BLOCKHAT

Security

MADNFT

Smart Contract Security Audit

Prepared by BlockHat

March 11th, 2023 - March 19th, 2023

Document Properties

Client	Jacob Clay
Version	2.0
Classification	Public

Scope

The MADNFT Contract in the MADNFT Repository

Repo	Owner
`https://github.com/madnfts/madnfts-solidity-contracts/tree/release/1.0`	MADNFTs

Files	MD5 Hash
contracts/EventsAndErrors.sol	1595c05c03c74d4125f5c57ead55cbb3
contracts/MAD.sol	224fd2f1d2afdf25fad3c9b2aefecb4b
contracts/MADFactory1155.sol	63c0b5ddb496f36907880f97c608cb00
contracts/MADFactory721.sol	f3a6d1d273480d37304cc23bbf40bfbf
contracts/MADMarketplace1155.sol	825fd921c210d9940ab4e754b93a5b0d
contracts/MADMarketplace721.sol	96ae2a73e696b476ed8ef986fdb4f973
contracts/MADRouter1155.sol	aaa9be5bfcb4afd9d60169f3ccc66dfb
contracts/MADRouter721.sol	084f1f113b3e639df48eb604c6eb99cb

contracts/Types.sol	211938fc82b786b14a391ee5f2a8ee5f
contracts/lib/utils/Counters.sol	d216f3aabd6e9fcf0c041a8edaa895df
contracts/lib/utils/CREATE3.sol	3a09bbdf98d7284b6585ea335ada4b8b
contracts/lib/utils/MerkleProof.sol	c964b36fee0132365c17d4a53aeca414
contracts/lib/utils/SafeTransferLib.sol	0b7b6b404e478867d79ed10c7d8d847b
contracts/lib/utils/Strings.sol	5ab70c6b68313b6d7b196dbca9f17c34
contracts/lib/tokens/ERC20.sol	3dfc6812f6e6197b27e223895344d352
contracts/lib/tokens/ERC721/Impl/ERC721Basic.sol	5618e75d6121d5458bf04a6147fba447
contracts/lib/tokens/ERC721/Impl/ERC721Lazy.sol	1bae6600438ace20cadc430f63038f1b
contracts/lib/tokens/ERC721/Impl/ERC721Minimal.sol	0fe039b83f1c2919b9139cd35efea205
contracts/lib/tokens/ERC721/Impl/ERC721Whitelist.sol	d67d93261e5426006494a64e9e0afc8b
contracts/lib/tokens/ERC721/Base/ERC721.sol	bf2f901d011ad5c0990e4831fc5148cc
contracts/lib/tokens/ERC721/Base/utils/ERC721Holder.sol	acc5d77cdf104884dd4d1158bf289d48
contracts/lib/tokens/ERC721/Base/interfaces/ERC721EventAndErrors.sol	5240053a3f2a57541ad8ef6a28560f98
contracts/lib/tokens/ERC721/Base/interfaces/IERC721.sol	7b54e6881e257c4c934dbb6dcc425b02
contracts/lib/tokens/ERC1155/Impl/ERC1155Basic.sol	348853cd9c321955ae879e939337b29a

contracts/lib/tokens/ERC1155/Impl/ERC1155Lazy.sol	4ab9ce5078e083cad982a72b9e5d9e66
contracts/lib/tokens/ERC1155/Impl/ERC1155Minimal.sol	ee19697195cf28b30e8acfda33750522
contracts/lib/tokens/ERC1155/Impl/ERC1155Whitelist.sol	bb5e42d2cc6180be1a14cba1ccfe6cdf
contracts/lib/tokens/ERC1155/Base/ERC1155B.sol	b09c1d1b5cc6e121f1acdb2ba40f73e3
contracts/lib/tokens/ERC1155/Base/utils/ERC1155Holder.sol	81c56017acbde380827c3f0ac97463df
contracts/lib/tokens/ERC1155/Base/interfaces/ERC1155EventAndErrors.sol	4b21c51b30b96d253b63ce4d90e1511a
contracts/lib/tokens/ERC1155/Base/interfaces/IERC1155.sol	bf24bd68d21bbe70e3874f9673e32870
contracts/lib/tokens/common/ERC2981.sol	ccd94fe2933c3a11a3fe5e661e10977d
contracts/lib/tokens/common/FeeOracle.sol	19e1b61b398275c51453133e7981bcfe
contracts/lib/test/erc1155-mock.sol	02fdd44bbe56a2fdfd659987ebb77048
contracts/lib/test/erc20-mock.sol	4af18803ffbbd8774c9030ca9b3e217b
contracts/lib/test/erc2981-mock.sol	92595f27f4b695874873df0dffdd68d2
contracts/lib/test/erc721-mock.sol	4b3024b5a73ab7f6aff779b6f738e9d5
contracts/lib/test/test-interfaces.sol	59a8c882c8fa1edbc28c99a04b7521f8
contracts/lib/splitter/SplitterEventsAndErrors.sol	367908e4e3cf487917bec89406e69fa1
contracts/lib/splitter/SplitterImpl.sol	c72f9d3da9ea65a97ac7a5767dd2981c
contracts/lib/security/DCPrevent.sol	3a2308d6e5759a1f81061109e1942ac8

contracts/lib/security/Pausable.sol	1286b0b6207ae026ec9922d7345c06f6
contracts/lib/security/ReentrancyGuard.sol	03355df147e2ef07cfb8c09d346a9cd2
contracts/lib/deployers/ERC1155Deployer.sol	0ea61a892fabec2a881a5034277856c9
contracts/lib/deployers/ERC721Deployer.sol	1f03487f68aea5b5c06ee10032e47cd3
contracts/lib/deployers/SplitterDeployer.sol	0c407b49fed828cc74553bccb81633c1
contracts/lib/auth/FactoryVerifier.sol	c7f3d59a47c84642f5ed386d1088944e
contracts/lib/auth/Owned.sol	a880f344c057b2682d5ec6f03db96abf

Contacts

COMPANY	CONTACT
BlockHat	www.fiverr.com/blockhat

Contents

1 Introduction

MADNFT engaged BlockHat to conduct a security assessment on the MADNFT beginning on March 11th, 2023 and ending March 19th, 2023. In this report, we detail our methodical approach to evaluate potential security issues associated with the implementation of smart contracts, by exposing possible semantic discrepancies between the smart contract code and design document, and by recommending additional ideas to optimize the existing code. Our findings indicate that the current version of smart contracts can still be enhanced further due to the presence of many security and performance concerns.

This document summarizes the findings of our audit.

1.1 About MADNFT

MADNFT is a nft marketplace that allows the minting of 721 and 1155 NFTs on the harmony blockchain. There is a configurable mint fee of 0.25ONE and configurable platform fee set at 10 % . User can trade other external harmony NFTs on the marketplace too.

Issuer	Jacob Clay
Website	`https://v1.madnfts.io/`
Type	Solidity Smart Contract
Audit Method	Whitebox

1.2 Approach & Methodology

BlockHat used a combination of manual and automated security testing to achieve a balance between efficiency, timeliness, practicability, and correctness within the audit's scope. While manual testing is advised for identifying problems in logic, procedure, and implementation, automated testing techniques help to expand the coverage of smart contracts and can quickly detect code that does not comply with security best practices.

1.2.1 Risk Methodology

Vulnerabilities or bugs identified by BlockHat are ranked using a risk assessment technique that considers both the LIKELIHOOD and IMPACT of a security incident. This framework is effective at conveying the features and consequences of technological vulnerabilities.

Its quantitative paradigm enables repeatable and precise measurement, while also revealing the underlying susceptibility characteristics that were used to calculate the Risk scores. A risk level will be assigned to each vulnerability on a scale of 5 to 1, with 5 indicating the greatest possibility or impact.

- Likelihood quantifies the probability of a certain vulnerability being discovered and exploited in the untamed.
- Impact quantifies the technical and economic costs of a successful attack.
- Severity indicates the risk's overall criticality.

Probability and impact are classified into three categories: H, M, and L, which correspond to high, medium, and low, respectively. Severity is determined by probability and impact and is categorized into four levels, namely Critical, High, Medium, and Low.

Impact	High	Critical	High	Medium
	Medium	High	Medium	Low
	Low	Medium	Low	Low
		High	Medium	Low
			Likelihood	

2 Findings Overview

2.1 Summary

The following is a synopsis of our conclusions from our analysis of the MADNFT implementation. During the first part of our audit, we examine the smart contract source code and run the codebase via a static code analyzer. The objective here is to find known coding problems statically and then manually check (reject or confirm) issues highlighted by the tool. Additionally, we check business logics, system processes, and DeFi-related components manually to identify potential hazards and/or defects.

2.2 Key Findings

In general, these smart contracts are well-designed and constructed, but their implementation might be improved by addressing the discovered flaws, which include , 6 high-severity, 2 medium-severity, 10 low-severity vulnerabilities.

Vulnerabilities	Severity	Status
Wrong set for outbid variable	HIGH	Not Fixed
Wrong Implementation	HIGH	Not Fixed
Update Settings	HIGH	Not Fixed
Wrong set for outbid variable	HIGH	Not Fixed
Wrong Implementation	HIGH	Not Fixed
Update Settings	HIGH	Not Fixed
Maxfeemint should be limited	MEDIUM	Not Fixed
Maxfeemint should be limited	MEDIUM	Not Fixed
Pausing in withdraw	LOW	Not Fixed
Incorrect initialization	LOW	Not Fixed
Pausing in withdraw	LOW	Not Fixed
Incorrect initialization	LOW	Not Fixed
Factory addresss verification	LOW	Not Fixed
Unnecessary modifier implementation	LOW	Not Fixed

Usage of block.timestamp in deadlin	LOW	Not Fixed
Factory addresss verification	LOW	Not Fixed
Unnecessary modifier implementation	LOW	Not Fixed
Usage of block.timestamp in deadlin	LOW	Not Fixed

3 Finding Details

A MADRouter721.sol

A.1 Maxfeemint should be limited [MEDIUM]

Description:

The variable maxFeemint in the smart contract does not have any maximum value conditions attached to it. This means that the owner of the contract can potentially set the feeMint value to an excessively high amount, resulting in a disproportionate fee for users of the contract. As a result.

Code:

Listing 1: MADRouter721

```
    constructor(
        FactoryVerifier _factory,
        address _paymentTokenAddress,
        address _recipient,
        uint256 _maxFeeMint,
        uint256 _maxFeeBurnt
    ) {
        require(_maxFeeMint > 0 && _maxFeeBurnt > 0, "Invalid max fee
            ↪ settings");
        maxFeeMint = _maxFeeMint;
        maxFeeBurn = _maxFeeBurnt;
        MADFactory721 = _factory;
        if (_paymentTokenAddress != address(0)) {
            _setPaymentToken(_paymentTokenAddress);
        }
        setRecipient(_recipient);
    }
```

Risk Level:

Likelihood – 3
Impact - 3

Recommendation:

We recommend adding a max value condition to the maxFeemint variable or fixing the value as a constant in the contract to prevent excessive feeMint amounts. By implementing a max value condition, the contract will be more secure and will ensure fair fees for users of the contract.

Status - Not Fixed

A.2 Pausing in withdraw [LOW]

Description:

The ability of the owner to pause withdrawals poses a centralization risk on the user's side.

Code:

Listing 2: MADRouter721

```
    function withdraw(address _token, ERC20 _erc20)
        external
        nonReentrant
        whenNotPaused
    {
        (bytes32 _colID, uint8 _tokenType) = _tokenRender(
            _token
        );

        if (_tokenType < 1) {
            address(_erc20) != address(0) &&
                _erc20.balanceOf(_token) != 0
                ? ERC721Minimal(_token).withdrawERC20(_erc20, recipient)
                : _token.balance != 0
                ? ERC721Minimal(_token).withdraw(recipient)
                : revert("NO_FUNDS");

            emit TokenFundsWithdrawn(
                _colID,
                _tokenType,
                msg.sender
            );
        }
```

Risk Level:

Likelihood – 2
Impact - 2

Recommendation:

We recommend removing the whenNotPaused modifier from the function.

Status - Not Fixed

A.3 Incorrect initialization [LOW]

Description:

The feeMint is initialized at 0.25 ether, which can potentially harm users if the fee value is not updated on a chain like Ethereum.

Code:

Listing 3: MADRouter721

```
    uint256 public feeMint = 0.25 ether;
```

Risk Level:

Likelihood – 1
Impact - 2

Recommendation:

We recommend initializing the value of the fees in the constructor.

Status - Not Fixed

B MADRouter1155.sol

B.1 Maxfeemint should be limited [MEDIUM]

Description:

The variable maxFeemint in the smart contract does not have any maximum value conditions attached to it. This means that the owner of the contract can potentially set the feeMint value to an excessively high amount, resulting in a disproportionate fee for users of the contract. As a result.

Code:

Listing 4: MADRouter1155

```
    constructor(
        FactoryVerifier _factory,
        address _paymentTokenAddress,
        address _recipient,
        uint256 _maxFeeMint,
        uint256 _maxFeeBurnt
    ) {
        require(_maxFeeMint > 0 && _maxFeeBurnt > 0, "Invalid max fee
            ↪ settings");
        maxFeeMint = _maxFeeMint;
        maxFeeBurn = _maxFeeBurnt;
        MADFactory1155 = _factory;
        if (_paymentTokenAddress != address(0)) {
            _setPaymentToken(_paymentTokenAddress);
        }
        setRecipient(_recipient);
    }
```

Risk Level:

Likelihood – 3
Impact - 3

Recommendation:

We recommend adding a max value condition to the maxFeemint variable or fixing the value as a constant in the contract to prevent excessive feeMint amounts. By implementing a max value condition, the contract will be more secure and will ensure fair fees for users of the contract.

Status - Not Fixed

B.2 Pausing in withdraw [LOW]

Description:

The ability of the owner to pause withdrawals poses a centralization risk on the user's side.

Code:

Listing 5: MADRouter1155

```
    function withdraw(address _token, ERC20 _erc20)
        external
        nonReentrant
        whenNotPaused
    {
        (bytes32 _colID, uint8 _tokenType) = _tokenRender(
            _token
        );

        if (_tokenType < 1) {
            address(_erc20) != address(0) &&
                _erc20.balanceOf(_token) != 0
                ? ERC1155Minimal(_token).withdrawERC20(_erc20, recipient)
                : _token.balance != 0
                ? ERC1155Minimal(_token).withdraw(recipient)
                : revert("NO_FUNDS");

            emit TokenFundsWithdrawn(
                _colID,
                _tokenType,
                msg.sender
            );
        }

        if (_tokenType == 1) {
            address(_erc20) != address(0) &&
                _erc20.balanceOf(_token) != 0
                ? ERC1155Basic(_token).withdrawERC20(_erc20, recipient)
                : _token.balance != 0
                ? ERC1155Basic(_token).withdraw(recipient)
                : revert("NO_FUNDS");

            emit TokenFundsWithdrawn(
                _colID,
                _tokenType,
                msg.sender
            );
        }
        if (_tokenType == 2) {
            address(_erc20) != address(0) &&
                _erc20.balanceOf(_token) != 0
                ? ERC1155Whitelist(_token).withdrawERC20(_erc20, recipient)
                : _token.balance != 0
                ? ERC1155Whitelist(_token).withdraw(recipient)
                : revert("NO_FUNDS");

            emit TokenFundsWithdrawn(
                _colID,
                _tokenType,
                msg.sender
            );
        }

        if (_tokenType > 2) {
            address(_erc20) != address(0) &&
                _erc20.balanceOf(_token) != 0
                ? ERC1155Lazy(_token).withdrawERC20(_erc20, recipient)
                : _token.balance != 0
                ? ERC1155Lazy(_token).withdraw(recipient)
                : revert("NO_FUNDS");

            emit TokenFundsWithdrawn(
                _colID,
                _tokenType,
                msg.sender
            );
        }
    }
```

Risk Level:

Likelihood – 2
Impact - 2

Recommendation:

We recommend removing the whenNotPaused modifier from the function.

