```
'npx hardhat compile --force' running
Compiled 13 Solidity files successfully

(node:5173) ExperimentalWarning: stream/web is an experimental feature. This feature could change at any time
(Use `node --trace-warnings ...` to show where the warning was created)


Reentrancy in PixelmonEvolution.evolvePixelmon(uint256[],uint256[],uint256[],uint256,uint256,uint256,bytes)
(contracts/PixelmonEvolution.sol#181-253):
   External calls:
   - SERUM_CONTRACT.safeBatchTransferFrom(msg.sender,BURNER_ADDRESS,serumIds,serumAmounts,)
(contracts/PixelmonEvolution.sol#225)
   - PIXELMON_CONTRACT.mintEvolvedPixelmon(address(this),evolutionStage) (contracts/PixelmonEvolution.sol#234)
   State variables written after the call(s):
   - nextEvolvePixelmonId ++ (contracts/PixelmonEvolution.sol#236)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-1

PixelmonEvolution.addStakedTokenInformation(uint256,uint256,address).owner (contracts/PixelmonEvolution.sol#274) shadows:
   - Ownable.owner() (node_modules/@openzeppelin/contracts/access/Ownable.sol#43-45) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

PixelmonEvolution.constructor(address,address,address).signer (contracts/PixelmonEvolution.sol#124) lacks a zero-check on :
         - SIGNER = signer (contracts/PixelmonEvolution.sol#127)
PixelmonEvolution.setSignerAddress(address).signer (contracts/PixelmonEvolution.sol#161) lacks a zero-check on :
         - SIGNER = signer (contracts/PixelmonEvolution.sol#162)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation

PixelmonEvolution.evolvePixelmon(uint256[],uint256[],uint256[],uint256,uint256,uint256,bytes) (contracts/PixelmonEvolution.sol#181-253) has
external calls inside a loop: PIXELMON_CONTRACT.mintEvolvedPixelmon(address(this),evolutionStage)
(contracts/PixelmonEvolution.sol#234)
```

```
PixelmonEvolution.evolvePixelmon(uint256[],uint256[],uint256[],uint256,uint256,uint256,bytes) (contracts/PixelmonEvolution.sol#181-253) has
external calls inside a loop: PIXELMON_CONTRACT.safeTransferFrom(msg.sender,address(this),tokenId,)
(contracts/PixelmonEvolution.sol#239)
PixelmonEvolution.claimPixelmonToken(uint256[]) (contracts/PixelmonEvolution.sol#257-268) has external calls inside a loop:
PIXELMON_CONTRACT.safeTransferFrom(address(this),msg.sender,tokenId,) (contracts/PixelmonEvolution.sol#264)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation/#calls-inside-a-loop

Reentrancy in PixelmonEvolution.evolvePixelmon(uint256[],uint256[],uint256[],uint256,uint256,uint256,bytes)
(contracts/PixelmonEvolution.sol#181-253):
   External calls:
   - SERUM_CONTRACT.safeBatchTransferFrom(msg.sender,BURNER_ADDRESS,serumIds,serumAmounts,)
(contracts/PixelmonEvolution.sol#225)
   State variables written after the call(s):
   - evolutionPair[tokenId] = nextEvolvePixelmonId (contracts/PixelmonEvolution.sol#231)
Reentrancy in PixelmonEvolution.evolvePixelmon(uint256[],uint256[],uint256[],uint256,uint256,uint256,bytes)
(contracts/PixelmonEvolution.sol#181-253):
   External calls:
   - SERUM_CONTRACT.safeBatchTransferFrom(msg.sender,BURNER_ADDRESS,serumIds,serumAmounts,)
(contracts/PixelmonEvolution.sol#225)
   - PIXELMON_CONTRACT.mintEvolvedPixelmon(address(this),evolutionStage) (contracts/PixelmonEvolution.sol#234)
   State variables written after the call(s):
   - addStakedTokenInformation(nextEvolvePixelmonId,stakedFor,msg.sender) (contracts/PixelmonEvolution.sol#235)
          - vault[tokenId] = StakedTokenInformation(owner,tokenId,block.timestamp,stakedFor) (contracts/PixelmonEvolution.sol#275)
Reentrancy in PixelmonEvolution.evolvePixelmon(uint256[],uint256[],uint256[],uint256,uint256,uint256,bytes)
(contracts/PixelmonEvolution.sol#181-253):
   External calls:
   - SERUM_CONTRACT.safeBatchTransferFrom(msg.sender,BURNER_ADDRESS,serumIds,serumAmounts,)
(contracts/PixelmonEvolution.sol#225)
   - PIXELMON_CONTRACT.mintEvolvedPixelmon(address(this),evolutionStage) (contracts/PixelmonEvolution.sol#234)
   - PIXELMON_CONTRACT.safeTransferFrom(msg.sender,address(this),tokenId,) (contracts/PixelmonEvolution.sol#239)
   State variables written after the call(s):
    - addStakedTokenInformation(tokenId,stakedFor,msg.sender) (contracts/PixelmonEvolution.sol#240)
            - vault[tokenId] = StakedTokenInformation(owner,tokenId,block.timestamp,stakedFor) (contracts/PixelmonEvolution.sol#275)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2

Reentrancy in PixelmonEvolution.evolvePixelmon(uint256[],uint256[],uint256[],uint256,uint256,uint256,bytes)
(contracts/PixelmonEvolution.sol#181-253):
    External calls:
    - SERUM_CONTRACT.safeBatchTransferFrom(msg.sender,BURNER_ADDRESS,serumIds,serumAmounts,)
(contracts/PixelmonEvolution.sol#225)
    Event emitted after the call(s):
    - PixelmonBatchEvolve(msg.sender,nonce,pixelmonTokenIds,serumIds,serumAmounts,evolutionStage,evolvedTokenStartingId,pixelmon
evolved) (contracts/PixelmonEvolution.sol#243-252)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

PixelmonEvolution.claimPixelmonToken(uint256[]) (contracts/PixelmonEvolution.sol#257-268) uses timestamp for comparisons
    Dangerous comparisons:
    - msg.sender != vault[tokenId].owner (contracts/PixelmonEvolution.sol#260)
