradeable.__Ownable_init() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#27-30) is not in mixedCase

Function OwnableUpgradeable.__Ownable_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#32-36) is not in mixedCase
Variable OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#74) is not in mixedCase
Function EIP712Upgradeable.__EIP712_init(string,string) (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#44-46) is not in mixedCase
Function EIP712Upgradeable.__EIP712_init_unchained(string,string) (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#48-53) is not in mixedCase
Function EIP712Upgradeable._EIP712NameHash() (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#107-109) is not in mixedCase
Function EIP712Upgradeable._EIP712VersionHash() (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#117-119) is not in mixedCase
Variable EIP712Upgradeable._HASHED_NAME (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#27) is not in mixedCase
Variable EIP712Upgradeable._HASHED_VERSION (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#28) is not in mixedCase
Variable EIP712Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#120) is not in mixedCase
Function ERC165Upgradeable.__ERC165_init() (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#25-27) is not in mixedCase
Function ERC165Upgradeable.__ERC165_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#29-33) is not in mixedCase
Variable ERC165Upgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/introspection/ERC165Upgradeable.sol#59) is not in mixedCase
Function ERC1155ReceiverUpgradeable.__ERC1155Receiver_init() (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/ERC1155ReceiverUpgradeable.sol#13-16) is not in mixedCase
Function ERC1155ReceiverUpgradeable.__ERC1155Receiver_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/ERC1155ReceiverUpgradeable.sol#18-23) is not in mixedCase
Variable ERC1155ReceiverUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/ERC1155ReceiverUpgradeable.sol#24) is not in mixedCase
Function ContextUpgradeable.__Context_init() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#17-19) is not in mixedCase
Function ContextUpgradeable.__Context_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#21-22) is not in mixedCase
Variable ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31) is not in mixedCase
Function RoyaltiesV2Upgradeable.__RoyaltiesV2Upgradeable_init_unchained() (node_modules/@rarible/royalties-upgradeable/contracts/RoyaltiesV2Upgradeable.sol#11-13) is not in mixedCase
Function HasContractURI.__HasContractURI_init_unchained(string) (contracts/HasContractURI.sol#16-19) is not in mixedCase
Parameter HasContractURI.__HasContractURI_init_unchained(string)._contractURI (contracts/HasContractURI.sol#16) is not in mixedCase
Variable HasContractURI.__gap (contracts/HasContractURI.sol#29) is not in mixedCase
Parameter LibURI.checkPrefix(string,string)._tokenURI (contracts/LibURI.sol#7) is not in mixedCase
Parameter ERC1155RaribleRevealWaveFactoryC2.createToken(string,string,string,string,uint256)._name (contracts/create-2/ERC1155RaribleRevealWaveFactoryC2.sol#30) is not in mixedCase
Parameter ERC1155RaribleRevealWaveFactoryC2.createToken(string,string,string,string,uint256)._symbol (contracts/create-2/ERC1155RaribleRevealWaveFactoryC2.sol#30) is not in mixedCase
Parameter ERC1155RaribleRevealWaveFactoryC2.createToken(string,string,string,string,address[],uint256)._name (contracts/create-2/ERC1155RaribleRevealWaveFactoryC2.sol#30) is not in mixedCase
Parameter ERC1155RaribleRevealWaveFactoryC2.createToken(string,string,string,string,address[],uint256)._symbol (contracts/create-2/ERC1155RaribleRevealWaveFactoryC2.sol#30) is not in mixedCase
Parameter ERC1155RaribleRevealWaveFactoryC2.getAddress(string,string,string,bytes)._data (contracts/create-2/ERC1155RaribleRevealWaveFactoryC2.sol#58) is not in mixedCase
Parameter ERC1155RaribleRevealWaveFactoryC2.getAddress(string,string,string,string,uint256)._name (contracts/create-2/ERC1155RaribleRevealWaveFactoryC2.sol#63) is not in mixedCase
Parameter ERC1155RaribleRevealWaveFactoryC2.getAddress(string,string,string,string,uint256)._symbol (contracts/create-2/ERC1155RaribleRevealWaveFactoryC2.sol#63) is not in mixedCase
Parameter ERC1155RaribleRevealWaveFactoryC2.getAddress(string,string,string,string,address[],uint256)._name (contracts/create-2/ERC1155RaribleRevealWaveFactoryC2.sol#63) is not in mixedCase
Parameter ERC1155RaribleRevealWaveFactoryC2.getData(string,string,string,string)._name (contracts/create-2/ERC1155RaribleRevealWaveFactoryC2.sol#77) is not in mixedCase
Parameter ERC1155RaribleRevealWaveFactoryC2.getData(string,string,string,string)._symbol (contracts/create-2/ERC1155RaribleRevealWaveFactoryC2.sol#77) is not in mixedCase
Parameter ERC1155RaribleRevealWaveFactoryC2.getAddress(string,string,string,string,address[])._name (contracts/create-2/ERC1155RaribleRevealWaveFactoryC2.sol#82) is not in mixedCase
Parameter ERC1155RaribleRevealWaveFactoryC2.getAddress(string,string,string,string,address[],uint256)._symbol (contracts/create-2/ERC1155RaribleRevealWaveFactoryC2.sol#82) is not in mixedCase
Parameter ERC1155RaribleRevealWaveFactoryC2.getAddress(string,string,string,string,address[],uint256)._salt (contracts/create-2/ERC1155RaribleRevealWaveFactoryC2.sol#82) is not in mixedCase
Parameter ERC1155RaribleRevealWaveFactoryC2.getData(string,string,string,string,address[])._name (contracts/create-2/ERC1155RaribleRevealWaveFactoryC2.sol#96) is not in mixedCase
Parameter ERC1155RaribleRevealWaveFactoryC2.getData(string,string,string,string,address[])._symbol (contracts/create-2/ERC1155RaribleRevealWaveFactoryC2.sol#96) is not in mixedCase
Parameter ERC1155Base.isApprovedForAll(address,address)._owner (contracts/erc-1155/ERC1155Base.sol#19) is not in mixedCase
Parameter ERC1155Base.isApprovedForAll(address,address)._operator (contracts/erc-1155/ERC1155Base.sol#19) is not in mixedCase
Function ERC1155Base.__ERC1155Base_init_unchained(string,string) (contracts/erc-1155/ERC1155Base.sol#73-76) is not in mixedCase
Parameter ERC1155Base.__ERC1155Base_init_unchained(string,string)._name (contracts/erc-1155/ERC1155Base.sol#73) is not in mixedCase
Parameter ERC1155Base.__ERC1155Base_init_unchained(string,string)._symbol (contracts/erc-1155/ERC1155Base.sol#73) is not in mixedCase
```