Status - Not Fixed

B.3 Incorrect initialization [LOW]

Description:

The feeMint is initialized at 0.25 ether, which can potentially harm users if the fee value is not updated on a chain like Ethereum.

Code:

Listing 6: MADRouter1155

```
    uint256 public feeMint = 0.25 ether;
```

Risk Level:

Likelihood – 1
Impact - 2

Recommendation:

We recommend initializing the value of the fees in the constructor.

Status - Not Fixed

C MADMarketplace721.sol

C.1 Wrong set for outbid variable [HIGH]

Description:

The outbid variable is set to 0 instead of userOutbid[users[i]].

Code:

Listing 7: MADMarketplace721

```
    function autoTransferFunds(address[] memory users)
        external
        onlyOwner
    {
        require(users.length < 20 && users.length > 0, "invalid user length");
        if (address(erc20) == address(0)) {
            for (uint256 i = 0; i < users.length; ++i) {
                if (userOutbid[users[i]] > 0) {
                    uint256 outbid = 0;
                    userOutbid[users[i]] = 0;
                    totalOutbid = totalOutbid - outbid;
                    SafeTransferLib.safeTransferETH(
                        msg.sender,
                        outbid
                    );
                }
                else {
                    revert("nothing to withdraw");
                }
            }
        } else {
            for (uint256 i = 0; i < users.length; i++) {
                _withdrawOutbid(users[i], erc20, 0, 0);
            }
        }
    }
```

Risk Level:

Likelihood – 4
Impact - 3

Recommendation:

We recommend assigning the value of userOutbid[users[i]] to the outbid variable.

Status - Not Fixed

C.2 Wrong Implementation [HIGH]

Description:

If the '_token' argument in the withdrawERC20 function is set to a token other than the ERC20 token payment, you cannot withdraw all the tokens, as doing so would be blocked by the 'require' statement _token.balanceOf(address(this)) - totalOutbid > 0.

Code:

Listing 8: MADMarketplace721

```
    function withdrawERC20(ERC20 _token)
        external
        onlyOwner
        whenPaused
    {
        require(_token.balanceOf(address(this)) - totalOutbid > 0, "No
           ↪ balance to withdraw");
        SafeTransferLib.safeTransfer(
            _token,
            msg.sender,
            // withdraw all except amount users have pending in contract
                ↪ (outbid)
           _token.balanceOf(address(this)) - totalOutbid
        );
    }
```

Risk Level:

Likelihood – 4
Impact - 3

Recommendation:

We recommend either withdrawing only the ERC20 payment token or adding a condition to the withdraw function that allows for withdrawing all token balances if the token is not the ERC20 payment token.

Status - Not Fixed

C.3 Update Settings [HIGH]

Description:

Firstly, _minOrderDuration should be greater than or equal to 600, not the opposite. Secondly, in the 'require' statement, the presence of an 'or' means that the owner could choose the second condition and neglect the first one.

Code:

Listing 9: MADMarketplace721

```
    function updateSettings(
        uint256 _minAuctionIncrement,
        uint256 _minOrderDuration,
        uint256 _minBidValue,
        uint256 _maxOrderDuration
    ) public onlyOwner {
        // minOrderDuration = _minOrderDuration;
        // minAuctionIncrement = _minAuctionIncrement;
        // minBidValue = _minBidValue;
        // maxOrderDuration = _maxOrderDuration;
        require(
            (_minAuctionIncrement <= 1200 &&
                _minOrderDuration <= 600 &&
                _minBidValue > 0)
                _maxOrderDuration >= _minOrderDuration,
            "Invalid Settings"
        );

        assembly {
            sstore(minOrderDuration.slot, _minOrderDuration)
            sstore(
                minAuctionIncrement.slot,
                _minAuctionIncrement
            )
            sstore(minBidValue.slot, _minBidValue)
            sstore(maxOrderDuration.slot, _maxOrderDuration)
        }

        emit AuctionSettingsUpdated(
            _minOrderDuration,
            _minAuctionIncrement,
            _minBidValue,
            _maxOrderDuration
        );
    }
```

Risk Level:

Likelihood – 3
Impact - 3

Recommendation:

We recommend setting a minimum for _minOrderDuration and a maximum for _maxOrderDuration, and replacing the 'or' with 'and'.

Status - Not Fixed

C.4 Factory addresss verification [LOW]

Description:

The address-type argument _factory should include a zero-address test, otherwise, the contract's functionality may become inaccessible.

Code:

Listing 10: MADMarketplace721

```
    function setFactory(FactoryVerifier _factory)
        public
        onlyOwner
    {
        assembly {
            // MADFactory721 = _factory;
            sstore(MADFactory721.slot, _factory)
        }
        emit FactoryUpdated(_factory);
    }
```

Risk Level:

Likelihood – 2
Impact - 2

Recommendation:

We recommend that you make sure the address provided in the argument is different from the address(0) by adding a require statement.

Status - Not Fixed

C.5 Unnecessary modifier implementation [LOW]

Description:

The owner's withdraw function does not require the contract to be paused in order to withdraw the funds

Code:

Listing 11: MADMarketplace721

```
    function withdraw() external onlyOwner whenPaused {
        require(address(this).balance - totalOutbid > 0, "No balance to
            ↪ withdraw");

        SafeTransferLib.safeTransferETH(
            msg.sender,
            // withdraw all except amount users have pending in contract
                ↪ (outbid)
            address(this).balance - totalOutbid
        );
    }

    /// @dev withdraw all ERC20 token value from contract
    function withdrawERC20(ERC20 _token)
        external
        onlyOwner
        whenPaused
    {
        require(_token.balanceOf(address(this)) - totalOutbid > 0, "No balance to withdraw");

        SafeTransferLib.safeTransfer(
            _token,
            msg.sender,
            // withdraw all except amount users have pending in contract (outbid)
           _token.balanceOf(address(this)) - totalOutbid
        );
    }
```

Risk Level:

Likelihood – 2
Impact - 2

Recommendation:

We recommend removing the whenPaused modifier from the withdraw and withdrawERC20 functions.

Status - Not Fixed

C.6 Usage of block.timestamp in deadlin [LOW]

Description:

Setting the deadline to block.timestamp in a swap is not recommended because it creates a risk that the transaction will not be included in a block before the deadline, which could result in the swap failing.

Code:

Listing 12: MADMarketplace721

```
        ISwapRouter.ExactInputSingleParams
            memory params = ISwapRouter
                .ExactInputSingleParams({
                    tokenIn: address(erc20),
                    tokenOut: address(_token),
                    fee: feeTier,
                    recipient: _sender,
                    deadline: block.timestamp,
                    amountIn: amountIn,
                    amountOutMinimum: minOut,
                    sqrtPriceLimitX96: priceLimit
                });
```

Risk Level:

Likelihood – 1
Impact - 2

Recommendation:

A better approach is to use a deadline that is a fixed amount of time in the future, rather than relying on the current block timestamp. This approach ensures that the deadline is consistent and independent of the current block timestamp.

Status - Not Fixed

D MADMarketplace1155.sol

D.1 Wrong set for outbid variable [HIGH]

Description:

The outbid variable is set to 0 instead of userOutbid[users[i]].

Code:

Listing 13: MADMarketplace1155

```
    function autoTransferFunds(address[] memory users)
        external
        onlyOwner
    {
        require(users.length < 20 && users.length > 0, "invalid user
            ↪ length");
        if (address(erc20) == address(0)) {
            for (uint256 i = 0; i < users.length; ++i) {
                if (userOutbid[users[i]] > 0) {
                    uint256 outbid = 0;
                    userOutbid[users[i]] = 0;
                    totalOutbid = totalOutbid - outbid;
                    SafeTransferLib.safeTransferETH(
                        msg.sender,
                        outbid
                    );
                }
                else {
                    revert("nothing to withdraw");
                }
            }
        } else {
            for (uint256 i = 0; i < users.length; i++) {
                _withdrawOutbid(users[i], erc20, 0, 0);
            }
        }
    }
```

Risk Level:

Likelihood – 4
Impact - 3

Recommendation:

We recommend assigning the value of userOutbid[users[i]] to the outbid variable.

Status - Not Fixed

D.2 Wrong Implementation [HIGH]

Description:

If the '_token' argument in the withdrawERC20 function is set to a token other than the ERC20 token payment, you cannot withdraw all the tokens, as doing so would be blocked by the 'require' statement _token.balanceOf(address(this)) - totalOutbid > 0.

Code:

Listing 14: MADMarketplace1155

```
    function withdrawERC20(ERC20 _token)
        external
        onlyOwner
        whenPaused
    {
        require(_token.balanceOf(address(this)) - totalOutbid > 0, "No
            ↪ balance to withdraw");
        SafeTransferLib.safeTransfer(
            _token,
            msg.sender,
            _token.balanceOf(address(this)) - totalOutbid
        );
    }
```

Risk Level:

Likelihood – 4
Impact - 3

Recommendation:

We recommend either withdrawing only the ERC20 payment token or adding a condition to the withdraw function that allows for withdrawing all token balances if the token is not the ERC20 payment token.

Status - Not Fixed

D.3 Update Settings [HIGH]

Description:

Firstly, _minOrderDuration should be greater than or equal to 600, not the opposite. Secondly, in the 'require' statement, the presence of an 'or' means that the owner could choose the second condition and neglect the first one.

Code:

Listing 15: MADMarketplace1155

```
    function updateSettings(
        uint256 _minAuctionIncrement,
        uint256 _minOrderDuration,
        uint256 _minBidValue,
        uint256 _maxOrderDuration
    ) public onlyOwner {
        // minOrderDuration = _minOrderDuration;
        // minAuctionIncrement = _minAuctionIncrement;
        // minBidValue = _minBidValue;
        // maxOrderDuration = _maxOrderDuration;
        require(
            (_minAuctionIncrement <= 1200 &&
                _minOrderDuration <= 600 &&
                _minBidValue > 0)
                _maxOrderDuration >= _minOrderDuration,
            "Invalid Settings"
        );

        assembly {
            sstore(minOrderDuration.slot, _minOrderDuration)
            sstore(
                minAuctionIncrement.slot,
                _minAuctionIncrement
            )
            sstore(minBidValue.slot, _minBidValue)
            sstore(maxOrderDuration.slot, _maxOrderDuration)
        }

        emit AuctionSettingsUpdated(
            _minOrderDuration,
            _minAuctionIncrement,
            _minBidValue,
            _maxOrderDuration
        );
    }
```

Risk Level:

Likelihood – 3
Impact - 3

Recommendation:

We recommend setting a minimum for _minOrderDuration and a maximum for _maxOrderDuration, and replacing the 'or' with 'and'.

Status - Not Fixed

D.4 Factory addresss verification [LOW]

Description:

The address-type argument _factory should include a zero-address test, otherwise, the contract's functionality may become inaccessible.

Code:

Listing 16: MADMarketplace1155

```
    function setFactory(FactoryVerifier _factory)
        public
        onlyOwner
    {
        assembly {
            // MADFactory721 = _factory;
            sstore(MADFactory721.slot, _factory)
        }
        emit FactoryUpdated(_factory);
    }
```

Risk Level:

Likelihood – 2
Impact - 2

Recommendation:

We recommend that you make sure the address provided in the argument is different than address(0) by adding a require statement.

Status - Not Fixed

D.5 Unnecessary modifier implementation [LOW]

Description:

The owner's withdraw function does not require the contract to be paused in order to withdraw the funds

Code:

Listing 17: MADMarketplace1155

```
    function withdraw() external onlyOwner whenPaused {
        require(address(this).balance - totalOutbid > 0, "No balance to withdraw");

        SafeTransferLib.safeTransferETH(
            msg.sender,
            // withdraw all except amount users have pending in contract (outbid)
            address(this).balance - totalOutbid
        );
    }

    /// @dev withdraw all ERC20 token value from contract
    function withdrawERC20(ERC20 _token)
        external
        onlyOwner
        whenPaused
    {
        require(_token.balanceOf(address(this)) - totalOutbid > 0, "No
            ↪ balance to withdraw");

        SafeTransferLib.safeTransfer(
            _token,
            msg.sender,
            // withdraw all except amount users have pending in contract
                ↪ (outbid)
           _token.balanceOf(address(this)) - totalOutbid
        );
    }
```

Risk Level:

Likelihood – 2
Impact - 2

Recommendation:

We recommend removing the whenPaused modifier from the withdraw and withdrawERC20 functions.

Status - Not Fixed

D.6 Usage of block.timestamp in deadlin [LOW]

Description:

Setting the deadline to block.timestamp in a swap is not recommended because it creates a risk that the transaction will not be included in a block before the deadline, which could result in the swap failing.

Code:

Listing 18: MADMarketplace1155

```
        ISwapRouter.ExactInputSingleParams
            memory params = ISwapRouter
                .ExactInputSingleParams({
                    tokenIn: address(erc20),
                    tokenOut: address(_token),
                    fee: feeTier,
                    recipient: _sender,
                    deadline: block.timestamp,
                    amountIn: amountIn,
                    amountOutMinimum: minOut,
                    sqrtPriceLimitX96: priceLimit
                });
```

Risk Level:

Likelihood – 1
Impact - 2

Recommendation:

A better approach is to use a deadline that is a fixed amount of time in the future, rather than relying on the current block timestamp. This approach ensures that the deadline is consistent and independent of the current block timestamp.

Status - Not Fixed

E MADFactory721.sol

E.1 Missing address verification [MEDIUM]

Description:

The address-type arguments newOwner _market _router _signer should include a zero-address test, otherwise, the contract's functionality may become inaccessible. If the contract ownership is lost. You need to re-deploy the same contract again.

Code:

Listing 19: MADFactory721

```
function setOwner(address newOwner)
    public
    override
    onlyOwner
{
    // owner = newOwner;
    assembly {
        sstore(owner.slot, newOwner)
    }

    emit OwnerUpdated(msg.sender, newOwner);
}
```

Listing 20: MADFactory721

```
function setMarket(address _market) public onlyOwner {
    assembly {
        sstore(market.slot, _market)
    }

    emit MarketplaceUpdated(_market);
}
```

Listing 21: MADFactory721

```
function setRouter(address _router) public onlyOwner {
   // router = _router;
   assembly {
     sstore(router.slot, _router)
   }

   emit RouterUpdated(_router);
 }
```

Listing 22: MADFactory721

```
function setSigner(address _signer) public onlyOwner {
  // signer = _signer;
  assembly {
    sstore(signer.slot, _signer)
  }

  emit SignerUpdated(_signer);
}
```

Listing 23: MADFactory721

```
constructor
(
  address _marketplace,
  address _router,
  address _signer
)
{
  setMarket(_marketplace);
  setRouter(_router);
  setSigner(_signer);
}
```

Risk Level:

Likelihood – 1
Impact - 4

Recommendation:

We recommend that you make sure the addresses provided in the arguments are different from the address(0).