PixelmonEvolution.checkTimeLock(uint256,uint256) (contracts/PixelmonEvolution.sol#330-338) uses timestamp for comparisons
    Dangerous comparisons:
    - require(bool,string)((block.timestamp - stakedAt) > stakedFor,Tokens cannot be claimed before its chosen minimum time lock period)
(contracts/PixelmonEvolution.sol#333-336)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

ECDSA.tryRecover(bytes32,bytes) (node_modules/@openzeppelin/contracts/utils/cryptography/ECDSA.sol#57-74) uses assembly
    - INLINE ASM (node_modules/@openzeppelin/contracts/utils/cryptography/ECDSA.sol#65-69)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Different versions of Solidity are used:
    - Version used: ['^0.8.0', '^0.8.16']
    - ^0.8.0 (node_modules/@openzeppelin/contracts/access/Ownable.sol#4)
    - ^0.8.0 (node_modules/@openzeppelin/contracts/interfaces/IERC721Receiver.sol#4)
```

```
	- ^0.8.0 (node_modules/@openzeppelin/contracts/security/ReentrancyGuard.sol#4)
	- ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC1155/IERC1155.sol#4)
	- ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/IERC721.sol#4)
	- ^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/IERC721Receiver.sol#4)
	- ^0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#4)
	- ^0.8.0 (node_modules/@openzeppelin/contracts/utils/Strings.sol#4)
	- ^0.8.0 (node_modules/@openzeppelin/contracts/utils/cryptography/ECDSA.sol#4)
	- ^0.8.0 (node_modules/@openzeppelin/contracts/utils/cryptography/draft-EIP712.sol#4)
	- ^0.8.0 (node_modules/@openzeppelin/contracts/utils/introspection/IERC165.sol#4)
	- ^0.8.16 (contracts/IPixelmon.sol#2)
	- ^0.8.16 (contracts/PixelmonEvolution.sol#2)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

PixelmonEvolution.evolvePixelmon(uint256[],uint256[],uint256[],uint256,uint256,uint256,bytes) (contracts/PixelmonEvolution.sol#181-253) has
costly operations inside a loop:
	- nextEvolvePixelmonId ++ (contracts/PixelmonEvolution.sol#236)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#costly-operations-inside-a-loop

Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/access/Ownable.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/interfaces/IERC721Receiver.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/security/ReentrancyGuard.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC1155/IERC1155.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/IERC721.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/token/ERC721/IERC721Receiver.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/Context.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/Strings.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/cryptography/ECDSA.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/cryptography/draft-EIP712.sol#4) allows old versions
Pragma version^0.8.0 (node_modules/@openzeppelin/contracts/utils/introspection/IERC165.sol#4) allows old versions
Pragma version^0.8.16 (contracts/IPixelmon.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version^0.8.16 (contracts/PixelmonEvolution.sol#2) necessitates a version too recent to be trusted. Consider deploying with
0.6.12/0.7.6/0.8.7
solc-0.8.16 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

PixelmonEvolution (contracts/PixelmonEvolution.sol#25-360) should inherit from IERC721Receiver
(node_modules/@openzeppelin/contracts/token/ERC721/IERC721Receiver.sol#11-27)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-inheritance

Variable EIP712._CACHED_DOMAIN_SEPARATOR (node_modules/@openzeppelin/contracts/utils/cryptography/draft-EIP712.sol#31) is not in
mixedCase
Variable EIP712._CACHED_CHAIN_ID (node_modules/@openzeppelin/contracts/utils/cryptography/draft-EIP712.sol#32) is not in mixedCase
Variable EIP712._CACHED_THIS (node_modules/@openzeppelin/contracts/utils/cryptography/draft-EIP712.sol#33) is not in mixedCase
Variable EIP712._HASHED_NAME (node_modules/@openzeppelin/contracts/utils/cryptography/draft-EIP712.sol#35) is not in mixedCase
Variable EIP712._HASHED_VERSION (node_modules/@openzeppelin/contracts/utils/cryptography/draft-EIP712.sol#36) is not in mixedCase
Variable EIP712._TYPE_HASH (node_modules/@openzeppelin/contracts/utils/cryptography/draft-EIP712.sol#37) is not in mixedCase
Variable PixelmonEvolution.PIXELMON_CONTRACT (contracts/PixelmonEvolution.sol#40) is not in mixedCase
Variable PixelmonEvolution.SERUM_CONTRACT (contracts/PixelmonEvolution.sol#42) is not in mixedCase
Variable PixelmonEvolution.SIGNER (contracts/PixelmonEvolution.sol#44) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

PixelmonEvolution.slitherConstructorConstantVariables() (contracts/PixelmonEvolution.sol#25-360) uses literals with too many digits:
    - BURNER_ADDRESS = 0x000000000000000000000000000000000000dEaD (contracts/PixelmonEvolution.sol#28)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits

renounceOwnership() should be declared external:
    - Ownable.renounceOwnership() (node_modules/@openzeppelin/contracts/access/Ownable.sol#61-63)
transferOwnership(address) should be declared external:
    - Ownable.transferOwnership(address) (node_modules/@openzeppelin/contracts/access/Ownable.sol#69-72)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
. analyzed (12 contracts with 78 detectors), 43 result(s) found
```