```
Variable ERC1155Base.__gap (contracts/erc-1155/ERC1155Base.sol#82) is not in mixedCase
Variable ERC1155BaseURI.__gap (contracts/erc-1155/ERC1155BaseURI.sol#68) is not in mixedCase
Function ERC1155BurnableUpgradeable.__ERC1155Burnable_init() (contracts/erc-1155/ERC1155BurnableUpgradeable.sol#15-19) is not in mixedCase
Function ERC1155BurnableUpgradeable.__ERC1155Burnable_init_unchained() (contracts/erc-1155/ERC1155BurnableUpgradeable.sol#21-22) is not in mixedCase
Variable ERC1155BurnableUpgradeable.__gap (contracts/erc-1155/ERC1155BurnableUpgradeable.sol#40) is not in mixedCase
Parameter ERC1155DefaultApproval.isApprovedForAll(address,address)._owner (contracts/erc-1155/ERC1155DefaultApproval.sol#17) is not in mixedCase
Parameter ERC1155DefaultApproval.isApprovedForAll(address,address)._operator (contracts/erc-1155/ERC1155DefaultApproval.sol#17) is not in mixedCase
Variable ERC1155DefaultApproval.__gap (contracts/erc-1155/ERC1155DefaultApproval.sol#20) is not in mixedCase
Function ERC1155Lazy.__ERC1155Lazy_init_unchained() (contracts/erc-1155/ERC1155Lazy.sol#24-26) is not in mixedCase
Parameter ERC1155Lazy.mintAndTransfer(LibERC1155LazyMint.Mint1155Data,address,uint256)._amount (contracts/erc-1155/ERC1155Lazy.sol#57) is not in mixedCase
Parameter ERC1155Lazy.updateAccount(uint256,address,address)._id (contracts/erc-1155/ERC1155Lazy.sol#127) is not in mixedCase
Parameter ERC1155Lazy.updateAccount(uint256,address,address)._from (contracts/erc-1155/ERC1155Lazy.sol#127) is not in mixedCase
Parameter ERC1155Lazy.updateAccount(uint256,address,address)._to (contracts/erc-1155/ERC1155Lazy.sol#127) is not in mixedCase
Parameter ERC1155Lazy.getCreators(uint256)._id (contracts/erc-1155/ERC1155Lazy.sol#132) is not in mixedCase
Variable ERC1155Lazy.__gap (contracts/erc-1155/ERC1155Lazy.sol#148) is not in mixedCase
Function ERC1155RaribleRevealWave.__ERC1155RaribleUser_init(string,string,string,string,address[],address,address) (contracts/erc-1155/ERC1155RaribleRevealWave.sol#29-37) is not in mixedCase
Parameter ERC1155RaribleRevealWave.__ERC1155RaribleUser_init(string,string,string,string,address[],address,address)._name (contracts/erc-1155/ERC1155RaribleRevealWave.sol#29) is not in mixedCase
Parameter ERC1155RaribleRevealWave.__ERC1155RaribleUser_init(string,string,string,string,address[],address,address)._symbol (contracts/erc-1155/ERC1155RaribleRevealWave.sol#29) is not in mixedCase
Function ERC1155RaribleRevealWave.__ERC1155Rarible_init(string,string,string,string,address,address) (contracts/erc-1155/ERC1155RaribleRevealWave.sol#39-44) is not in mixedCase
Parameter ERC1155RaribleRevealWave.__ERC1155Rarible_init(string,string,string,string,address,address)._name (contracts/erc-1155/ERC1155RaribleRevealWave.sol#39) is not in mixedCase
Parameter ERC1155RaribleRevealWave.__ERC1155Rarible_init(string,string,string,string,address,address)._symbol (contracts/erc-1155/ERC1155RaribleRevealWave.sol#39) is not in mixedCase
Function ERC1155RaribleRevealWave.__ERC1155Rarible_init_unchained(string,string,string,string,address,address) (contracts/erc-1155/ERC1155RaribleRevealWave.sol#46-62) is not in mixedCase
Parameter ERC1155RaribleRevealWave.__ERC1155Rarible_init_unchained(string,string,string,string,address,address)._name (contracts/erc-1155/ERC1155RaribleRevealWave.sol#46) is not in mixedCase
Parameter ERC1155RaribleRevealWave.__ERC1155Rarible_init_unchained(string,string,string,string,address,address)._symbol (contracts/erc-1155/ERC1155RaribleRevealWave.sol#46) is not in mixedCase
Parameter ERC1155RaribleRevealWave.getRevealWaveIdByTokenId(uint256)._tokenId (contracts/erc-1155/ERC1155RaribleRevealWave.sol#66) is not in mixedCase
Parameter ERC1155RaribleRevealWave.setRevealWave(uint256,string,string,bool,uint256)._waveId (contracts/erc-1155/ERC1155RaribleRevealWave.sol#73) is not in mixedCase
Parameter ERC1155RaribleRevealWave.setRevealWave(uint256,string,string,bool,uint256)._name (contracts/erc-1155/ERC1155RaribleRevealWave.sol#74) is not in mixedCase
Parameter ERC1155RaribleRevealWave.setRevealWave(uint256,string,string,bool,uint256)._hiddenBaseURI (contracts/erc-1155/ERC1155RaribleRevealWave.sol#75) is not in mixedCase
Parameter ERC1155RaribleRevealWave.setRevealWave(uint256,string,string,bool,uint256)._addTokenURIToHiddenBaseURI (contracts/erc-1155/ERC1155RaribleRevealWave.sol#76) is not in mixedCase
Parameter ERC1155RaribleRevealWave.setRevealWave(uint256,string,string,bool,uint256)._revealDate (contracts/erc-1155/ERC1155RaribleRevealWave.sol#77) is not in mixedCase
Parameter ERC1155RaribleRevealWave.changeHiddenBaseURI(uint256,string)._waveId (contracts/erc-1155/ERC1155RaribleRevealWave.sol#88) is not in mixedCase
Parameter ERC1155RaribleRevealWave.changeHiddenBaseURI(uint256,string)._baseURI (contracts/erc-1155/ERC1155RaribleRevealWave.sol#88) is not in mixedCase
Parameter ERC1155RaribleRevealWave.changeAddTokenURIToHiddenBaseURI(uint256,bool)._waveId (contracts/erc-1155/ERC1155RaribleRevealWave.sol#97) is not in mixedCase
Parameter ERC1155RaribleRevealWave.changeAddTokenURIToHiddenBaseURI(uint256,bool)._addTokenURIToHiddenBaseURI (contracts/erc-1155/ERC1155RaribleRevealWave.sol#98) is not in mixedCase
Parameter ERC1155RaribleRevealWave.changeRevealBaseURI(uint256,string)._waveId (contracts/erc-1155/ERC1155RaribleRevealWave.sol#105) is not in mixedCase
Parameter ERC1155RaribleRevealWave.changeRevealBaseURI(uint256,string)._revealBaseURI (contracts/erc-1155/ERC1155RaribleRevealWave.sol#106) is not in mixedCase
Parameter ERC1155RaribleRevealWave.changeRevealDate(uint256,uint256)._waveId (contracts/erc-1155/ERC1155RaribleRevealWave.sol#116) is not in mixedCase
Parameter ERC1155RaribleRevealWave.changeRevealDate(uint256,uint256)._revealDate (contracts/erc-1155/ERC1155RaribleRevealWave.sol#117) is not in mixedCase
Parameter ERC1155RaribleRevealWave.resetIsRevealedAndRevealURI(uint256)._waveId (contracts/erc-1155/ERC1155RaribleRevealWave.sol#124) is not in mixedCase
Parameter ERC1155RaribleRevealWave.changeName(uint256,string)._waveId (contracts/erc-1155/ERC1155RaribleRevealWave.sol#133) is not in mixedCase
Parameter ERC1155RaribleRevealWave.changeName(uint256,string)._name (contracts/erc-1155/ERC1155RaribleRevealWave.sol#134) is not in mixedCase
Parameter ERC1155RaribleRevealWave.assignRevealWaveIdToTokenId(uint256,uint256)._tokenId (contracts/erc-1155/ERC1155RaribleRevealWave.sol#142) is not in mixedCase
Parameter ERC1155RaribleRevealWave.assignRevealWaveIdToTokenId(uint256,uint256)._waveId (contracts/erc-1155/ERC1155RaribleRevealWave.sol#142) is not in mixedCase
Parameter ERC1155RaribleRevealWave.mintAndTransfer(LibERC1155LazyMint.Mint1155Data,address,uint256)._amount (contracts/erc-1155/ERC1155RaribleRevealWave.sol#175) is not in mixedCase
Variable ERC1155RaribleRevealWave.__gap (contracts/erc-1155/ERC1155RaribleRevealWave.sol#196) is not in mixedCase
Function ERC1155Upgradeable.__ERC1155_init(string) (contracts/erc-1155/ERC1155Upgradeable.sol#56-60) is not in mixedCase
Function ERC1155Upgradeable.__ERC1155_init_unchained(string) (contracts/erc-1155/ERC1155Upgradeable.sol#62-70) is not in mixedCase
Variable ERC1155Upgradeable.__gap (contracts/erc-1155/ERC1155Upgradeable.sol#421) is not in mixedCase
Variable ERC1155Upgradeable._balances (contracts/erc-1155/ERC1155Upgradeable.sol#27) is not in mixedCase
Function Mint1155Validator.__Mint1155Validator_init_unchained() (contracts/erc-1155/Mint1155Validator.sol#9-11) is not in mixedCase
Variable Mint1155Validator.__gap (contracts/erc-1155/Mint1155Validator.sol#16) is not in mixedCase
Variable ERC1271Validator.__gap (contracts/erc-1271/ERC1271Validator.sol#34) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#94)" inEIP712Upgradeable (node_modules/@openzeppelin/contracts-upgradeable/drafts/EIP712Upgradeable.sol#25-121)
Redundant expression "this (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#20)" inContextUpgradeable (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#16-32)
Redundant expression "this (node_modules/@openzeppelin/contracts/utils/Context.sol#21)" inContext (node_modules/@openzeppelin/contracts/utils/Context.sol#15-24)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