Status - Fixed

The MAD team has fixed the issue by adding require statements to make sure that the addresses provided in the arguments are different from the address(0).

F MADFactory1155.sol

F.1 Missing address verification [MEDIUM]

Description:

The address-type arguments newOwner _market _router _signer should include a zero-address test, otherwise, the contract's functionality may become inaccessible. If the contract ownership is lost. You need to re-deploy the same contract again.

Code:

Listing 24: MADFactory1155

```
    function setOwner(address newOwner)
        public
        override
        onlyOwner
    {
        // owner = newOwner;
        assembly {
            sstore(owner.slot, newOwner)
        }

        emit OwnerUpdated(msg.sender, newOwner);
    }
```

Listing 25: MADFactory1155

```
   function setMarket(address _market) public onlyOwner {
        assembly {
            sstore(market.slot, _market)
        }

        emit MarketplaceUpdated(_market);
    }
```

Listing 26: MADFactory1155

```
  function setRouter(address _router) public onlyOwner {
        // router = _router;
        assembly {
            sstore(router.slot, _router)
        }

        emit RouterUpdated(_router);
    }
```

Listing 27: MADFactory1155

```
    function setSigner(address _signer) public onlyOwner {
        // signer = _signer;
        assembly {
            sstore(signer.slot, _signer)
        }

        emit SignerUpdated(_signer);
    }
```

Listing 28: MADFactory1155

```
    constructor
    (
        address _marketplace,
        address _router,
        address _signer
    )
    {
        setMarket(_marketplace);
        setRouter(_router);
        setSigner(_signer);
    }
```

Risk Level:

Likelihood – 1

Impact - 4

Recommendation:

We recommend that you make sure the addresses provided in the arguments are different from the address(0).

Status - Fixed

The MAD team has fixed the issue by adding require statements to make sure that the addresses provided in the arguments are different from the address(0).

4 Best Practices

BP.1 Missing token pair check

Description:

In the '_withdrawOutbid' function, if the '_token' argument is not the ERC20 payment token, the function calls the 'swap' function. However, if the '_token' entered by the user has no token pair, the transaction will fail.

Code:

Listing 29: MADMarketplace721

```
    function _withdrawOutbid(
        address _sender,
        ERC20 _token,
        uint256 minOut,
        uint160 priceLimit
    ) private {
        require(
            address(erc20) != address(0) &&
                address(_token) != address(0),
            "not erc20"
        );
        require(
            userOutbid[_sender] > 0,
            "nothing to withdraw"
        );

        uint256 amountIn = userOutbid[_sender];
        userOutbid[_sender] = 0;
        totalOutbid -= amountIn;

        if (_token == erc20) { //
```

```
            SafeTransferLib.safeTransfer(
                _token,
                _sender,
                amountIn
            );
            emit WithdrawOutbid(_sender, address(_token), amountIn); // amount withdrawn
            return;
        }
```

Code:

Listing 30: MADMarketplace1144

```
    function _withdrawOutbid(
        address _sender,
        ERC20 _token,
        uint256 minOut,
        uint160 priceLimit
    ) private {
        require(
            address(erc20) != address(0) &&
                address(_token) != address(0),
            "not erc20"
        );
        require(
            userOutbid[_sender] > 0,
            "nothing to withdraw"
        );

        uint256 amountIn = userOutbid[_sender];
        userOutbid[_sender] = 0;
        totalOutbid -= amountIn;

        if (_token == erc20) { //
```

```
            SafeTransferLib.safeTransfer(
                _token,
                _sender,
                amountIn
            );
            emit WithdrawOutbid(_sender, address(_token), amountIn); // amount withdrawn
            return;
        }
```