ERC1155RaribleRevealWaveFactoryC2.getCreationBytecode(bytes) (contracts/create-2/ERC1155RaribleRevealWaveFactoryC2.sol#50-60) uses literals with too many digits:
        - abi.encodePacked(type()(BeaconProxy).creationCode,abi.encode(beacon,_data)) (contracts/create-2/ERC1155RaribleRevealWaveFactoryC2.sol#59)
ERC1271.slitherConstructorConstantVariables() (contracts/erc-1271/ERC1271.sol#5-26) uses literals with too many digits:
        - ERC1271.RETURN_INVALID_SIGNATURE = 0x00000000 (contracts/erc-1271/ERC1271.sol#9)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits

ERC1155ReceiverUpgradeable (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/ERC1155ReceiverUpgradeable.sol#12-25) does not implement functions:
        - IERC1155ReceiverUpgradeable.onERC1155BatchReceived(address,address,uint256[],uint256[],bytes) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/IERC1155ReceiverUpgradeable.sol#48-56)
        - IERC1155ReceiverUpgradeable.onERC1155Received(address,address,uint256,uint256,bytes) (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/IERC1155ReceiverUpgradeable.sol#25-33)
ERC1271 (contracts/erc-1271/ERC1271.sol#5-26) does not implement functions:
        - ERC1271.isValidSignature(bytes32,bytes) (contracts/erc-1271/ERC1271.sol#20)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unimplemented-functions

ERC1155ReceiverUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/ERC1155ReceiverUpgradeable.sol#26) is never used in ERC1155ReceiverUpgradeable (node_modules/@openzeppelin/contracts-upgradeable/token/ERC11
55/ERC1155ReceiverUpgradeable.sol#12-25)
LibERC1155LazyMint._INTERFACE_ID_MINT_AND_TRANSFER (node_modules/@rarible/lazy-mint/contracts/erc-1155/LibERC1155LazyMint.sol#9) is never used in LibERC1155LazyMint (node_modules/@rarible/lazy-mint/contracts/erc-1155/LibERC1155LazyMint.s
ol#7-40)
ERC1155RaribleRevealWave.__gap (contracts/erc-1155/ERC1155RaribleRevealWave.sol#196) is never used in ERC1155RaribleRevealWave (contracts/erc-1155/ERC1155RaribleRevealWave.sol#8-197)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