5 Tests

Results:

```
Downloading compiler 0.8.16
Compiled 48 Solidity files successfully
/// ... .. .. ..
/// x*8888x.:*8888: -"888: dF
/// X 48888X `8888H 8888 '88bu.
/// X8x. 8888X 8888X !888> u '*88888bu
/// X8888 X8888 88888 "*8%- us888u. ^"*8888N
/// '*888!X8888> X8888 xH8> .@88 "8888" beWE "888L
/// `?8 `8888 X888X X888> 9888 9888 888E 888E
/// -^ '888" X888 8888> 9888 9888 888E 888E
/// dx '88~x. !88~ 8888> 9888 9888 888E 888F
/// .8888Xf.888x:! X888X.: 9888 9888 .888N..888
/// :""888":~"888" `888*" "888*""888" `"888*""
/// "~' "~ "" ^Y" ^Y' "" MADNFTs © 2022.



  ERC1155Basic
    Init
      Splitter and ERC1155 should initialize (147ms)
      accounts have been funded
    Only owner setters
      Should set base URI, emit event and revert if not owner (118ms)
      Should set public mint state, emit event & revert if not owner (79
         ↪ ms)
    Mint
      Should revert if public mint is turned off (38ms)
      Should revert if max supply has reached max (5272ms)
      Should revert if price is wrong (44ms)
      Should mint, update storage and emit events (82ms)
      Should handle multiple mints (4769ms)
    Batch mint
      Should revert if supply has reached max (5284ms)
      Should revert if public mint is turned off
      Should revert if price is wrong (38ms)
      Should batch mint, update storage and emit events (111ms)
      Should handle multiple batch mints (210ms)
    Burn
      Should revert if not owner
      Should revert if id is already burnt/hasn't been minted (125ms)
      Should revert if ids length is less than 2 (42ms)
      Should burn tokens, update storage and emit event (189ms)
    Batch burn
      Should revert if caller is not the owner (68ms)
      Should revert if id is already burnt/hasn't been minted (101ms)
      Should batch burn tokens, update storage and emit event (190ms)
      Should handle multiple batch burns (328ms)
    Withdraw
      Should withdraw contract's funds (170ms)
      Should withdraw contract's ERC20s (213ms)
    Public getters
      Should query royalty info
      Should query token uri and revert if not yet minted (85ms)
      Should query total supply
      Should query base uri
    Interface IDs
      Should support interfaces (44ms)

  ERC1155Lazy
    Init
      Splitter and ERC1155 should initialize (64ms)
      accounts have been funded
    Lazy mint
      Should mint, update storage and emit events (378ms)
      Should revert if voucher has already been used (232ms)
```

```
      Should revert if signature is invalid (38ms)
      Should revert if price is wrong
    Lazy batch mint
      Should mint, update storage and emit events (148ms)
      Should revert if voucherId has already been used (83ms)
      Should revert if signature is invalid
      Should revert if price is wrong
    Only owner functions
      Should set URI and emit event (54ms)
      Should withdraw and update balances (523ms)
    Burn
      Should revert if not owner
      Should revert if id is already burnt/hasn't been minted (247ms)
      Should revert if ids length is less than 2 (52ms)
      Should burn update storage and emit events (284ms)
    Batch burn
      Should revert if caller is not the owner (211ms)
      Should revert if id is already burnt/hasn't been minted (212ms)
      Should batch burn tokens, update storage and emit event (268ms)
      Should handle multiple batch burns (413ms)
    Public getters
      Should query royalty info
      Should retrieve the domain separator
      Should retrive URI and total supply (293ms)
      Should retrive tokenURI and revert if not yet minted (204ms)
      Should support interfaces (40ms)

  ERC1155Minimal
    Init
      Splitter and ERC1155 should initialize (57ms)
      accounts have been funded
    Safe Minting
      Should revert if not the owner
      Should mint, update storage and emit events (46ms)
      Should revert if already minted (62ms)
    Burning
      Should revert if has not been minted
      Should revert if not the owner (50ms)
      Should burn, update storage and emit events (85ms)
      Should revert if already burned (83ms)
    Public Minting
      Should update public mint state (48ms)
      Should revert if public mint is off
      Should revert if price is wrong (50ms)
      Should revert if already minted (72ms)
      Should mint, update storage and emit events (77ms)
    Withdrawing
      Should revert if not the owner (75ms)
      Should update balances of contract and owner (132ms)
      Should withdraw contract's ERC20s (214ms)
    Royalties
      Should retrive royalty info
    Token URI
      Should revert if ID is not 1
      Should revert if token was not minted
      Should retrieve tokenURI (47ms)
    Interface IDs
      Should support interfaces (41ms)

  ERC1155Whitelist
    Init
      Splitter and ERC721 should initialize (136ms)
      accounts have been funded
    Only owner setters
      Should check for whitelist & freeclaim event emitting/error
         ↪ handling (100ms)
      Should set URI and emit event (60ms)
      Should set mint states (105ms)
    Public mint
      Should revert if value under/overflows
      Should revert if public mint state is off
      Should revert if available supply has reached max (5747ms)
      Should revert if price is wrong (42ms)
      Should mint, update storage and emit events (118ms)
    Batch mint
      Should revert if supply has reached max (5686ms)
      Should revert if public mint is turned off
      Should revert if price is wrong (53ms)
      Should batch mint, update storage and emit events (104ms)
      Should handle multiple batch mints (206ms)
    Whitelist mint
      Should revert if value under/overflows
      Should revert if whitelist mint state is off
      Should revert if whitelist supply has reached max (6370ms)
      Should revert if price is wrong (43ms)
      Should revert if address is not whitelisted (46ms)
      Should mint, update storage and emit events (133ms)
    Whitelist batch mint
      Should revert if value under/overflows
      Should revert if whitelist mint state is off
      Should revert if whitelist supply has reached max (6270ms)
      Should revert if price is wrong (51ms)
      Should revert if address is not whitelisted (42ms)
      Should mint, update storage and emit events (134ms)
    Free claim
      Should revert if free claim state is off
      Should revert if available supply has reached max (6416ms)
      Should revert if address is not whitelisted (39ms)
      Should revert if user has already claimed (73ms)
      Should mint, update storage and emit events (97ms)
      Should gift tokens (219ms)
    Mint and batch mint to creator
```

```
       Should mint to creator (164ms)
       Should batch mint to creator (180ms)
    Burn
       Should revert if not owner
       Should revert if id is already burnt/hasn't been minted (115ms)
       Should revert if ids length is less than 2
       Should burn tokens, update storage and emit event (187ms)
    Batch burn
       Should revert if caller is not the owner (80ms)
       Should revert if id is already burnt/hasn't been minted (96ms)
       Should batch burn tokens, update storage and emit event (200ms)
       Should handle multiple batch burns (346ms)
    Withdraw
       Should withdraw contract's funds (165ms)
       Should withdraw contract's ERC20s (204ms)
    Public getters
       Should query royalty info
       Should query token uri and revert if not yet minted (80ms)
       Should query total supply
       Should query base uri
    Interface IDs
       Should support interfaces (43ms)

  ERC721Basic
    Init
       Splitter and ERC721 should initialize (84ms)
       accounts have been funded
    Only owner setters
       Should set base URI, emit event and revert if not owner (73ms)
       Should set public mint state, emit event & revert if not owner (60
          ↪ ms)
    Mint
       Should revert if public mint is turned off
       Should revert if max supply has reached max (6721ms)
```

```
      Should revert if price is wrong (47ms)
      Should mint, update storage and emit events (84ms)
      Should handle multiple mints (6393ms)
    Burn
      Should revert if not owner
      Should revert if id is already burnt/hasn't been minted (105ms)
      Should revert if ids length is less than 2
      Should burn tokens, update storage and emit event (202ms)
    Withdraw
      Should withdraw contract's funds (146ms)
      Should withdraw contract's ERC20s (208ms)
    Public getters
      Should query royalty info
      Should query token uri and revert if not yet minted (78ms)
      Should query total supply
      Should query base uri
      Should support interfaces (45ms)

  ERC721Lazy
    Init
      Splitter and ERC721 should initialize (78ms)
      accounts have been funded
    Lazy mint
      Should mint, update storage and emit events (354ms)
      Should revert if voucher has already been used (224ms)
      Should revert if signature is invalid
      Should revert if price is wrong
    Only owner functions
      Should set baseURI and emit event (53ms)
      Should withdraw and update balances (486ms)
    Burn
      Should revert if not owner
      Should revert if id is already burnt/hasn't been minted (214ms)
      Should revert if ids length is less than 2
```

```
        Should burn update storage and emit events (297ms)
      Public getters
        Should retrieve the domain separator
        Should retrive baseURI and total supply (270ms)
        Should retrive tokenURI and revert if not yet minted (210ms)
        Should query royalty info
        Should support interfaces

    ERC721Minimal
      Init
        Splitter and ERC721 should initialize (79ms)
        accounts have been funded
      Safe Minting
        Should revert if not the owner
        Should mint, update storage and emit events (52ms)
        Should revert if already minted (48ms)
      Burning
        Should revert if has not been minted
        Should revert if not the owner (51ms)
        Should burn, update storage and emit events (78ms)
        Should revert if already burned (66ms)
      Public Minting
        Should update public mint state (53ms)
        Should revert if public mint is off
        Should revert if price is wrong (49ms)
        Should revert if already minted (92ms)
        Should mint, update storage and emit events (71ms)
      Withdrawing
        Should revert if not the owner (73ms)
        Should update balances of contract and owner (120ms)
        Should withdraw contract's ERC20s (209ms)
      Royalties
        Should retrive royalty info
      Token URI
```

```
       Should revert if ID is not 1
       Should revert if token was not minted
       Should retrieve tokenURI (51ms)
    Interface IDs
       Should support interfaces

  ERC721Whitelist
    Init
       Splitter and ERC721 should initialize (159ms)
       accounts have been funded
    Only owner setters
       Should check for whitelist & freeclaim event emitting/error
          ↪ handling (89ms)
       Should set baseURI and emit event (39ms)
       Should set mint states (103ms)
    Public mint
       Should revert if value under/overflows
       Should revert if public mint state is off
       Should revert if available supply has reached max (5434ms)
       Should revert if price is wrong
       Should mint, update storage and emit events (124ms)
    Whitelist mint
       Should revert if value under/overflows
       Should revert if whitelist mint state is off
       Should revert if whitelist supply has reached max (6799ms)
       Should revert if price is wrong
       Should revert if address is not whitelisted (41ms)
       Should mint, update storage and emit events (133ms)
    Free claim
       Should revert if free claim state is off
       Should revert if available supply has reached max (6480ms)
       Should revert if address is not whitelisted (46ms)
       Should revert if user has already claimed (58ms)
       Should mint, update storage and emit events (124ms)
       Should mint to creator (131ms)
       Should gift tokens (237ms)
    Burn
       Should revert if not owner
       Should revert if id is already burnt/hasn't been minted (108ms)
       Should revert if ids length is less than 2
       Should burn update storage and emit events (183ms)
    Public getters
       Should retrive baseURI and total supply (136ms)
       Should retrive tokenURI and revert if not yet minted (47ms)
       Should support interfaces
    Withdrawing
       Should revert if not the owner (82ms)
       Should update balances of contract and owner (124ms)
       Should withdraw contract's ERC20s (199ms)

  MADFactory1155
    Init
       Factory should initialize
    Splitter check
       Should revert if repeated salt is provided (204ms)
       Should deploy splitter without ambassador, update storage and emit
          ↪  events (172ms)
       Should deploy splitter with ambassador, update storage and emit
          ↪ events (197ms)
    Create collection
       Should deploy ERC1155Minimal, update storage and emit events (455
          ↪ ms)
       Should deploy ERC1155Basic, update storage and emit events (485ms)
          ↪
       Should deploy ERC1155Whitelist, update storage and emit events
          ↪ (947ms)
       Should deploy ERC1155Lazy, update storage and emit events (485ms)
    Only owner functions
```

```
       Should update contract's owner (61ms)
       Should set new marketplace instance (62ms)
       Should update ERC1155Lazy signer (45ms)
       Should update router's address (44ms)
       Should initialize paused and unpaused states (111ms)
    Helpers
       Should retrieve user's colID indexes (1270ms)
       Should get collection ID from address
       Should retrieve collection type (453ms)
       Should enable marketplace no-fee listing (1009ms)
       Should verify a collection's creator (395ms)

  MADFactory721
    Init
       Factory should initialize
    Splitter check
       Should revert if repeated salt is provided (183ms)
       Should deploy splitter without ambassador, update storage and emit
          ↪  events (187ms)
       Should deploy splitter with ambassador, update storage and emit
          ↪ events (195ms)
       Should deploy splitter with ambassador and project, update storage
          ↪  and emit events (215ms)
    Create collection
       Should deploy ERC721Minimal, update storage and emit events (431ms
          ↪ )
       Should deploy ERC721Basic, update storage and emit events (693ms)
       Should deploy ERC721Whitelist, update storage and emit events (480
          ↪ ms)
       Should deploy ERC721Lazy, update storage and emit events (452ms)
    Only owner functions
       Should update contract's owner (52ms)
       Should set new marketplace instance (65ms)
       Should update ERC721Lazy signer
```

```
      Should update router's address (42ms)
      Should initialize paused and unpaused states (102ms)
    Helpers
      Should retrieve user's colID indexes (1160ms)
      Should get collection ID from address
      Should retrieve collection type (445ms)
      Should enable marketplace no-fee listing (906ms)
      Should verify a collection's creator (383ms)

  MADMarketplace1155
    Init
      Marketplace should initialize (39ms)
    Owner Functions
      Should update factory address (66ms)
      Should update marketplace settings (39ms)
      Should initialize paused and unpaused states (203ms)
      Should update recipient (42ms)
      Should update contract's owner (44ms)
      Should withdraw to owner (110ms)
      Should delete order (686ms)
    Fixed Price Listing
      Should revert if transaction approval hasn't been set (902ms)
      Should revert if duration is less than min allowed (447ms)
      Should revert if price is invalid (444ms)
      Should list fixed price order, update storage and emit event (541
         ↪ ms)
      Should handle multiple fixed price orders (1462ms)
    Dutch Auction Listing
      Should revert if transaction approval hasn't been set (492ms)
      Should revert if duration is less than min allowed (437ms)
      Should revert if startPrice is invalid (794ms)
      Should list dutch auction order, update storage and emit event
         ↪ (573ms)
      Should handle multiple dutch auction orders (1452ms)
    English Auction Listing
      Should revert if transaction approval hasn't been set (485ms)
      Should revert if duration is less than min allowed (435ms)
      Should revert if startPrice is invalid (451ms)
      Should list english auction order, update storage and emit event
         ↪ (891ms)
      Should handle multiple english auction orders (1162ms)
    Bidding
      Should revert if price is wrong (936ms)
      Should revert if not English Auction (513ms)
      Should revert if order was canceled (564ms)
      Should revert if order has timed out (515ms)
      Should revert if bidder is the seller (810ms)
      Should bid, update storage and emit events (566ms)
    Buying
      Should revert if price is wrong (495ms)
      Should revert if order is an English Auction (501ms)
      Should revert if order was canceled (948ms)
      Should revert if order has timed out (510ms)
      Should revert if token has already been sold (602ms)
      Should buy inhouse minted tokens, update storage and emit events
         ↪ (1506ms)
      Should verify inhouse minted tokens balance changes (1118ms)
      Should buy third party minted tokens with ERC2981 support (463ms)
      Should buy third party minted tokens without ERC2981 support (398
         ↪ ms)
      Should verify inhouse minted tokens balance changes - set fees
         ↪ (1404ms)
      Should buy third party minted tokens with ERC2981 support - set
         ↪ fees (482ms)
      Should buy third party minted tokens without ERC2981 support - set
         ↪  fees (431ms)
    Claim
      Should revert if caller is seller or bidder (565ms)
```

```
      Should revert if token has already been claimed (630ms)
      Should revert if orderType is not an english auction (246ms)
      Should revert if auction hasn't ended (508ms)
      Should claim inhouse minted tokens, update storage and emit events
         ↪  (1036ms)
      Should verify inhouse minted tokens balance changes (636ms)
      Should claim third party minted tokens with ERC2981 support (362ms
         ↪ )
      Should claim third party minted tokens without ERC2981 support
         ↪ (288ms)
    Order Cancelling
      Should revert due to already sold fixed price order (564ms)
      Should revert due to already sold dutch auction order (597ms)
      Should revert due to already sold english auction order (992ms)
      Should cancel fixed price order (580ms)
      Should cancel dutch auction order (567ms)
      Should cancel english auction order (572ms)
    Public Helpers
      Should fetch the length of orderIds for a token (1142ms)
      Should fetch the length of orderIds for a seller (1139ms)

  MADMarketplace721
    Init
      Marketplace should initialize (47ms)
    Owner Functions
      Should update factory address (43ms)
      Should update marketplace settings (39ms)
      Should initialize paused and unpaused states (186ms)
      Should update recipient (38ms)
      Should update contract's owner (51ms)
      Should withdraw to owner (98ms)
      Should delete order (648ms)
    Fixed Price Listing
      Should revert if transaction approval hasn't been set (524ms)
      Should revert if duration is less than min allowed (587ms)
      Should revert if price is invalid (450ms)
      Should list fixed price order, update storage and emit event (540
         ↪ ms)
      Should handle multiple fixed price orders (1123ms)
    Dutch Auction Listing
      Should revert if transaction approval hasn't been set (775ms)
      Should revert if duration is less than min allowed (441ms)
      Should revert if startPrice is invalid (449ms)
      Should list dutch auction order, update storage and emit event
         ↪ (539ms)
      Should handle multiple dutch auction orders (1488ms)
    English Auction Listing
      Should revert if transaction approval hasn't been set (486ms)
      Should revert if duration is less than min allowed (432ms)
      Should revert if startPrice is invalid (426ms)
      Should list english auction order, update storage and emit event
         ↪ (525ms)
      Should handle multiple english auction orders (1432ms)
    Bidding
      Should revert if price is wrong (553ms)
      Should revert if not English Auction (485ms)
      Should revert if order was canceled (534ms)
      Should revert if order has timed out (863ms)
      Should revert if bidder is the seller (494ms)
      Should bid, update storage and emit events (577ms)
    Buying
      Should revert if price is wrong (507ms)
      Should revert if order is an English Auction (496ms)
      Should revert if order was canceled (871ms)
      Should revert if order has timed out (489ms)
      Should revert if token has already been sold (582ms)
      Should buy inhouse minted tokens, update storage and emit events
         ↪ (1476ms)
      Should verify inhouse minted tokens balance changes (1072ms)
BigNumber { value: "347222222222222264" } BigNumber { value:
   ↪ "868055555555556" }
      Should buy third party minted tokens with ERC2981 support (520ms)
      Should buy third party minted tokens without ERC2981 support (432
         ↪ ms)
      Should verify inhouse minted tokens balance changes - fee change
         ↪ update (1091ms)
BigNumber { value: "347222222222222264" } BigNumber { value:
   ↪ "1736111111111113" }
      Should buy third party minted tokens with ERC2981 support - fee
         ↪ change update (927ms)
      Should buy third party minted tokens without ERC2981 support - fee
         ↪  change update (441ms)
    Claim
      Should revert if caller is seller or bidder (546ms)
      Should revert if token has already been claimed (624ms)
      Should revert if orderType is not an english auction (292ms)
      Should revert if auction hasn't ended (534ms)
      Should claim inhouse minted tokens, update storage and emit events
         ↪  (650ms)
      Should verify inhouse minted tokens balance changes (942ms)
      Should claim third party minted tokens with ERC2981 support (350ms
         ↪ )
      Should claim third party minted tokens without ERC2981 support
         ↪ (283ms)
    Order Cancelling
      Should revert due to already sold fixed price order (610ms)
      Should revert due to already sold dutch auction order (589ms)
      Should revert due to already sold english auction order (635ms)
      Should cancel fixed price order (593ms)
      Should cancel dutch auction order (942ms)
      Should cancel english auction order (598ms)
    Public Helpers
```

```
       Should fetch the length of orderIds for a token (769ms)
       Should fetch the length of orderIds for a seller (1164ms)

  MADRouter1155
    Init
       Router should initialize
    Set URI
       Should revert for invalid collection type (414ms)
       Should set URI for 1155Basic collection type (518ms)
       Should set URI for 1155Whitelist collection type (961ms)
       Should set URI for 1155Lazy collection type (502ms)
    Whitelist Settings
       Should revert for invalid collection type (840ms)
       Should set whitelist config for 1155Whitelist collection type (519
          ↪ ms)
    FreeClaim Settings
       Should revert for invalid collection type (775ms)
       Should set freeClaim config for 1155Whitelist collection type (502
          ↪ ms)
    Minimal SafeMint
       Should revert for invalid collection type (831ms)
(node:2115) PromiseRejectionHandledWarning: Promise rejection was
   ↪ handled asynchronously (rejection id: 14)
(Use `node --trace-warnings ...` to show where the warning was created)
       Should call safeMint for 1155Minimal collection type (440ms)
    Burn
       Should burn token for 1155Minimal collection type (467ms)
       Should burn tokens for 1155Basic collection type (556ms)
       Should burn tokens for 1155Whitelist collection type (977ms)
       Should burn tokens for 1155Lazy collection type (628ms)
    Batch Burn
       Should revert for invalid collection type (429ms)
       Should batch burn token for 1155Basic collection type (817ms)
       Should batch burn tokens for 1155Whitelist collection type (668ms)
```

```
         ↪
      Should batch burn tokens for 1155Lazy collection type (983ms)
    Set MintState
      Should revert for invalid stateType
      Should revert for invalid tokenType (357ms)
      Should set publicMintState for minimal, basic and whitelist
         ↪ colTypes (1448ms)
      Should set whitelistMintState for whitelist colType (554ms)
      Should set freeClaimState for whitelist colType (854ms)
    Whitelist Creator Mint
      Should revert for invalid coltype (425ms)
      Should mint to creator (893ms)
    Whitelist Creator Batch Mint
      Should revert for invalid coltype (426ms)
      Should batch mint to creator (411ms)
      Should mint to creator (965ms)
    Whitelist token gifting
      Should revert for invalid coltype (412ms)
      Should gift tokens (954ms)
    Creator Withdraw
      Should withdraw balance and ERC20 for all colTypes (3690ms)
    Only Owner
      Should update contract's owner (43ms)
      Should initialize paused and unpaused states (213ms)
    Minimal SafeMint
      Should call safeMint for 1155Minimal collection type (519ms)
    Burn-setfees
      Should burn token for 1155Minimal collection type (578ms)
fee is BigNumber { value: "500000000000000000" }
      Should burn tokens for 1155Basic collection type (1149ms)
      Should burn tokens for 1155Whitelist collection type (783ms)
      Should burn tokens for 1155Lazy collection type (997ms)
    Batch Burn
      Should revert for invalid collection type (468ms)
      Should batch burn token for 1155Basic collection type (681ms)
      Should batch burn tokens for 1155Whitelist collection type (1244ms
         ↪ )
      Should batch burn tokens for 1155Lazy collection type (692ms)
    Whitelist Creator Mint
      Should revert for invalid coltype (761ms)
      Should mint to creator (644ms)
    Whitelist Creator Batch Mint
      Should mint to creator (962ms)
    Whitelist token gifting
      Should gift tokens (671ms)

  MADRouter721
    Init
      Router should initialize
    Set baseURI
      Should revert for invalid collection type (407ms)
      Should set baseURI for 721Basic collection type (491ms)
      Should set baseURI for 721Whitelist collection type (504ms)
      Should set baseURI for 721Lazy collection type (858ms)
    Whitelist Settings
      Should revert for invalid collection type (408ms)
      Should set whitelist config for 721Whitelist collection type (483
         ↪ ms)
    FreeClaim Settings
      Should revert for invalid collection type (689ms)
      Should set freeClaim config for 721Whitelist collection type (459
         ↪ ms)
    Minimal SafeMint
      Should revert for invalid collection type (400ms)
(node:2115) PromiseRejectionHandledWarning: Promise rejection was
   ↪ handled asynchronously (rejection id: 15)
BigNumber { value: "25000000000000000" }
minted successfully
```

```
      Should call safeMint for 721Minimal collection type (868ms)
    Burn
      Should burn token for 721Minimal collection type (479ms)
      Should burn tokens for 721Basic collection type (485ms)
      Should burn tokens for 721Whitelist collection type (569ms)
      Should burn tokens for 721Lazy collection type (926ms)
    Set MintState
      Should revert for invalid stateType
      Should revert for invalid tokenType (360ms)
      Should set publicMintState for minimal, basic and whitelist
         ↪ colTypes (1395ms)
      Should set whitelistMintState for whitelist colType (475ms)
      Should set freeClaimState for whitelist colType (489ms)
    Whitelist Creator Mint
      Should revert for invalid coltype (718ms)
      Should mint to creator (552ms)
    Whitelist token gifting
      Should revert for invalid coltype (381ms)
      Should gift tokens (973ms)
    Creator Withdraw
      Should withdraw balance and ERC20 for all colTypes (3234ms)
    Only Owner
      Should update contract's owner (50ms)
      Should initialize paused and unpaused states (209ms)
    Minimal SafeMint-setBaseFee
      Should call safeMint for 721Minimal collection type (508ms)
    Burn-setBaseFee
      Should burn tokens for 721Basic collection type (574ms)
      Should burn tokens for 721Whitelist collection type (966ms)
      Should burn tokens for 721Lazy collection type (628ms)
    Whitelist Creator Mint-setBaseFee
      Should mint to creator (604ms)
    Whitelist token gifting-setBaseFee
      Should gift tokens (857ms)
```

```
Royalties
   Royalties should initialize
   Should retrive royalty info
   Should accept recipient and fee change (95ms)
   Should support interfaces

Splitter
  Init
    Splitter should initialize (65ms)
    accounts have been funded
  Reverts
    should revert if no payees are provided
    should revert if more payees than shares are provided (39ms)
    should revert if more shares than payees are provided
    should revert if dead address is provided as payee
    should revert if a share is set to zero
    should revert if a provided payees are duplicated
    should revert if a provided payees are duplicated (44ms)
    should revert if account has no shares to claim
    should revert if there are no funds to claim
    should revert if account has no ERC20 shares to claim (94ms)
    should revert if there is no ERC20 to claim (99ms)
  Receive Payments
    should accept value and autodistribute to payees (165ms)
    should accept ERC20 (102ms)
  Release Payments
    should release value to payee (69ms)
    should release all pending balance to payees (71ms)
    should release ERC20 to payee (168ms)


471 passing (4m)
```

6 Static Analysis (Slither)

Description:

Block Hat expanded the coverage of the specific contract areas using automated testing methodologies. Slither, a Solidity static analysis framework, was one of the tools used. Slither was run on all-scoped contracts in both text and binary formats. This tool can be used to test mathematical relationships between Solidity instances statically and variables that allow for the detection of errors or inconsistent usage of the contracts' APIs throughout the entire codebase.

Results:

```
MADFactory1155.market (contracts/MADFactory1155.sol#79) is never
    ↪ initialized. It is used in:
        - MADFactory1155.setMarket(address) (contracts/MADFactory1155.sol
            ↪ #508-515)
        - MADFactory1155._isMarket() (contracts/MADFactory1155.sol
            ↪ #752-759)
MADFactory1155.signer (contracts/MADFactory1155.sol#82) is never
    ↪ initialized. It is used in:
        - MADFactory1155.createCollection(uint8,string,string,string,
            ↪ uint256,uint256,string,address,uint256) (contracts/
            ↪ MADFactory1155.sol#330-485)
        - MADFactory1155.setSigner(address) (contracts/MADFactory1155.sol
            ↪ #531-539)
MADFactory721.market (contracts/MADFactory721.sol#79) is never
    ↪ initialized. It is used in:
        - MADFactory721.setMarket(address) (contracts/MADFactory721.sol
            ↪ #520-528)
        - MADFactory721._isMarket() (contracts/MADFactory721.sol#766-773)
MADFactory721.signer (contracts/MADFactory721.sol#82) is never
    ↪ initialized. It is used in:
        - MADFactory721.createCollection(uint8,string,string,string,
           ↪ uint256,uint256,string,address,uint256) (contracts/
           ↪ MADFactory721.sol#334-497)
        - MADFactory721.setSigner(address) (contracts/MADFactory721.sol
           ↪ #544-552)
MADMarketplace1155.feeSelector (contracts/MADMarketplace1155.sol#67-68)
   ↪ is never initialized. It is used in:
        - MADMarketplace1155.buy(bytes32) (contracts/MADMarketplace1155.
           ↪ sol#281-361)
        - MADMarketplace1155.claim(bytes32) (contracts/MADMarketplace1155
           ↪ .sol#366-427)
        - MADMarketplace1155._feeResolver(uint256,uint256,uint256) (
           ↪ contracts/MADMarketplace1155.sol#1102-1126)
MADMarketplace1155.minOrderDuration (contracts/MADMarketplace1155.sol
   ↪ #70) is never initialized. It is used in:
        - MADMarketplace1155.updateSettings(uint256,uint256,uint256,
           ↪ uint256) (contracts/MADMarketplace1155.sol#507-541)
        - MADMarketplace1155._makeOrderChecks(uint256,uint256) (contracts
           ↪ /MADMarketplace1155.sol#1170-1209)
MADMarketplace1155.minAuctionIncrement (contracts/MADMarketplace1155.sol
   ↪ #71) is never initialized. It is used in:
        - MADMarketplace1155.bid(bytes32) (contracts/MADMarketplace1155.
           ↪ sol#190-276)
        - MADMarketplace1155.updateSettings(uint256,uint256,uint256,
           ↪ uint256) (contracts/MADMarketplace1155.sol#507-541)
MADMarketplace1155.minBidValue (contracts/MADMarketplace1155.sol#72) is
   ↪ never initialized. It is used in:
        - MADMarketplace1155.updateSettings(uint256,uint256,uint256,
           ↪ uint256) (contracts/MADMarketplace1155.sol#507-541)
        - MADMarketplace1155._bidChecks(uint8,uint256,address,uint256,
           ↪ uint256,uint256) (contracts/MADMarketplace1155.sol
           ↪ #1235-1291)
MADMarketplace1155.maxOrderDuration (contracts/MADMarketplace1155.sol
   ↪ #73) is never initialized. It is used in:
        - MADMarketplace1155.updateSettings(uint256,uint256,uint256,
           ↪ uint256) (contracts/MADMarketplace1155.sol#507-541)
        - MADMarketplace1155._makeOrderChecks(uint256,uint256) (contracts
           ↪ /MADMarketplace1155.sol#1170-1209)
MADMarketplace1155.recipient (contracts/MADMarketplace1155.sol#75) is
    ↪ never initialized. It is used in:
        - MADMarketplace1155.setRecipient(address) (contracts/
           ↪ MADMarketplace1155.sol#571-586)
        - MADMarketplace1155._intPath(Types.Order1155,uint256,bytes32,
           ↪ address,uint256) (contracts/MADMarketplace1155.sol
           ↪ #896-970)
        - MADMarketplace1155._extPath0(Types.Order1155,uint256,bytes32,
           ↪ address) (contracts/MADMarketplace1155.sol#972-1047)
        - MADMarketplace1155._extPath1(Types.Order1155,uint256,bytes32,
           ↪ address) (contracts/MADMarketplace1155.sol#1049-1100)
MADMarketplace1155.MADFactory1155 (contracts/MADMarketplace1155.sol#76)
    ↪ is never initialized. It is used in:
        - MADMarketplace1155.buy(bytes32) (contracts/MADMarketplace1155.
           ↪ sol#281-361)
        - MADMarketplace1155.claim(bytes32) (contracts/MADMarketplace1155
           ↪ .sol#366-427)
        - MADMarketplace1155.setFactory(FactoryVerifier) (contracts/
           ↪ MADMarketplace1155.sol#471-480)
MADMarketplace721.feeSelector (contracts/MADMarketplace721.sol#67-68) is
    ↪  never initialized. It is used in:
        - MADMarketplace721.buy(bytes32) (contracts/MADMarketplace721.sol
           ↪ #263-348)
        - MADMarketplace721.claim(bytes32) (contracts/MADMarketplace721.
           ↪ sol#353-416)
        - MADMarketplace721._feeResolver(uint256,uint256) (contracts/
           ↪ MADMarketplace721.sol#1034-1054)
MADMarketplace721.minOrderDuration (contracts/MADMarketplace721.sol#70)
    ↪ is never initialized. It is used in:
        - MADMarketplace721.updateSettings(uint256,uint256,uint256,
           ↪ uint256) (contracts/MADMarketplace721.sol#485-519)
        - MADMarketplace721._makeOrderChecks(uint256,uint256) (contracts/
           ↪ MADMarketplace721.sol#1098-1137)
MADMarketplace721.maxOrderDuration (contracts/MADMarketplace721.sol#71)
    ↪ is never initialized. It is used in:
        - MADMarketplace721.updateSettings(uint256,uint256,uint256,
           ↪ uint256) (contracts/MADMarketplace721.sol#485-519)
        - MADMarketplace721._makeOrderChecks(uint256,uint256) (contracts/
           ↪ MADMarketplace721.sol#1098-1137)
MADMarketplace721.minAuctionIncrement (contracts/MADMarketplace721.sol
    ↪ #72) is never initialized. It is used in:
        - MADMarketplace721.bid(bytes32) (contracts/MADMarketplace721.sol
           ↪ #169-258)
        - MADMarketplace721.updateSettings(uint256,uint256,uint256,
           ↪ uint256) (contracts/MADMarketplace721.sol#485-519)
MADMarketplace721.minBidValue (contracts/MADMarketplace721.sol#73) is
    ↪ never initialized. It is used in:
        - MADMarketplace721.updateSettings(uint256,uint256,uint256,
           ↪ uint256) (contracts/MADMarketplace721.sol#485-519)
        - MADMarketplace721._bidChecks(uint8,uint256,address,uint256,
           ↪ uint256,uint256) (contracts/MADMarketplace721.sol
           ↪ #1163-1219)
MADMarketplace721.recipient (contracts/MADMarketplace721.sol#75) is
    ↪ never initialized. It is used in:
        - MADMarketplace721.setRecipient(address) (contracts/
           ↪ MADMarketplace721.sol#544-559)
        - MADMarketplace721._intPath(Types.Order721,uint256,bytes32,
           ↪ address,uint256) (contracts/MADMarketplace721.sol#851-918)
        - MADMarketplace721._extPath0(Types.Order721,uint256,bytes32,
           ↪ address) (contracts/MADMarketplace721.sol#920-986)
        - MADMarketplace721._extPath1(Types.Order721,uint256,bytes32,
           ↪ address) (contracts/MADMarketplace721.sol#988-1032)
```

```
MADMarketplace721.MADFactory721 (contracts/MADMarketplace721.sol#76) is
   ↪ never initialized. It is used in:
       - MADMarketplace721.buy(bytes32) (contracts/MADMarketplace721.sol
          ↪ #263-348)
       - MADMarketplace721.claim(bytes32) (contracts/MADMarketplace721.
          ↪ sol#353-416)
       - MADMarketplace721.setFactory(FactoryVerifier) (contracts/
          ↪ MADMarketplace721.sol#451-460)
MADRouter1155.recipient (contracts/MADRouter1155.sol#50) is never
   ↪ initialized. It is used in:
       - MADRouter1155.setRecipient(address) (contracts/MADRouter1155.
          ↪ sol#107-115)
       - MADRouter1155.withdraw(address,ERC20) (contracts/MADRouter1155.
          ↪ sol#487-555)
MADRouter721.recipient (contracts/MADRouter721.sol#56) is never
   ↪ initialized. It is used in:
       - MADRouter721.withdraw(address,ERC20) (contracts/MADRouter721.
          ↪ sol#374-442)
       - MADRouter721.setRecipient(address) (contracts/MADRouter721.sol
          ↪ #577-585)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation
   ↪ #uninitialized-state-variables

MADMarketplace1155.getCurrentPrice(bytes32) (contracts/
   ↪ MADMarketplace1155.sol#1359-1425) performs a multiplication on
   ↪ the result of a division:
       -_tick_getCurrentPrice_asm_0 = _startPrice_getCurrentPrice_asm_0
          ↪ - _endPrice_getCurrentPrice_asm_0 /
          ↪ _endTime_getCurrentPrice_asm_0 -
          ↪ _startTime_getCurrentPrice_asm_0 (contracts/
          ↪ MADMarketplace1155.sol#1398-1401)
       -price = _startPrice_getCurrentPrice_asm_0 - timestamp()() -
          ↪ _startTime_getCurrentPrice_asm_0 *
          ↪ _tick_getCurrentPrice_asm_0 (contracts/MADMarketplace1155.
            ↪ sol#1402-1405)
MADMarketplace721.getCurrentPrice(bytes32) (contracts/MADMarketplace721.
    ↪ sol#1287-1353) performs a multiplication on the result of a
    ↪ division:
        -_tick_getCurrentPrice_asm_0 = _startPrice_getCurrentPrice_asm_0
            ↪ - _endPrice_getCurrentPrice_asm_0 /
            ↪ _endTime_getCurrentPrice_asm_0 -
            ↪ _startTime_getCurrentPrice_asm_0 (contracts/
            ↪ MADMarketplace721.sol#1326-1329)
        -price = _startPrice_getCurrentPrice_asm_0 - timestamp()() -
            ↪ _startTime_getCurrentPrice_asm_0 *
            ↪ _tick_getCurrentPrice_asm_0 (contracts/MADMarketplace721.
            ↪ sol#1330-1333)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation
    ↪ #divide-before-multiply

Contract locking ether found:
        Contract MADMarketplace1155 (contracts/MADMarketplace1155.sol
            ↪ #15-1452) has payable functions:
         - MADMarketplace1155.bid(bytes32) (contracts/MADMarketplace1155.
             ↪ sol#190-276)
         - MADMarketplace1155.buy(bytes32) (contracts/MADMarketplace1155.
             ↪ sol#281-361)
         - MADMarketplace1155.receive() (contracts/MADMarketplace1155.sol
             ↪ #463)
        But does not have a function to withdraw the ether
Contract locking ether found:
        Contract MADMarketplace721 (contracts/MADMarketplace721.sol
            ↪ #15-1380) has payable functions:
         - MADMarketplace721.bid(bytes32) (contracts/MADMarketplace721.
             ↪ sol#169-258)
         - MADMarketplace721.buy(bytes32) (contracts/MADMarketplace721.
             ↪ sol#263-348)
```

```
           - MADMarketplace721.receive() (contracts/MADMarketplace721.sol
              ↪ #443)
          But does not have a function to withdraw the ether
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation
   ↪ #contracts-that-lock-ether

Reentrancy in MADFactory1155.createCollection(uint8,string,string,string
   ↪ ,uint256,uint256,string,address,uint256) (contracts/
   ↪ MADFactory1155.sol#330-485):
          External calls:
          - (tokenSalt,deployed) = ERC1155MinimalDeployer.
             ↪ _1155MinimalDeploy(_tokenSalt,_uri,_price,_splitter,router
             ↪ ,_royalty,erc20) (contracts/MADFactory1155.sol#352-361)
          - (tokenSalt,deployed) = ERC1155BasicDeployer._1155BasicDeploy(
             ↪ _tokenSalt,_uri,_price,_maxSupply,_splitter,router,
             ↪ _royalty,erc20) (contracts/MADFactory1155.sol#385-395)
          State variables written after the call(s):
          - userTokens[tx.origin].push(colId_scope_2) (contracts/
             ↪ MADFactory1155.sol#398)
Reentrancy in MADFactory1155.createCollection(uint8,string,string,string
   ↪ ,uint256,uint256,string,address,uint256) (contracts/
   ↪ MADFactory1155.sol#330-485):
          External calls:
          - (tokenSalt,deployed) = ERC1155MinimalDeployer.
             ↪ _1155MinimalDeploy(_tokenSalt,_uri,_price,_splitter,router
             ↪ ,_royalty,erc20) (contracts/MADFactory1155.sol#352-361)
          - (tokenSalt,deployed) = ERC1155BasicDeployer._1155BasicDeploy(
             ↪ _tokenSalt,_uri,_price,_maxSupply,_splitter,router,
             ↪ _royalty,erc20) (contracts/MADFactory1155.sol#385-395)
          - (tokenSalt,deployed) = ERC1155WhitelistDeployer.
             ↪ _1155WhitelistDeploy(_tokenSalt,_uri,_price,_maxSupply,
             ↪ _splitter,router,_royalty,erc20) (contracts/MADFactory1155
             ↪ .sol#419-429)
          State variables written after the call(s):
```

```
        - userTokens[tx.origin].push(colId_scope_5) (contracts/
           ↪ MADFactory1155.sol#432)
  Reentrancy in MADFactory1155.createCollection(uint8,string,string,string
     ↪ ,uint256,uint256,string,address,uint256) (contracts/
     ↪ MADFactory1155.sol#330-485):
        External calls:
        - (tokenSalt,deployed) = ERC1155MinimalDeployer.
           ↪ _1155MinimalDeploy(_tokenSalt,_uri,_price,_splitter,router
           ↪ ,_royalty,erc20) (contracts/MADFactory1155.sol#352-361)
        - (tokenSalt,deployed) = ERC1155BasicDeployer._1155BasicDeploy(
           ↪ _tokenSalt,_uri,_price,_maxSupply,_splitter,router,
           ↪ _royalty,erc20) (contracts/MADFactory1155.sol#385-395)
        - (tokenSalt,deployed) = ERC1155WhitelistDeployer.
           ↪ _1155WhitelistDeploy(_tokenSalt,_uri,_price,_maxSupply,