renounceOwnership() should be declared external:
        - OwnableUpgradeable.renounceOwnership() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#60-63)
transferOwnership(address) should be declared external:
        - OwnableUpgradeable.transferOwnership(address) (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#69-73)
renounceOwnership() should be declared external:
        - Ownable.renounceOwnership() (node_modules/@openzeppelin/contracts/access/Ownable.sol#54-57)
transferOwnership(address) should be declared external:
        - Ownable.transferOwnership(address) (node_modules/@openzeppelin/contracts/access/Ownable.sol#63-67)
getAddress(string,string,string,string,uint256) should be declared external:
        - ERC1155RaribleRevealWaveFactoryC2.getAddress(string,string,string,string,uint256) (contracts/create-2/ERC1155RaribleRevealWaveFactoryC2.sol#63-75)
getAddress(string,string,string,string,address[],uint256) should be declared external:
        - ERC1155RaribleRevealWaveFactoryC2.getAddress(string,string,string,string,address[],uint256) (contracts/create-2/ERC1155RaribleRevealWaveFactoryC2.sol#82-94)
setRevealWave(uint256,string,string,bool,uint256) should be declared external:
        - ERC1155RaribleRevealWave.setRevealWave(uint256,string,string,bool,uint256) (contracts/erc-1155/ERC1155RaribleRevealWave.sol#72-85)
changeHiddenBaseURI(uint256,string) should be declared external:
        - ERC1155RaribleRevealWave.changeHiddenBaseURI(uint256,string) (contracts/erc-1155/ERC1155RaribleRevealWave.sol#87-94)
changeAddTokenURIToHiddenBaseURI(uint256,bool) should be declared external:
        - ERC1155RaribleRevealWave.changeAddTokenURIToHiddenBaseURI(uint256,bool) (contracts/erc-1155/ERC1155RaribleRevealWave.sol#96-102)
changeRevealBaseURI(uint256,string) should be declared external:
        - ERC1155RaribleRevealWave.changeRevealBaseURI(uint256,string) (contracts/erc-1155/ERC1155RaribleRevealWave.sol#104-113)
changeRevealDate(uint256,uint256) should be declared external:
        - ERC1155RaribleRevealWave.changeRevealDate(uint256,uint256) (contracts/erc-1155/ERC1155RaribleRevealWave.sol#115-121)
resetIsRevealedAndRevealURI(uint256) should be declared external:
        - ERC1155RaribleRevealWave.resetIsRevealedAndRevealURI(uint256) (contracts/erc-1155/ERC1155RaribleRevealWave.sol#123-129)
changeName(uint256,string) should be declared external:
        - ERC1155RaribleRevealWave.changeName(uint256,string) (contracts/erc-1155/ERC1155RaribleRevealWave.sol#132-139)
assignRevealWaveIdToTokenId(uint256,uint256) should be declared external:
        - ERC1155RaribleRevealWave.assignRevealWaveIdToTokenId(uint256,uint256) (contracts/erc-1155/ERC1155RaribleRevealWave.sol#142-145)
balanceOfBatch(address[],uint256[]) should be declared external:
        - ERC1155Upgradeable.balanceOfBatch(address[],uint256[]) (contracts/erc-1155/ERC1155Upgradeable.sol#105-124)
safeBatchTransferFrom(address,address,uint256[],uint256[],bytes) should be declared external:
        - ERC1155Upgradeable.safeBatchTransferFrom(address,address,uint256[],uint256[],bytes) (contracts/erc-1155/ERC1155Upgradeable.sol#178-214)
isValidSignature(bytes32,bytes) should be declared external:
        - ERC1271.isValidSignature(bytes32,bytes) (contracts/erc-1271/ERC1271.sol#20)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

ERC1155WoonklyNFTBeacon.sol

```
Address.isContract(address) (node_modules/@openzeppelin/contracts/utils/Address.sol#26-35) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/utils/Address.sol#33)
Address._verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#171-188) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts/utils/Address.sol#180-183)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Different versions of Solidity is used:
        - Version used: ['>=0.6.0<0.8.0', '>=0.6.2<0.8.0', '^0.7.0']
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/access/Ownable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/proxy/IBeacon.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/proxy/UpgradeableBeacon.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts/utils/Address.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#3)
        - ^0.7.0 (contracts/beacons/ERC1155RaribleBeacon.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

Address._verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#171-188) is never used and should be removed
Address.functionCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#75-81) is never used and should be removed
Address.functionCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#89-91) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#104-106) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#114-121) is never used and should be removed
Address.functionDelegateCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#153-155) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#163-169) is never used and should be removed
Address.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts/utils/Address.sol#129-131) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts/utils/Address.sol#139-145) is never used and should be removed
Address.sendValue(address,uint256) (node_modules/@openzeppelin/contracts/utils/Address.sol#53-59) is never used and should be removed
Context._msgData() (node_modules/@openzeppelin/contracts/utils/Context.sol#20-23) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/access/Ownable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/proxy/IBeacon.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/proxy/UpgradeableBeacon.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts/utils/Address.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#3) is too complex
Pragma version^0.7.0 (contracts/beacons/ERC1155RaribleBeacon.sol#3) allows old versions
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

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

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

```
Variable UpgradeableBeacon._implementation (node_modules/@openzeppelin/contracts/proxy/UpgradeableBeacon.sol#16) is too similar to UpgradeableBeacon.constructor(address).implementation_ (node_modules/@openzeppelin/contracts/proxy/Upgrade
ableBeacon.sol#27)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-are-too-similar

renounceOwnership() should be declared external:
        - Ownable.renounceOwnership() (node_modules/@openzeppelin/contracts/access/Ownable.sol#54-57)
transferOwnership(address) should be declared external:
        - Ownable.transferOwnership(address) (node_modules/@openzeppelin/contracts/access/Ownable.sol#63-67)
implementation() should be declared external:
        - UpgradeableBeacon.implementation() (node_modules/@openzeppelin/contracts/proxy/UpgradeableBeacon.sol#34-36)
upgradeTo(address) should be declared external:
        - UpgradeableBeacon.upgradeTo(address) (node_modules/@openzeppelin/contracts/proxy/UpgradeableBeacon.sol#48-51)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