           ↪ _splitter,router,_royalty,erc20) (contracts/MADFactory1155
           ↪ .sol#419-429)
        - (tokenSalt,deployed) = ERC1155LazyDeployer._1155LazyDeploy(
           ↪ _tokenSalt,_uri,_splitter,router,signer,_royalty,erc20) (
           ↪ contracts/MADFactory1155.sol#453-462)
        State variables written after the call(s):
        - userTokens[tx.origin].push(colId_scope_8) (contracts/
           ↪ MADFactory1155.sol#465)
  Reentrancy in MADFactory721.createCollection(uint8,string,string,string,
     ↪ uint256,uint256,string,address,uint256) (contracts/MADFactory721.
     ↪ sol#334-497):
        External calls:
        - (tokenSalt,deployed) = ERC721MinimalDeployer._721MinimalDeploy(
           ↪ _tokenSalt,_name,_symbol,_baseURI,_price,_splitter,router,
           ↪ _royalty,erc20) (contracts/MADFactory721.sol#356-367)
        - (tokenSalt,deployed) = ERC721BasicDeployer._721BasicDeploy(
           ↪ _tokenSalt,_name,_symbol,_baseURI,_price,_maxSupply,
           ↪ _splitter,router,_royalty,erc20) (contracts/MADFactory721.
           ↪ sol#391-403)
        State variables written after the call(s):
```

```
        - userTokens[tx.origin].push(colId_scope_2) (contracts/
           ↪ MADFactory721.sol#406)
  Reentrancy in MADFactory721.createCollection(uint8,string,string,string,
     ↪ uint256,uint256,string,address,uint256) (contracts/MADFactory721.
     ↪ sol#334-497):
        External calls:
        - (tokenSalt,deployed) = ERC721MinimalDeployer._721MinimalDeploy(
           ↪ _tokenSalt,_name,_symbol,_baseURI,_price,_splitter,router,
           ↪ _royalty,erc20) (contracts/MADFactory721.sol#356-367)
        - (tokenSalt,deployed) = ERC721BasicDeployer._721BasicDeploy(
           ↪ _tokenSalt,_name,_symbol,_baseURI,_price,_maxSupply,
           ↪ _splitter,router,_royalty,erc20) (contracts/MADFactory721.
           ↪ sol#391-403)
        - (tokenSalt,deployed) = ERC721WhitelistDeployer.
           ↪ _721WhitelistDeploy(_tokenSalt,_name,_symbol,_baseURI,
           ↪ _price,_maxSupply,_splitter,router,_royalty,erc20) (
           ↪ contracts/MADFactory721.sol#427-439)
        State variables written after the call(s):
        - userTokens[tx.origin].push(colId_scope_5) (contracts/
           ↪ MADFactory721.sol#442)
  Reentrancy in MADFactory721.createCollection(uint8,string,string,string,
     ↪ uint256,uint256,string,address,uint256) (contracts/MADFactory721.
     ↪ sol#334-497):
        External calls:
        - (tokenSalt,deployed) = ERC721MinimalDeployer._721MinimalDeploy(
           ↪ _tokenSalt,_name,_symbol,_baseURI,_price,_splitter,router,
           ↪ _royalty,erc20) (contracts/MADFactory721.sol#356-367)
        - (tokenSalt,deployed) = ERC721BasicDeployer._721BasicDeploy(
           ↪ _tokenSalt,_name,_symbol,_baseURI,_price,_maxSupply,
           ↪ _splitter,router,_royalty,erc20) (contracts/MADFactory721.
           ↪ sol#391-403)
        - (tokenSalt,deployed) = ERC721WhitelistDeployer.
           ↪ _721WhitelistDeploy(_tokenSalt,_name,_symbol,_baseURI,
           ↪ _price,_maxSupply,_splitter,router,_royalty,erc20) (
          ↪ contracts/MADFactory721.sol#427-439)
       - (tokenSalt,deployed) = ERC721LazyDeployer._721LazyDeploy(
          ↪ _tokenSalt,_name,_symbol,_baseURI,_splitter,router,signer,
          ↪ _royalty,erc20) (contracts/MADFactory721.sol#463-474)
       State variables written after the call(s):
       - userTokens[tx.origin].push(colId_scope_8) (contracts/
          ↪ MADFactory721.sol#477)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation
   ↪ #reentrancy-vulnerabilities-1

MADFactory1155.creatorCheck(bytes32) (contracts/MADFactory1155.sol
   ↪ #711-734) uses tx.origin for authorization: creator == origin()()
   ↪  (contracts/MADFactory1155.sol#725-727)
MADFactory721.creatorCheck(bytes32) (contracts/MADFactory721.sol
   ↪ #725-748) uses tx.origin for authorization: creator == origin()()
   ↪  (contracts/MADFactory721.sol#739-741)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation
   ↪ #dangerous-usage-of-txorigin

MADFactory1155.createCollection(uint8,string,string,string,uint256,
   ↪ uint256,string,address,uint256).tokenSalt_scope_0 (contracts/
   ↪ MADFactory1155.sol#385) is a local variable never initialized
MADFactory1155.createCollection(uint8,string,string,string,uint256,
   ↪ uint256,string,address,uint256).deployed_scope_4 (contracts/
   ↪ MADFactory1155.sol#419) is a local variable never initialized
MADFactory721.createCollection(uint8,string,string,string,uint256,
   ↪ uint256,string,address,uint256).deployed_scope_4 (contracts/
   ↪ MADFactory721.sol#427) is a local variable never initialized
MADFactory1155.createCollection(uint8,string,string,string,uint256,
   ↪ uint256,string,address,uint256).deployed_scope_1 (contracts/
   ↪ MADFactory1155.sol#385) is a local variable never initialized
MADFactory721.createCollection(uint8,string,string,string,uint256,
   ↪ uint256,string,address,uint256).tokenSalt_scope_0 (contracts/
   ↪ MADFactory721.sol#391) is a local variable never initialized
```

```
MADFactory721.createCollection(uint8,string,string,string,uint256,
    ↪ uint256,string,address,uint256).deployed_scope_1 (contracts/
    ↪ MADFactory721.sol#391) is a local variable never initialized
MADFactory1155.createCollection(uint8,string,string,string,uint256,
    ↪ uint256,string,address,uint256).tokenSalt_scope_6 (contracts/
    ↪ MADFactory1155.sol#453) is a local variable never initialized
MADFactory1155.createCollection(uint8,string,string,string,uint256,
    ↪ uint256,string,address,uint256).deployed_scope_7 (contracts/
    ↪ MADFactory1155.sol#453) is a local variable never initialized
MADFactory721.createCollection(uint8,string,string,string,uint256,
    ↪ uint256,string,address,uint256).tokenSalt_scope_6 (contracts/
    ↪ MADFactory721.sol#463) is a local variable never initialized
MADFactory721.createCollection(uint8,string,string,string,uint256,
    ↪ uint256,string,address,uint256).deployed_scope_7 (contracts/
    ↪ MADFactory721.sol#463) is a local variable never initialized
MADFactory1155.createCollection(uint8,string,string,string,uint256,
    ↪ uint256,string,address,uint256).tokenSalt_scope_3 (contracts/
    ↪ MADFactory1155.sol#419) is a local variable never initialized
MADFactory721.createCollection(uint8,string,string,string,uint256,
    ↪ uint256,string,address,uint256).tokenSalt_scope_3 (contracts/
    ↪ MADFactory721.sol#427) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation
    ↪ #uninitialized-local-variables

MADRouter1155._tokenRender(address) (contracts/MADRouter1155.sol
    ↪ #595-603) ignores return value by MADFactory1155.creatorCheck(
    ↪ colID) (contracts/MADRouter1155.sol#601)
MADRouter721._tokenRender(address) (contracts/MADRouter721.sol#482-490)
    ↪ ignores return value by MADFactory721.creatorCheck(colID) (
    ↪ contracts/MADRouter721.sol#488)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation
    ↪ #unused-return
```

```
MADRouter1155.feeLookup(bytes4).fee (contracts/MADRouter1155.sol#568) is
    ↪  written in both
        fee = sload(uint256)(feeBurn) (contracts/MADRouter1155.sol#581)
        fee = 0x00 (contracts/MADRouter1155.sol#584)
MADRouter721.feeLookup(bytes4).fee (contracts/MADRouter721.sol#455) is
    ↪ written in both
        fee = sload(uint256)(feeBurn) (contracts/MADRouter721.sol#468)
        fee = 0x00 (contracts/MADRouter721.sol#471)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation
    ↪ #write-after-write

MADFactory1155.constructor(address,address,address,address)._router (
    ↪ contracts/MADFactory1155.sol#94) lacks a zero-check on :
                - router = _router (contracts/MADFactory1155.sol#105)
MADFactory721.constructor(address,address,address,address)._router (
    ↪ contracts/MADFactory721.sol#94) lacks a zero-check on :
                - router = _router (contracts/MADFactory721.sol#105)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation
    ↪ #missing-zero-address-validation

MADMarketplace1155._withdrawOutbid(address,ERC20,uint256,uint160) (
    ↪ contracts/MADMarketplace1155.sol#688-740) has external calls
    ↪ inside a loop: amountOut = swapRouter.exactInputSingle(params) (
    ↪ contracts/MADMarketplace1155.sol#736-738)
MADMarketplace721._withdrawOutbid(address,ERC20,uint256,uint160) (
    ↪ contracts/MADMarketplace721.sol#667-719) has external calls
    ↪ inside a loop: amountOut = swapRouter.exactInputSingle(params) (
    ↪ contracts/MADMarketplace721.sol#715-717)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation
    ↪ /#calls-inside-a-loop

Variable 'MADFactory1155.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).tokenSalt (contracts/
    ↪ MADFactory1155.sol#352)' in MADFactory1155.createCollection(uint8
    ↪ ,string,string,string,uint256,uint256,string,address,uint256) (
    ↪ contracts/MADFactory1155.sol#330-485) potentially used before
    ↪ declaration: (tokenSalt,deployed) = ERC1155BasicDeployer.
    ↪ _1155BasicDeploy(_tokenSalt,_uri,_price,_maxSupply,_splitter,
    ↪ router,_royalty,erc20) (contracts/MADFactory1155.sol#385-395)
Variable 'MADFactory1155.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).deployed (contracts/
    ↪ MADFactory1155.sol#352)' in MADFactory1155.createCollection(uint8
    ↪ ,string,string,string,uint256,uint256,string,address,uint256) (
    ↪ contracts/MADFactory1155.sol#330-485) potentially used before
    ↪ declaration: (tokenSalt,deployed) = ERC1155BasicDeployer.
    ↪ _1155BasicDeploy(_tokenSalt,_uri,_price,_maxSupply,_splitter,
    ↪ router,_royalty,erc20) (contracts/MADFactory1155.sol#385-395)
Variable 'MADFactory1155.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).tokenSalt (contracts/
    ↪ MADFactory1155.sol#352)' in MADFactory1155.createCollection(uint8
    ↪ ,string,string,string,uint256,uint256,string,address,uint256) (
    ↪ contracts/MADFactory1155.sol#330-485) potentially used before
    ↪ declaration: (tokenSalt,deployed) = ERC1155WhitelistDeployer.
    ↪ _1155WhitelistDeploy(_tokenSalt,_uri,_price,_maxSupply,_splitter,
    ↪ router,_royalty,erc20) (contracts/MADFactory1155.sol#419-429)
Variable 'MADFactory1155.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).deployed (contracts/
    ↪ MADFactory1155.sol#352)' in MADFactory1155.createCollection(uint8
    ↪ ,string,string,string,uint256,uint256,string,address,uint256) (
    ↪ contracts/MADFactory1155.sol#330-485) potentially used before
    ↪ declaration: (tokenSalt,deployed) = ERC1155WhitelistDeployer.
    ↪ _1155WhitelistDeploy(_tokenSalt,_uri,_price,_maxSupply,_splitter,
    ↪ router,_royalty,erc20) (contracts/MADFactory1155.sol#419-429)
Variable 'MADFactory1155.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).tokenSalt (contracts/
    ↪ MADFactory1155.sol#352)' in MADFactory1155.createCollection(uint8
    ↪ ,string,string,string,uint256,uint256,string,address,uint256) (
    ↪ contracts/MADFactory1155.sol#330-485) potentially used before
    ↪ declaration: (tokenSalt,deployed) = ERC1155LazyDeployer.
    ↪ _1155LazyDeploy(_tokenSalt,_uri,_splitter,router,signer,_royalty,
    ↪ erc20) (contracts/MADFactory1155.sol#453-462)
Variable 'MADFactory1155.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).deployed (contracts/
    ↪ MADFactory1155.sol#352)' in MADFactory1155.createCollection(uint8
    ↪ ,string,string,string,uint256,uint256,string,address,uint256) (
    ↪ contracts/MADFactory1155.sol#330-485) potentially used before
    ↪ declaration: (tokenSalt,deployed) = ERC1155LazyDeployer.
    ↪ _1155LazyDeploy(_tokenSalt,_uri,_splitter,router,signer,_royalty,
    ↪ erc20) (contracts/MADFactory1155.sol#453-462)
Variable 'MADFactory721.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).deployed (contracts/
    ↪ MADFactory721.sol#356)' in MADFactory721.createCollection(uint8,
    ↪ string,string,string,uint256,uint256,string,address,uint256) (
    ↪ contracts/MADFactory721.sol#334-497) potentially used before
    ↪ declaration: (tokenSalt,deployed) = ERC721BasicDeployer.
    ↪ _721BasicDeploy(_tokenSalt,_name,_symbol,_baseURI,_price,
    ↪ _maxSupply,_splitter,router,_royalty,erc20) (contracts/
    ↪ MADFactory721.sol#391-403)
Variable 'MADFactory721.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).tokenSalt (contracts/
    ↪ MADFactory721.sol#356)' in MADFactory721.createCollection(uint8,
    ↪ string,string,string,uint256,uint256,string,address,uint256) (
    ↪ contracts/MADFactory721.sol#334-497) potentially used before
    ↪ declaration: (tokenSalt,deployed) = ERC721BasicDeployer.
    ↪ _721BasicDeploy(_tokenSalt,_name,_symbol,_baseURI,_price,
    ↪ _maxSupply,_splitter,router,_royalty,erc20) (contracts/
    ↪ MADFactory721.sol#391-403)
Variable 'MADFactory721.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).deployed (contracts/
    ↪ MADFactory721.sol#356)' in MADFactory721.createCollection(uint8,