ERC721LazyMintTransferProxy.sol

```
OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#74) shadows:
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variable-shadowing

AddressUpgradeable.isContract(address) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#26-35) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#33)
AddressUpgradeable._verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#147-164) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#156-159)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Different versions of Solidity is used:
        - Version used: ['0.7.6', '>=0.4.24<0.8.0', '>=0.6.0<0.8.0', '>=0.6.2<0.8.0', '>=0.6.9<0.8.0']
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/introspection/IERC165Upgradeable.sol#3)
        - >=0.4.24<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/proxy/Initializable.sol#4)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC721/IERC721Upgradeable.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#3)
        - >=0.6.9<0.8.0 (node_modules/@rarible/exchange-interfaces/contracts/ITransferProxy.sol#3)
        - v2 (node_modules/@rarible/exchange-interfaces/contracts/ITransferProxy.sol#4)
        - >=0.6.2<0.8.0 (node_modules/@rarible/lazy-mint/contracts/erc-721/IERC721LazyMint.sol#3)
        - v2 (node_modules/@rarible/lazy-mint/contracts/erc-721/IERC721LazyMint.sol#4)
        - >=0.6.2<0.8.0 (node_modules/@rarible/lazy-mint/contracts/erc-721/LibERC721LazyMint.sol#3)
        - 0.7.6 (node_modules/@rarible/lib-asset/contracts/LibAsset.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/LibPart.sol#3)
        - >=0.6.9<0.8.0 (contracts/transfer-proxy/lazy-mint/erc721/ERC721LazyMintTransferProxy.sol#3)
        - v2 (contracts/transfer-proxy/lazy-mint/erc721/ERC721LazyMintTransferProxy.sol#4)
        - >=0.6.9<0.8.0 (contracts/transfer-proxy/roles/OperatorRole.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

AddressUpgradeable._verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#147-164) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#79-81) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#89-91) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#104-106) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#114-121) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#128-131) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#139-145) is never used and should be removed
AddressUpgradeable.sendValue(address,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#53-59) is never used and should be removed
ContextUpgradeable.__Context_init() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#17-19) is never used and should be removed
ContextUpgradeable._msgData() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#27-30) is never used and should be removed
LibAsset.hash(LibAsset.Asset) (node_modules/@rarible/lib-asset/contracts/LibAsset.sol#39-45) is never used and should be removed
LibAsset.hash(LibAsset.AssetType) (node_modules/@rarible/lib-asset/contracts/LibAsset.sol#31-37) is never used and should be removed
LibERC721LazyMint.hash(LibERC721LazyMint.Mint721Data) (node_modules/@rarible/lazy-mint/contracts/erc-721/LibERC721LazyMint.sol#21-37) is never used and should be removed
LibPart.hash(LibPart.Part) (node_modules/@rarible/royalties/contracts/LibPart.sol#13-15) is never used and should be removed
OwnableUpgradeable.__Ownable_init() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#27-30) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/introspection/IERC165Upgradeable.sol#3) is too complex
Pragma version>=0.4.24<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/proxy/Initializable.sol#4) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC721/IERC721Upgradeable.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#3) is too complex
Pragma version>=0.6.9<0.8.0 (node_modules/@rarible/exchange-interfaces/contracts/ITransferProxy.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/lazy-mint/contracts/erc-721/IERC721LazyMint.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/lazy-mint/contracts/erc-721/LibERC721LazyMint.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/LibPart.sol#3) is too complex
Pragma version>=0.6.9<0.8.0 (contracts/transfer-proxy/lazy-mint/erc721/ERC721LazyMintTransferProxy.sol#3) is too complex
Pragma version>=0.6.9<0.8.0 (contracts/transfer-proxy/roles/OperatorRole.sol#3) is too complex
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in AddressUpgradeable.sendValue(address,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#53-59):
        - (success) = recipient.call{value: amount}() (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#57)
Low level call in AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#114-121):
        - (success,returndata) = target.call{value: value}(data) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#118)
Low level call in AddressUpgradeable.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#139-145):
        - (success,returndata) = target.staticcall(data) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#143)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function OwnableUpgradeable.__Ownable_init() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#27-30) is not in mixedCase
Function OwnableUpgradeable.__Ownable_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#32-36) is not in mixedCase
Variable OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#74) is not in mixedCase
Function ContextUpgradeable.__Context_init() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#17-19) is not in mixedCase
Function ContextUpgradeable.__Context_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#21-22) is not in mixedCase
Variable ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31) is not in mixedCase
Function OperatorRole.__OperatorRole_init() (contracts/transfer-proxy/roles/OperatorRole.sol#10-13) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#28)" inContextUpgradeable (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#16-32)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

LibERC721LazyMint.__INTERFACE_ID_MINT_AND_TRANSFER (node_modules/@rarible/lazy-mint/contracts/erc-721/LibERC721LazyMint.sol#9) is never used in LibERC721LazyMint (node_modules/@rarible/lazy-mint/contracts/erc-721/LibERC721LazyMint.sol#7-3
9)
OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#74) is never used in ERC721LazyMintTransferProxy (contracts/transfer-proxy/lazy-mint/erc721/ERC721LazyMintTransferProxy.sol#11-17)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

renounceOwnership() should be declared external:
        - OwnableUpgradeable.renounceOwnership() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#60-63)
transferOwnership(address) should be declared external:
        - OwnableUpgradeable.transferOwnership(address) (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#69-73)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

ERC1155LazyMintTransferProxy.sol

```
OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#74) shadows:
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variable-shadowing

AddressUpgradeable.isContract(address) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#26-35) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#33)
AddressUpgradeable._verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#147-164) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#156-159)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Different versions of Solidity is used:
        - Version used: ['0.7.6', '>=0.4.24<0.8.0', '>=0.6.0<0.8.0', '>=0.6.2<0.8.0', '>=0.6.9<0.8.0']
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/introspection/IERC165Upgradeable.sol#3)
        - >=0.4.24<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/proxy/Initializable.sol#4)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/IERC1155Upgradeable.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#3)
        - >=0.6.9<0.8.0 (node_modules/@rarible/exchange-interfaces/contracts/ITransferProxy.sol#3)
        - v2 (node_modules/@rarible/exchange-interfaces/contracts/ITransferProxy.sol#4)
        - >=0.6.2<0.8.0 (node_modules/@rarible/lazy-mint/contracts/erc-1155/IERC1155LazyMint.sol#3)
        - v2 (node_modules/@rarible/lazy-mint/contracts/erc-1155/IERC1155LazyMint.sol#4)
        - >=0.6.2<0.8.0 (node_modules/@rarible/lazy-mint/contracts/erc-1155/LibERC1155LazyMint.sol#3)
        - 0.7.6 (node_modules/@rarible/lib-asset/contracts/LibAsset.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/LibPart.sol#3)
        - >=0.6.9<0.8.0 (contracts/transfer-proxy/lazy-mint/erc1155/ERC1155LazyMintTransferProxy.sol#3)
        - v2 (contracts/transfer-proxy/lazy-mint/erc1155/ERC1155LazyMintTransferProxy.sol#4)
        - >=0.6.9<0.8.0 (contracts/transfer-proxy/roles/OperatorRole.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

AddressUpgradeable._verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#147-164) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#79-81) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#89-91) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#104-106) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#114-121) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#128-131) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#139-145) is never used and should be removed
AddressUpgradeable.sendValue(address,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#53-59) is never used and should be removed
ContextUpgradeable.__Context_init() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#17-19) is never used and should be removed
ContextUpgradeable._msgData() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#27-30) is never used and should be removed
LibAsset.hash(LibAsset.Asset) (node_modules/@rarible/lib-asset/contracts/LibAsset.sol#39-45) is never used and should be removed
LibAsset.hash(LibAsset.AssetType) (node_modules/@rarible/lib-asset/contracts/LibAsset.sol#31-37) is never used and should be removed
LibERC1155LazyMint.hash(LibERC1155LazyMint.Mint1155Data) (node_modules/@rarible/lazy-mint/contracts/erc-1155/LibERC1155LazyMint.sol#22-39) is never used and should be removed
LibPart.hash(LibPart.Part) (node_modules/@rarible/royalties/contracts/LibPart.sol#13-15) is never used and should be removed
OwnableUpgradeable.__Ownable_init() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#27-30) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
```

```
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/introspection/IERC165Upgradeable.sol#3) is too complex
Pragma version>=0.4.24<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/proxy/Initializable.sol#4) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/IERC1155Upgradeable.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#3) is too complex
Pragma version>=0.6.9<0.8.0 (node_modules/@rarible/exchange-interfaces/contracts/ITransferProxy.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/lazy-mint/contracts/erc-1155/IERC1155LazyMint.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/lazy-mint/contracts/erc-1155/LibERC1155LazyMint.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/LibPart.sol#3) is too complex
Pragma version>=0.6.9<0.8.0 (contracts/transfer-proxy/lazy-mint/erc1155/ERC1155LazyMintTransferProxy.sol#3) is too complex
Pragma version>=0.6.9<0.8.0 (contracts/transfer-proxy/roles/OperatorRole.sol#3) is too complex
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

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

Function OwnableUpgradeable.__Ownable_init() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#27-30) is not in mixedCase
Function OwnableUpgradeable.__Ownable_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#32-36) is not in mixedCase
Variable OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#74) is not in mixedCase
Function ContextUpgradeable.__Context_init() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#17-19) is not in mixedCase
Function ContextUpgradeable.__Context_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#21-22) is not in mixedCase
Variable ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31) is not in mixedCase
Function OperatorRole.__OperatorRole_init() (contracts/transfer-proxy/roles/OperatorRole.sol#10-13) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#28)" inContextUpgradeable (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#16-32)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

LibERC1155LazyMint._INTERFACE_ID_MINT_AND_TRANSFER (node_modules/@rarible/lazy-mint/contracts/erc-1155/LibERC1155LazyMint.sol#9) is never used in LibERC1155LazyMint (node_modules/@rarible/lazy-mint/contracts/erc-1155/LibERC1155LazyMint.s
ol#7-40)
OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#74) is never used in ERC1155LazyMintTransferProxy (contracts/transfer-proxy/lazy-mint/erc1155/ERC1155LazyMintTransferProxy.sol#11-16
)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

renounceOwnership() should be declared external:
        - OwnableUpgradeable.renounceOwnership() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#60-63)
transferOwnership(address) should be declared external:
        - OwnableUpgradeable.transferOwnership(address) (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#69-73)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

TransferProxy.sol

```
OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#74) shadows:
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variable-shadowing

AddressUpgradeable.isContract(address) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#26-35) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#33)
AddressUpgradeable._verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#147-164) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#156-159)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Different versions of Solidity is used:
        - Version used: ['>=0.4.24<0.8.0', '>=0.6.0<0.8.0', '>=0.6.2<0.8.0', '>=0.6.9<0.8.0']
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/introspection/IERC165Upgradeable.sol#3)
        - >=0.4.24<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/proxy/Initializable.sol#4)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/IERC1155Upgradeable.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC721/IERC721Upgradeable.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#3)
        - >=0.6.9<0.8.0 (node_modules/@rarible/exchange-interfaces/contracts/INftTransferProxy.sol#3)
        - v2 (node_modules/@rarible/exchange-interfaces/contracts/INftTransferProxy.sol#4)
        - >=0.6.9<0.8.0 (contracts/transfer-proxy/proxy/TransferProxy.sol#3)
        - >=0.6.9<0.8.0 (contracts/transfer-proxy/roles/OperatorRole.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

AddressUpgradeable._verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#147-164) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#79-81) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#89-91) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#104-106) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#114-121) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#129-131) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#139-145) is never used and should be removed
AddressUpgradeable.sendValue(address,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#53-59) is never used and should be removed
ContextUpgradeable.__Context_init() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#17-19) is never used and should be removed
ContextUpgradeable._msgData() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#27-30) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/introspection/IERC165Upgradeable.sol#3) is too complex
Pragma version>=0.4.24<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/proxy/Initializable.sol#4) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC1155/IERC1155Upgradeable.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC721/IERC721Upgradeable.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#3) is too complex
Pragma version>=0.6.9<0.8.0 (node_modules/@rarible/exchange-interfaces/contracts/INftTransferProxy.sol#3) is too complex
Pragma version>=0.6.9<0.8.0 (contracts/transfer-proxy/proxy/TransferProxy.sol#3) is too complex
Pragma version>=0.6.9<0.8.0 (contracts/transfer-proxy/roles/OperatorRole.sol#3) is too complex
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

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

Function OwnableUpgradeable.__Ownable_init() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#27-30) is not in mixedCase
Function OwnableUpgradeable.__Ownable_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#32-36) is not in mixedCase
Variable OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#74) is not in mixedCase
Function ContextUpgradeable.__Context_init() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#17-19) is not in mixedCase
Function ContextUpgradeable.__Context_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#21-22) is not in mixedCase
Variable ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31) is not in mixedCase
Function TransferProxy.__TransferProxy_init() (contracts/transfer-proxy/proxy/TransferProxy.sol#10-12) is not in mixedCase
Function OperatorRole.__OperatorRole_init() (contracts/transfer-proxy/roles/OperatorRole.sol#10-13) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#28)" inContextUpgradeable (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#16-32)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#74) is never used in TransferProxy (contracts/transfer-proxy/proxy/TransferProxy.sol#8-21)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

renounceOwnership() should be declared external:
        - OwnableUpgradeable.renounceOwnership() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#60-63)
transferOwnership(address) should be declared external:
        - OwnableUpgradeable.transferOwnership(address) (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#69-73)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

ERC20TransferProxy.sol

```
OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#74) shadows:
        - ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variable-shadowing

AddressUpgradeable.isContract(address) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#26-35) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#33)
AddressUpgradeable._verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#147-164) uses assembly
        - INLINE ASM (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#156-159)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Different versions of Solidity is used:
        - Version used: ['>=0.4.24<0.8.0', '>=0.6.0<0.8.0', '>=0.6.2<0.8.0', '>=0.6.9<0.8.0']
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#3)
        - >=0.4.24<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/proxy/Initializable.sol#4)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/IERC20Upgradeable.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#3)
        - >=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#3)
        - >=0.6.9<0.8.0 (node_modules/@rarible/exchange-interfaces/contracts/IERC20TransferProxy.sol#3)
        - v2 (node_modules/@rarible/exchange-interfaces/contracts/IERC20TransferProxy.sol#4)
        - >=0.6.9<0.8.0 (contracts/transfer-proxy/proxy/ERC20TransferProxy.sol#3)
        - >=0.6.9<0.8.0 (contracts/transfer-proxy/roles/OperatorRole.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

AddressUpgradeable._verifyCallResult(bool,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#147-164) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#79-81) is never used and should be removed
AddressUpgradeable.functionCall(address,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#89-91) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#104-106) is never used and should be removed
AddressUpgradeable.functionCallWithValue(address,bytes,uint256,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#114-121) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#129-131) is never used and should be removed
AddressUpgradeable.functionStaticCall(address,bytes,string) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#139-145) is never used and should be removed
AddressUpgradeable.sendValue(address,uint256) (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#53-59) is never used and should be removed
ContextUpgradeable.__Context_init() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#17-19) is never used and should be removed
ContextUpgradeable._msgData() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#27-30) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
```

```
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#3) is too complex
Pragma version>=0.4.24<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/proxy/Initializable.sol#4) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/token/ERC20/IERC20Upgradeable.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/AddressUpgradeable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#3) is too complex
Pragma version>=0.6.9<0.8.0 (node_modules/@rarible/exchange-interfaces/contracts/IERC20TransferProxy.sol#3) is too complex
Pragma version>=0.6.9<0.8.0 (contracts/transfer-proxy/proxy/ERC20TransferProxy.sol#3) is too complex
Pragma version>=0.6.9<0.8.0 (contracts/transfer-proxy/roles/OperatorRole.sol#3) is too complex
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

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

Function OwnableUpgradeable.__Ownable_init() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#27-30) is not in mixedCase
Function OwnableUpgradeable.__Ownable_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#32-36) is not in mixedCase
Variable OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#74) is not in mixedCase
Function ContextUpgradeable.__Context_init() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#17-19) is not in mixedCase
Function ContextUpgradeable.__Context_init_unchained() (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#21-22) is not in mixedCase
Variable ContextUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#31) is not in mixedCase
Function ERC20TransferProxy.__ERC20TransferProxy_init() (contracts/transfer-proxy/proxy/ERC20TransferProxy.sol#10-12) is not in mixedCase
Function OperatorRole.__OperatorRole_init() (contracts/transfer-proxy/roles/OperatorRole.sol#10-13) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#28)" inContextUpgradeable (node_modules/@openzeppelin/contracts-upgradeable/utils/ContextUpgradeable.sol#16-32)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

OwnableUpgradeable.__gap (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#74) is never used in ERC20TransferProxy (contracts/transfer-proxy/proxy/ERC20TransferProxy.sol#8-17)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

renounceOwnership() should be declared external:
        - OwnableUpgradeable.renounceOwnership() (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#60-63)
transferOwnership(address) should be declared external:
        - OwnableUpgradeable.transferOwnership(address) (node_modules/@openzeppelin/contracts-upgradeable/access/OwnableUpgradeable.sol#69-73)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

AssetMatcherCollection.sol

```
Different versions of Solidity is used:
        - Version used: ['^0.7.6', '^>=0.6.2<0.8.0']
        - 0.7.6 (node_modules/@rarible/exchange-interfaces/contracts/IAssetMatcher.sol#3)
        - v2 (node_modules/@rarible/exchange-interfaces/contracts/IAssetMatcher.sol#4)
        - >=0.6.2<0.8.0 (node_modules/@rarible/lazy-mint/contracts/erc-1155/LibERC1155LazyMint.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@rarible/lazy-mint/contracts/erc-721/LibERC721LazyMint.sol#3)
        - 0.7.6 (node_modules/@rarible/lib-asset/contracts/LibAsset.sol#3)
        - >=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/LibPart.sol#3)
        - 0.7.6 (contracts/AssetMatcherCollection.sol#3)
        - v2 (contracts/AssetMatcherCollection.sol#4)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

LibAsset.hash(LibAsset.Asset) (node_modules/@rarible/lib-asset/contracts/LibAsset.sol#39-45) is never used and should be removed
LibAsset.hash(LibAsset.AssetType) (node_modules/@rarible/lib-asset/contracts/LibAsset.sol#21-37) is never used and should be removed
LibERC1155LazyMint.hash(LibERC1155LazyMint.Mint1155Data) (node_modules/@rarible/lazy-mint/contracts/erc-1155/LibERC1155LazyMint.sol#22-39) is never used and should be removed
LibERC721LazyMint.hash(LibERC721LazyMint.Mint721Data) (node_modules/@rarible/lazy-mint/contracts/erc-721/LibERC721LazyMint.sol#21-37) is never used and should be removed
LibPart.hash(LibPart.Part) (node_modules/@rarible/royalties/contracts/LibPart.sol#13-15) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/lazy-mint/contracts/erc-1155/LibERC1155LazyMint.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/lazy-mint/contracts/erc-721/LibERC721LazyMint.sol#3) is too complex
Pragma version>=0.6.2<0.8.0 (node_modules/@rarible/royalties/contracts/LibPart.sol#3) is too complex
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

LibERC1155LazyMint._INTERFACE_ID_MINT_AND_TRANSFER (node_modules/@rarible/lazy-mint/contracts/erc-1155/LibERC1155LazyMint.sol#9) is never used in LibERC1155LazyMint (node_modules/@rarible/lazy-mint/contracts/erc-1155/LibERC1155LazyMint.s
o1#7-40)
LibERC721LazyMint._INTERFACE_ID_MINT_AND_TRANSFER (node_modules/@rarible/lazy-mint/contracts/erc-721/LibERC721LazyMint.sol#9) is never used in LibERC721LazyMint (node_modules/@rarible/lazy-mint/contracts/erc-721/LibERC721LazyMint.sol#7-3
9)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

matchAssets(LibAsset.AssetType,LibAsset.AssetType) should be declared external:
        - AssetMatcherCollection.matchAssets(LibAsset.AssetType,LibAsset.AssetType) (contracts/AssetMatcherCollection.sol#17-32)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

- No major issues found by Slither.

4.2 AUTOMATED SECURITY SCAN

Description:

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

MythX results:

ProxyAdmin.sol

No issues found by MythX

ExchangeV2.sol

No issues found by MythX

RoyaltiesRegistry.sol

Report for contracts/royalties-registry/RoyaltiesRegistry.sol
https://dashboard.mythx.io/#/console/analyses/23f23a4e-054d-4a5d-9152-381d51d0f5ed

Line	SWC Title	Severity	Short Description
3	(SWC-103) Floating Pragma	Low	A floating pragma is set.
16	(SWC-123) Requirement Violation	Low	Requirement violation.
119	(SWC-123) Requirement Violation	Low	Requirement violation.
119	(SWC-104) Unchecked Call Return Value	Medium	Unchecked return value from external call.
125	(SWC-104) Unchecked Call Return Value	Medium	Unchecked return value from external call.
125	(SWC-113) DoS with Failed Call	Medium	Multiple calls are executed in the same transaction.
131	(SWC-104) Unchecked Call Return Value	Medium	Unchecked return value from external call.
243	(SWC-104) Unchecked Call Return Value	Medium	Unchecked return value from external call.

ERC721WoonklyNFT.sol

No issues found by MythX

ERC721WoonklyNFTMinimal.sol

Report for contracts/erc-721-minimal/ERC721RaribleMinimal.sol
https://dashboard.mythx.io/#/console/analyses/1b6a58bd-ae4e-4437-9991-ac6c83e9a747

Line	SWC Title	Severity	Short Description
10	(SWC-108) State Variable Default Visibility	Low	State variable visibility is not set.

ERC721WoonklyNFTRevealWave.sol

Report for contracts/erc-721/ERC721RaribleRevealWave.sol
https://dashboard.mythx.io/#/console/analyses/0ef6f57d-b27e-4b23-b800-e560ca445729

Line	SWC Title	Severity	Short Description
24	(SWC-108) State Variable Default Visibility	Low	State variable visibility is not set.

ERC721WoonklyNFTMinimalRevealWave.sol

Report for contracts/erc-721-minimal/ERC721RaribleMinimalRevealWave.sol
https://dashboard.mythx.io/#/console/analyses/c1ac15a1-8aaf-43ff-a889-a469c72a05df

Line	SWC Title	Severity	Short Description
24	(SWC-108) State Variable Default Visibility	Low	State variable visibility is not set.

ERC721WoonklyNFTFactoryC2.sol

Report for contracts/create-2/ERC721RaribleFactoryC2.sol
https://dashboard.mythx.io/#/console/analyses/5151d07a-0090-48f7-a658-0190374e8fc4

Line	SWC Title	Severity	Short Description
3	(SWC-103) Floating Pragma	Low	A floating pragma is set.
18	(SWC-108) State Variable Default Visibility	Low	State variable visibility is not set.
19	(SWC-108) State Variable Default Visibility	Low	State variable visibility is not set.

ERC721WoonklyNFTRevealWaveFactoryC2.sol

Report for contracts/create-2/ERC721RaribleRevealWaveFactoryC2.sol
https://dashboard.mythx.io/#/console/analyses/35cd74a2-3a7c-4620-ba5f-6bcbc97ae48c

Line	SWC Title	Severity	Short Description
3	(SWC-103) Floating Pragma	Low	A floating pragma is set.
19	(SWC-108) State Variable Default Visibility	Low	State variable visibility is not set.
20	(SWC-108) State Variable Default Visibility	Low	State variable visibility is not set.

ERC721WoonklyNFTMinimalBeacon.sol

Report for contracts/beacons/ERC721RaribleMinimalBeacon.sol
https://dashboard.mythx.io/#/console/analyses/112f07b7-c7f7-4f93-8f60-ce78d36128b5

Line	SWC Title	Severity	Short Description
3	(SWC-103) Floating Pragma	Low	A floating pragma is set.

ERC721WoonklyNFTBeacon.sol

Report for contracts/beacons/ERC721RaribleBeacon.sol
https://dashboard.mythx.io/#/console/analyses/3c80de3e-71f4-430a-9359-47cf934ec3bd

Line	SWC Title	Severity	Short Description
3	(SWC-103) Floating Pragma	Low	A floating pragma is set.

ERC1155WoonklyNFT.sol

Report for contracts/erc-1155/ERC1155Rarible.sol
https://dashboard.mythx.io/#/console/analyses/e145b6ea-2278-419e-a414-310d024cee87
https://dashboard.mythx.io/#/console/analyses/9f1c4c10-5143-467d-b9d4-0b62fce02622

Line	SWC Title	Severity	Short Description
8	(SWC-123) Requirement Violation	Low	Requirement violation.
10	(SWC-108) State Variable Default Visibility	Low	State variable visibility is not set.

ERC1155WoonklyNFTRevealWave.sol

Report for contracts/erc-1155/ERC1155RaribleRevealWave.sol
https://dashboard.mythx.io/#/console/analyses/4e2a810a-06be-4dbb-9a1c-369806a48762

Line	SWC Title	Severity	Short Description
24	(SWC-108) State Variable Default Visibility	Low	State variable visibility is not set.

ERC1155WoonklyNFTFactoryC2.sol

Report for contracts/create-2/ERC1155RaribleFactoryC2.sol
https://dashboard.mythx.io/#/console/analyses/9f1c4c10-5143-467d-b9d4-0b62fce02622

Line	SWC Title	Severity	Short Description
3	(SWC-103) Floating Pragma	Low	A floating pragma is set.
18	(SWC-108) State Variable Default Visibility	Low	State variable visibility is not set.
19	(SWC-108) State Variable Default Visibility	Low	State variable visibility is not set.

ERC1155WoonklyNFTRevealWaveFactoryC2.sol

Report for contracts/create-2/ERC1155RaribleRevealWaveFactoryC2.sol
https://dashboard.mythx.io/#/console/analyses/856778f9-b096-4a62-af99-368b2191c444

Line	SWC Title	Severity	Short Description
3	(SWC-103) Floating Pragma	Low	A floating pragma is set.
18	(SWC-108) State Variable Default Visibility	Low	State variable visibility is not set.
19	(SWC-108) State Variable Default Visibility	Low	State variable visibility is not set.

ERC1155WoonklyNFTBeacon.sol

Report for contracts/beacons/ERC1155RaribleBeacon.sol
https://dashboard.mythx.io/#/console/analyses/52b7286e-dde4-4117-b54e-ea00dad5163c

Line	SWC Title	Severity	Short Description
3	(SWC-103) Floating Pragma	Low	A floating pragma is set.

ERC721LazyMintTransferProxy.sol

Report for contracts/transfer-proxy/lazy-mint/erc721/ERC721LazyMintTransferProxy.sol
https://dashboard.mythx.io/#/console/analyses/b42da0fc-c578-4a7b-ad1d-d04a84163be2
https://dashboard.mythx.io/#/console/analyses/27e8464c-9a85-4f1c-8719-8a255f82b79c

Line	SWC Title	Severity	Short Description
3	(SWC-103) Floating Pragma	Low	A floating pragma is set.

ERC1155LazyMintTransferProxy.sol

Report for contracts/transfer-proxy/lazy-mint/erc1155/ERC1155LazyMintTransferProxy.sol
https://dashboard.mythx.io/#/console/analyses/8f3b311c-d71a-4d77-9825-d31cab4fe80b

Line	SWC Title	Severity	Short Description
3	(SWC-103) Floating Pragma	Low	A floating pragma is set.

TransferProxy.sol

Report for contracts/transfer-proxy/proxy/TransferProxy.sol
https://dashboard.mythx.io/#/console/analyses/5ec0337e-cb53-47d3-a2f8-b109d3b07090

Line	SWC Title	Severity	Short Description
3	(SWC-103) Floating Pragma	Low	A floating pragma is set.
15	(SWC-107) Reentrancy	Low	A call to a user-supplied address is executed.
19	(SWC-107) Reentrancy	Low	A call to a user-supplied address is executed.

ERC20TransferProxy.sol

Report for contracts/transfer-proxy/proxy/ERC20TransferProxy.sol
https://dashboard.mythx.io/#/console/analyses/ebcf96a2-b83c-4d63-b901-a638bc1540ed

Line	SWC Title	Severity	Short Description
3	(SWC-103) Floating Pragma	Low	A floating pragma is set.
8	(SWC-123) Requirement Violation	Low	Requirement violation.
15	(SWC-123) Requirement Violation	Low	Requirement violation.
15	(SWC-107) Reentrancy	Low	A call to a user-supplied address is executed.

AssetMatcherCollection.sol

No issues found by MythX

- The Integer Overflows and Underflows flagged by MythX were checked individually and were determined to be mathematically impossible.
- Assert violations are false positives.

THANK YOU FOR CHOOSING

// HALBORN