    ↪ string,string,string,uint256,uint256,string,address,uint256) (
    ↪ contracts/MADFactory721.sol#334-497) potentially used before
    ↪ declaration: (tokenSalt,deployed) = ERC721WhitelistDeployer.
    ↪ _721WhitelistDeploy(_tokenSalt,_name,_symbol,_baseURI,_price,
    ↪ _maxSupply,_splitter,router,_royalty,erc20) (contracts/
    ↪ MADFactory721.sol#427-439)
Variable 'MADFactory721.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).tokenSalt (contracts/
    ↪ MADFactory721.sol#356)' in MADFactory721.createCollection(uint8,
    ↪ string,string,string,uint256,uint256,string,address,uint256) (
    ↪ contracts/MADFactory721.sol#334-497) potentially used before
    ↪ declaration: (tokenSalt,deployed) = ERC721WhitelistDeployer.
    ↪ _721WhitelistDeploy(_tokenSalt,_name,_symbol,_baseURI,_price,
    ↪ _maxSupply,_splitter,router,_royalty,erc20) (contracts/
    ↪ MADFactory721.sol#427-439)
Variable 'MADFactory721.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).deployed (contracts/
    ↪ MADFactory721.sol#356)' in MADFactory721.createCollection(uint8,
    ↪ string,string,string,uint256,uint256,string,address,uint256) (
    ↪ contracts/MADFactory721.sol#334-497) potentially used before
    ↪ declaration: (tokenSalt,deployed) = ERC721LazyDeployer.
    ↪ _721LazyDeploy(_tokenSalt,_name,_symbol,_baseURI,_splitter,router
    ↪ ,signer,_royalty,erc20) (contracts/MADFactory721.sol#463-474)
Variable 'MADFactory721.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).tokenSalt (contracts/
    ↪ MADFactory721.sol#356)' in MADFactory721.createCollection(uint8,
    ↪ string,string,string,uint256,uint256,string,address,uint256) (
    ↪ contracts/MADFactory721.sol#334-497) potentially used before
    ↪ declaration: (tokenSalt,deployed) = ERC721LazyDeployer.
    ↪ _721LazyDeploy(_tokenSalt,_name,_symbol,_baseURI,_splitter,router
    ↪ ,signer,_royalty,erc20) (contracts/MADFactory721.sol#463-474)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation
    ↪ #pre-declaration-usage-of-local-variables

Reentrancy in MADFactory1155.createCollection(uint8,string,string,string
    ↪ ,uint256,uint256,string,address,uint256) (contracts/
    ↪ MADFactory1155.sol#330-485):
       External calls:
       - (tokenSalt,deployed) = ERC1155MinimalDeployer.
           ↪ _1155MinimalDeploy(_tokenSalt,_uri,_price,_splitter,router
           ↪ ,_royalty,erc20) (contracts/MADFactory1155.sol#352-361)
       State variables written after the call(s):
       - colInfo[colId] = Types.Collection1155(tx.origin,Types.
           ↪ ERC1155Type.ERC1155Minimal,tokenSalt,block.number,
           ↪ _splitter) (contracts/MADFactory1155.sol#366-372)
       - userTokens[tx.origin].push(colId) (contracts/MADFactory1155.sol
           ↪ #364)
Reentrancy in MADFactory1155.createCollection(uint8,string,string,string
    ↪ ,uint256,uint256,string,address,uint256) (contracts/
    ↪ MADFactory1155.sol#330-485):
       External calls:
       - (tokenSalt,deployed) = ERC1155MinimalDeployer.
           ↪ _1155MinimalDeploy(_tokenSalt,_uri,_price,_splitter,router
           ↪ ,_royalty,erc20) (contracts/MADFactory1155.sol#352-361)
       - (tokenSalt,deployed) = ERC1155BasicDeployer._1155BasicDeploy(
           ↪ _tokenSalt,_uri,_price,_maxSupply,_splitter,router,
           ↪ _royalty,erc20) (contracts/MADFactory1155.sol#385-395)
       State variables written after the call(s):
       - colInfo[colId_scope_2] = Types.Collection1155(tx.origin,Types.
           ↪ ERC1155Type.ERC1155Basic,tokenSalt_scope_0,block.number,
           ↪ _splitter) (contracts/MADFactory1155.sol#400-406)
Reentrancy in MADFactory1155.createCollection(uint8,string,string,string
    ↪ ,uint256,uint256,string,address,uint256) (contracts/
    ↪ MADFactory1155.sol#330-485):
       External calls:
       - (tokenSalt,deployed) = ERC1155MinimalDeployer.
           ↪ _1155MinimalDeploy(_tokenSalt,_uri,_price,_splitter,router
           ↪ ,_royalty,erc20) (contracts/MADFactory1155.sol#352-361)
       - (tokenSalt,deployed) = ERC1155BasicDeployer._1155BasicDeploy(
           ↪ _tokenSalt,_uri,_price,_maxSupply,_splitter,router,
          ↪ _royalty,erc20) (contracts/MADFactory1155.sol#385-395)
       - (tokenSalt,deployed) = ERC1155WhitelistDeployer.
          ↪ _1155WhitelistDeploy(_tokenSalt,_uri,_price,_maxSupply,
          ↪ _splitter,router,_royalty,erc20) (contracts/MADFactory1155
          ↪ .sol#419-429)
       State variables written after the call(s):
       - colInfo[colId_scope_5] = Types.Collection1155(tx.origin,Types.
          ↪ ERC1155Type.ERC1155Whitelist,tokenSalt_scope_3,block.
          ↪ number,_splitter) (contracts/MADFactory1155.sol#434-440)
Reentrancy in MADFactory1155.createCollection(uint8,string,string,string
   ↪ ,uint256,uint256,string,address,uint256) (contracts/
   ↪ MADFactory1155.sol#330-485):
       External calls:
       - (tokenSalt,deployed) = ERC1155MinimalDeployer.
          ↪ _1155MinimalDeploy(_tokenSalt,_uri,_price,_splitter,router
          ↪ ,_royalty,erc20) (contracts/MADFactory1155.sol#352-361)
       - (tokenSalt,deployed) = ERC1155BasicDeployer._1155BasicDeploy(
          ↪ _tokenSalt,_uri,_price,_maxSupply,_splitter,router,
          ↪ _royalty,erc20) (contracts/MADFactory1155.sol#385-395)
       - (tokenSalt,deployed) = ERC1155WhitelistDeployer.
          ↪ _1155WhitelistDeploy(_tokenSalt,_uri,_price,_maxSupply,
          ↪ _splitter,router,_royalty,erc20) (contracts/MADFactory1155
          ↪ .sol#419-429)
       - (tokenSalt,deployed) = ERC1155LazyDeployer._1155LazyDeploy(
          ↪ _tokenSalt,_uri,_splitter,router,signer,_royalty,erc20) (
          ↪ contracts/MADFactory1155.sol#453-462)
       State variables written after the call(s):
       - colInfo[colId_scope_8] = Types.Collection1155(tx.origin,Types.
          ↪ ERC1155Type.ERC1155Lazy,tokenSalt_scope_6,block.number,
          ↪ _splitter) (contracts/MADFactory1155.sol#467-473)
Reentrancy in MADFactory721.createCollection(uint8,string,string,string,
   ↪ uint256,uint256,string,address,uint256) (contracts/MADFactory721.
   ↪ sol#334-497):
       External calls:
       - (tokenSalt,deployed) = ERC721MinimalDeployer._721MinimalDeploy(
          ↪ _tokenSalt,_name,_symbol,_baseURI,_price,_splitter,router,
          ↪ _royalty,erc20) (contracts/MADFactory721.sol#356-367)
       State variables written after the call(s):
       - colInfo[colId] = Types.Collection721(tx.origin,Types.ERC721Type
          ↪ .ERC721Minimal,tokenSalt,block.number,_splitter) (
          ↪ contracts/MADFactory721.sol#372-378)
       - userTokens[tx.origin].push(colId) (contracts/MADFactory721.sol
          ↪ #370)
Reentrancy in MADFactory721.createCollection(uint8,string,string,string,
   ↪ uint256,uint256,string,address,uint256) (contracts/MADFactory721.
   ↪ sol#334-497):
       External calls:
       - (tokenSalt,deployed) = ERC721MinimalDeployer._721MinimalDeploy(
          ↪ _tokenSalt,_name,_symbol,_baseURI,_price,_splitter,router,
          ↪ _royalty,erc20) (contracts/MADFactory721.sol#356-367)
       - (tokenSalt,deployed) = ERC721BasicDeployer._721BasicDeploy(
          ↪ _tokenSalt,_name,_symbol,_baseURI,_price,_maxSupply,
          ↪ _splitter,router,_royalty,erc20) (contracts/MADFactory721.
          ↪ sol#391-403)
       State variables written after the call(s):
       - colInfo[colId_scope_2] = Types.Collection721(tx.origin,Types.
          ↪ ERC721Type.ERC721Basic,tokenSalt_scope_0,block.number,
          ↪ _splitter) (contracts/MADFactory721.sol#408-414)
Reentrancy in MADFactory721.createCollection(uint8,string,string,string,
   ↪ uint256,uint256,string,address,uint256) (contracts/MADFactory721.
   ↪ sol#334-497):
       External calls:
       - (tokenSalt,deployed) = ERC721MinimalDeployer._721MinimalDeploy(
          ↪ _tokenSalt,_name,_symbol,_baseURI,_price,_splitter,router,
          ↪ _royalty,erc20) (contracts/MADFactory721.sol#356-367)
       - (tokenSalt,deployed) = ERC721BasicDeployer._721BasicDeploy(
          ↪ _tokenSalt,_name,_symbol,_baseURI,_price,_maxSupply,
          ↪ _splitter,router,_royalty,erc20) (contracts/MADFactory721.
          ↪ sol#391-403)
       - (tokenSalt,deployed) = ERC721WhitelistDeployer.
          ↪ _721WhitelistDeploy(_tokenSalt,_name,_symbol,_baseURI,
          ↪ _price,_maxSupply,_splitter,router,_royalty,erc20) (
          ↪ contracts/MADFactory721.sol#427-439)
       State variables written after the call(s):
       - colInfo[colId_scope_5] = Types.Collection721(tx.origin,Types.
          ↪ ERC721Type.ERC721Whitelist,tokenSalt_scope_3,block.number,
          ↪ _splitter) (contracts/MADFactory721.sol#444-450)
Reentrancy in MADFactory721.createCollection(uint8,string,string,string,
   ↪ uint256,uint256,string,address,uint256) (contracts/MADFactory721.
   ↪ sol#334-497):
       External calls:
       - (tokenSalt,deployed) = ERC721MinimalDeployer._721MinimalDeploy(
          ↪ _tokenSalt,_name,_symbol,_baseURI,_price,_splitter,router,
          ↪ _royalty,erc20) (contracts/MADFactory721.sol#356-367)
       - (tokenSalt,deployed) = ERC721BasicDeployer._721BasicDeploy(
          ↪ _tokenSalt,_name,_symbol,_baseURI,_price,_maxSupply,
          ↪ _splitter,router,_royalty,erc20) (contracts/MADFactory721.
          ↪ sol#391-403)
       - (tokenSalt,deployed) = ERC721WhitelistDeployer.
          ↪ _721WhitelistDeploy(_tokenSalt,_name,_symbol,_baseURI,
          ↪ _price,_maxSupply,_splitter,router,_royalty,erc20) (
          ↪ contracts/MADFactory721.sol#427-439)
       - (tokenSalt,deployed) = ERC721LazyDeployer._721LazyDeploy(
          ↪ _tokenSalt,_name,_symbol,_baseURI,_splitter,router,signer,
          ↪ _royalty,erc20) (contracts/MADFactory721.sol#463-474)
       State variables written after the call(s):
       - colInfo[colId_scope_8] = Types.Collection721(tx.origin,Types.
          ↪ ERC721Type.ERC721Lazy,tokenSalt_scope_6,block.number,
          ↪ _splitter) (contracts/MADFactory721.sol#479-485)
Reentrancy in MADFactory1155.splitterCheck(string,address,address,
   ↪ uint256,uint256) (contracts/MADFactory1155.sol#130-310):
       External calls:
       - _splitter = SplitterDeployer._SplitterDeploy(_splitterSalt,
          ↪ _payees,_shares) (contracts/MADFactory1155.sol#152-156)
       State variables written after the call(s):
       - splitterInfo[tx.origin][_splitter] = Types.SplitterConfig(
          ↪ _splitter,splitterSalt,address(0),address(0),0,0,true) (
          ↪ contracts/MADFactory1155.sol#158-167)
Reentrancy in MADFactory1155.splitterCheck(string,address,address,
   ↪ uint256,uint256) (contracts/MADFactory1155.sol#130-310):
       External calls:
       - _splitter_scope_2 = SplitterDeployer._SplitterDeploy(
          ↪ _splitterSalt,_payees_scope_0,_shares_scope_1) (contracts/
          ↪ MADFactory1155.sol#191-195)
       State variables written after the call(s):
       - splitterInfo[tx.origin][_splitter_scope_2] = Types.
          ↪ SplitterConfig(_splitter_scope_2,splitterSalt,_ambassador,
          ↪ address(0),_ambShare,0,true) (contracts/MADFactory1155.sol
          ↪ #197-206)
Reentrancy in MADFactory1155.splitterCheck(string,address,address,
   ↪ uint256,uint256) (contracts/MADFactory1155.sol#130-310):
       External calls:
       - _splitter_scope_5 = SplitterDeployer._SplitterDeploy(
          ↪ _splitterSalt,_payees_scope_3,_shares_scope_4) (contracts/
          ↪ MADFactory1155.sol#230-234)
       State variables written after the call(s):
       - splitterInfo[tx.origin][_splitter_scope_5] = Types.
          ↪ SplitterConfig(_splitter_scope_5,splitterSalt,address(0),
          ↪ _project,0,_projectShare,true) (contracts/MADFactory1155.
          ↪ sol#236-245)
Reentrancy in MADFactory1155.splitterCheck(string,address,address,
   ↪ uint256,uint256) (contracts/MADFactory1155.sol#130-310):
       External calls:
       - _splitter_scope_8 = SplitterDeployer._SplitterDeploy(
          ↪ _splitterSalt,_payees_scope_6,_shares_scope_7) (contracts/
          ↪ MADFactory1155.sol#277-281)
```

```
       State variables written after the call(s):
       - splitterInfo[tx.origin][_splitter_scope_8] = Types.
          ↪ SplitterConfig(_splitter_scope_8,splitterSalt,_ambassador,
          ↪ _project,_ambShare,_projectShare,true) (contracts/
          ↪ MADFactory1155.sol#283-292)
Reentrancy in MADFactory721.splitterCheck(string,address,address,uint256
   ↪ ,uint256) (contracts/MADFactory721.sol#134-314):
       External calls:
       - _splitter = SplitterDeployer._SplitterDeploy(_splitterSalt,
          ↪ _payees,_shares) (contracts/MADFactory721.sol#156-160)
       State variables written after the call(s):
       - splitterInfo[tx.origin][_splitter] = Types.SplitterConfig(
          ↪ _splitter,splitterSalt,address(0),address(0),0,0,true) (
          ↪ contracts/MADFactory721.sol#162-171)
Reentrancy in MADFactory721.splitterCheck(string,address,address,uint256
   ↪ ,uint256) (contracts/MADFactory721.sol#134-314):
       External calls:
       - _splitter_scope_2 = SplitterDeployer._SplitterDeploy(
          ↪ _splitterSalt,_payees_scope_0,_shares_scope_1) (contracts/
          ↪ MADFactory721.sol#195-199)
       State variables written after the call(s):
       - splitterInfo[tx.origin][_splitter_scope_2] = Types.
          ↪ SplitterConfig(_splitter_scope_2,splitterSalt,_ambassador,
          ↪ address(0),_ambShare,0,true) (contracts/MADFactory721.sol
          ↪ #201-210)
Reentrancy in MADFactory721.splitterCheck(string,address,address,uint256
   ↪ ,uint256) (contracts/MADFactory721.sol#134-314):
       External calls:
       - _splitter_scope_5 = SplitterDeployer._SplitterDeploy(
          ↪ _splitterSalt,_payees_scope_3,_shares_scope_4) (contracts/
          ↪ MADFactory721.sol#234-238)
       State variables written after the call(s):
       - splitterInfo[tx.origin][_splitter_scope_5] = Types.
          ↪ SplitterConfig(_splitter_scope_5,splitterSalt,address(0),
            ↪ _project,0,_projectShare,true) (contracts/MADFactory721.
            ↪ sol#240-249)
Reentrancy in MADFactory721.splitterCheck(string,address,address,uint256
    ↪ ,uint256) (contracts/MADFactory721.sol#134-314):
        External calls:
        - _splitter_scope_8 = SplitterDeployer._SplitterDeploy(
            ↪ _splitterSalt,_payees_scope_6,_shares_scope_7) (contracts/
            ↪ MADFactory721.sol#281-285)
        State variables written after the call(s):
        - splitterInfo[tx.origin][_splitter_scope_8] = Types.
            ↪ SplitterConfig(_splitter_scope_8,splitterSalt,_ambassador,
            ↪ _project,_ambShare,_projectShare,true) (contracts/
            ↪ MADFactory721.sol#287-296)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation
    ↪ #reentrancy-vulnerabilities-2

Reentrancy in MADMarketplace721._extPath0(Types.Order721,uint256,bytes32
    ↪ ,address) (contracts/MADMarketplace721.sol#920-986):
        External calls:
        - _order.token.safeTransferFrom(address(this),_to,_order.tokenId)
            ↪  (contracts/MADMarketplace721.sol#973-977)
        Event emitted after the call(s):
        - Claim(_order.token,_order.tokenId,_orderId,_order.seller,_to,
            ↪ _price) (contracts/MADMarketplace721.sol#978-985)
Reentrancy in MADMarketplace721._extPath1(Types.Order721,uint256,bytes32
    ↪ ,address) (contracts/MADMarketplace721.sol#988-1032):
        External calls:
        - _order.token.safeTransferFrom(address(this),_to,_order.tokenId)
            ↪  (contracts/MADMarketplace721.sol#1019-1023)
        Event emitted after the call(s):
        - Claim(_order.token,_order.tokenId,_orderId,_order.seller,_to,
            ↪ _price) (contracts/MADMarketplace721.sol#1024-1031)
Reentrancy in MADMarketplace721._intPath(Types.Order721,uint256,bytes32,
    ↪ address,uint256) (contracts/MADMarketplace721.sol#851-918):
        External calls:
        - _order.token.safeTransferFrom(address(this),_to,_order.tokenId)
            ↪  (contracts/MADMarketplace721.sol#905-909)
        Event emitted after the call(s):
        - Claim(_order.token,_order.tokenId,_orderId,_order.seller,_to,
            ↪ _price) (contracts/MADMarketplace721.sol#910-917)
Reentrancy in MADMarketplace1155._withdrawOutbid(address,ERC20,uint256,
    ↪ uint160) (contracts/MADMarketplace1155.sol#688-740):
        External calls:
        - amountOut = swapRouter.exactInputSingle(params) (contracts/
            ↪ MADMarketplace1155.sol#736-738)
        Event emitted after the call(s):
        - WithdrawOutbid(_sender,address(_token),amountOut) (contracts/
            ↪ MADMarketplace1155.sol#739)
Reentrancy in MADMarketplace721._withdrawOutbid(address,ERC20,uint256,
    ↪ uint160) (contracts/MADMarketplace721.sol#667-719):
        External calls:
        - amountOut = swapRouter.exactInputSingle(params) (contracts/
            ↪ MADMarketplace721.sol#715-717)
        Event emitted after the call(s):
        - WithdrawOutbid(_sender,address(_token),amountOut) (contracts/
            ↪ MADMarketplace721.sol#718)
Reentrancy in MADFactory1155.createCollection(uint8,string,string,string
    ↪ ,uint256,uint256,string,address,uint256) (contracts/
    ↪ MADFactory1155.sol#330-485):
        External calls:
        - (tokenSalt,deployed) = ERC1155MinimalDeployer.
            ↪ _1155MinimalDeploy(_tokenSalt,_uri,_price,_splitter,router
            ↪ ,_royalty,erc20) (contracts/MADFactory1155.sol#352-361)
        Event emitted after the call(s):
        - ERC1155MinimalCreated(_splitter,deployed,_name,_symbol,_royalty
            ↪ ,_maxSupply,_price) (contracts/MADFactory1155.sol#374-382)
Reentrancy in MADFactory1155.createCollection(uint8,string,string,string
    ↪ ,uint256,uint256,string,address,uint256) (contracts/
```

```
    ↪ MADFactory1155.sol#330-485):
        External calls:
        - (tokenSalt,deployed) = ERC1155MinimalDeployer.
           ↪ _1155MinimalDeploy(_tokenSalt,_uri,_price,_splitter,router
           ↪ ,_royalty,erc20) (contracts/MADFactory1155.sol#352-361)
        - (tokenSalt,deployed) = ERC1155BasicDeployer._1155BasicDeploy(
           ↪ _tokenSalt,_uri,_price,_maxSupply,_splitter,router,
           ↪ _royalty,erc20) (contracts/MADFactory1155.sol#385-395)
        Event emitted after the call(s):
        - ERC1155BasicCreated(_splitter,deployed_scope_1,_name,_symbol,
           ↪ _royalty,_maxSupply,_price) (contracts/MADFactory1155.sol
           ↪ #408-416)
Reentrancy in MADFactory1155.createCollection(uint8,string,string,string
    ↪ ,uint256,uint256,string,address,uint256) (contracts/
    ↪ MADFactory1155.sol#330-485):
        External calls:
        - (tokenSalt,deployed) = ERC1155MinimalDeployer.
           ↪ _1155MinimalDeploy(_tokenSalt,_uri,_price,_splitter,router
           ↪ ,_royalty,erc20) (contracts/MADFactory1155.sol#352-361)
        - (tokenSalt,deployed) = ERC1155BasicDeployer._1155BasicDeploy(
           ↪ _tokenSalt,_uri,_price,_maxSupply,_splitter,router,
           ↪ _royalty,erc20) (contracts/MADFactory1155.sol#385-395)
        - (tokenSalt,deployed) = ERC1155WhitelistDeployer.
           ↪ _1155WhitelistDeploy(_tokenSalt,_uri,_price,_maxSupply,
           ↪ _splitter,router,_royalty,erc20) (contracts/MADFactory1155
           ↪ .sol#419-429)
        Event emitted after the call(s):
        - ERC1155WhitelistCreated(_splitter,deployed_scope_4,_name,
           ↪ _symbol,_royalty,_maxSupply,_price) (contracts/
           ↪ MADFactory1155.sol#442-450)
Reentrancy in MADFactory1155.createCollection(uint8,string,string,string
    ↪ ,uint256,uint256,string,address,uint256) (contracts/
    ↪ MADFactory1155.sol#330-485):
        External calls:
        - (tokenSalt,deployed) = ERC1155MinimalDeployer.
           ↪ _1155MinimalDeploy(_tokenSalt,_uri,_price,_splitter,router
           ↪ ,_royalty,erc20) (contracts/MADFactory1155.sol#352-361)
        - (tokenSalt,deployed) = ERC1155BasicDeployer._1155BasicDeploy(
           ↪ _tokenSalt,_uri,_price,_maxSupply,_splitter,router,
           ↪ _royalty,erc20) (contracts/MADFactory1155.sol#385-395)
        - (tokenSalt,deployed) = ERC1155WhitelistDeployer.
           ↪ _1155WhitelistDeploy(_tokenSalt,_uri,_price,_maxSupply,
           ↪ _splitter,router,_royalty,erc20) (contracts/MADFactory1155
           ↪ .sol#419-429)
        - (tokenSalt,deployed) = ERC1155LazyDeployer._1155LazyDeploy(
           ↪ _tokenSalt,_uri,_splitter,router,signer,_royalty,erc20) (
           ↪ contracts/MADFactory1155.sol#453-462)
        Event emitted after the call(s):
        - ERC1155LazyCreated(_splitter,deployed_scope_7,_name,_symbol,
           ↪ _royalty,_maxSupply,_price) (contracts/MADFactory1155.sol
           ↪ #475-483)
Reentrancy in MADFactory721.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256) (contracts/MADFactory721.
    ↪ sol#334-497):
        External calls:
        - (tokenSalt,deployed) = ERC721MinimalDeployer._721MinimalDeploy(
           ↪ _tokenSalt,_name,_symbol,_baseURI,_price,_splitter,router,
           ↪ _royalty,erc20) (contracts/MADFactory721.sol#356-367)
        Event emitted after the call(s):
        - ERC721MinimalCreated(_splitter,deployed,_name,_symbol,_royalty,
           ↪ _maxSupply,_price) (contracts/MADFactory721.sol#380-388)
Reentrancy in MADFactory721.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256) (contracts/MADFactory721.
    ↪ sol#334-497):
        External calls:
        - (tokenSalt,deployed) = ERC721MinimalDeployer._721MinimalDeploy(
           ↪ _tokenSalt,_name,_symbol,_baseURI,_price,_splitter,router,
           ↪ _royalty,erc20) (contracts/MADFactory721.sol#356-367)
```

```
       - (tokenSalt,deployed) = ERC721BasicDeployer._721BasicDeploy(
          ↪ _tokenSalt,_name,_symbol,_baseURI,_price,_maxSupply,
          ↪ _splitter,router,_royalty,erc20) (contracts/MADFactory721.
          ↪ sol#391-403)
       Event emitted after the call(s):
       - ERC721BasicCreated(_splitter,deployed_scope_1,_name,_symbol,
          ↪ _royalty,_maxSupply,_price) (contracts/MADFactory721.sol
          ↪ #416-424)
Reentrancy in MADFactory721.createCollection(uint8,string,string,string,
   ↪ uint256,uint256,string,address,uint256) (contracts/MADFactory721.
   ↪ sol#334-497):
       External calls:
       - (tokenSalt,deployed) = ERC721MinimalDeployer._721MinimalDeploy(
          ↪ _tokenSalt,_name,_symbol,_baseURI,_price,_splitter,router,
          ↪ _royalty,erc20) (contracts/MADFactory721.sol#356-367)
       - (tokenSalt,deployed) = ERC721BasicDeployer._721BasicDeploy(
          ↪ _tokenSalt,_name,_symbol,_baseURI,_price,_maxSupply,
          ↪ _splitter,router,_royalty,erc20) (contracts/MADFactory721.
          ↪ sol#391-403)
       - (tokenSalt,deployed) = ERC721WhitelistDeployer.
          ↪ _721WhitelistDeploy(_tokenSalt,_name,_symbol,_baseURI,
          ↪ _price,_maxSupply,_splitter,router,_royalty,erc20) (
          ↪ contracts/MADFactory721.sol#427-439)
       Event emitted after the call(s):
       - ERC721WhitelistCreated(_splitter,deployed_scope_4,_name,_symbol
          ↪ ,_royalty,_maxSupply,_price) (contracts/MADFactory721.sol
          ↪ #452-460)
Reentrancy in MADFactory721.createCollection(uint8,string,string,string,
   ↪ uint256,uint256,string,address,uint256) (contracts/MADFactory721.
   ↪ sol#334-497):
       External calls:
       - (tokenSalt,deployed) = ERC721MinimalDeployer._721MinimalDeploy(
          ↪ _tokenSalt,_name,_symbol,_baseURI,_price,_splitter,router,
          ↪ _royalty,erc20) (contracts/MADFactory721.sol#356-367)
```

```
        - (tokenSalt,deployed) = ERC721BasicDeployer._721BasicDeploy(
          ↪ _tokenSalt,_name,_symbol,_baseURI,_price,_maxSupply,
          ↪ _splitter,router,_royalty,erc20) (contracts/MADFactory721.
          ↪ sol#391-403)
        - (tokenSalt,deployed) = ERC721WhitelistDeployer.
          ↪ _721WhitelistDeploy(_tokenSalt,_name,_symbol,_baseURI,
          ↪ _price,_maxSupply,_splitter,router,_royalty,erc20) (
          ↪ contracts/MADFactory721.sol#427-439)
        - (tokenSalt,deployed) = ERC721LazyDeployer._721LazyDeploy(
          ↪ _tokenSalt,_name,_symbol,_baseURI,_splitter,router,signer,
          ↪ _royalty,erc20) (contracts/MADFactory721.sol#463-474)
        Event emitted after the call(s):
        - ERC721LazyCreated(_splitter,deployed_scope_7,_name,_symbol,
          ↪ _royalty,_maxSupply,_price) (contracts/MADFactory721.sol
          ↪ #487-495)
Reentrancy in MADRouter721.setBase(address,string) (contracts/
    ↪ MADRouter721.sol#120-141):
        External calls:
        - ERC721Basic(_token).setBaseURI(_baseURI) (contracts/
          ↪ MADRouter721.sol#130)
        Event emitted after the call(s):
        - BaseURI(_colID,_baseURI) (contracts/MADRouter721.sol#131)
Reentrancy in MADRouter721.setBase(address,string) (contracts/
    ↪ MADRouter721.sol#120-141):
        External calls:
        - ERC721Whitelist(_token).setBaseURI(_baseURI) (contracts/
          ↪ MADRouter721.sol#133)
        Event emitted after the call(s):
        - BaseURI(_colID,_baseURI) (contracts/MADRouter721.sol#134)
Reentrancy in MADRouter721.setBase(address,string) (contracts/
    ↪ MADRouter721.sol#120-141):
        External calls:
        - ERC721Lazy(_token).setBaseURI(_baseURI) (contracts/MADRouter721
          ↪ .sol#136)
```

```
       Event emitted after the call(s):
       - BaseURI(_colID,_baseURI) (contracts/MADRouter721.sol#137)
Reentrancy in MADRouter1155.setMintState(address,bool,uint8) (contracts/
   ↪ MADRouter1155.sol#194-218):
       External calls:
       - _stateType0(_tokenType,_token,_state) (contracts/MADRouter1155.
          ↪ sol#205)
              - ERC1155Minimal(_token).setPublicMintState(_state) (
                 ↪ contracts/MADRouter1155.sol#616)
              - ERC1155Basic(_token).setPublicMintState(_state) (
                 ↪ contracts/MADRouter1155.sol#618)
              - ERC1155Whitelist(_token).setPublicMintState(_state) (
                 ↪ contracts/MADRouter1155.sol#620-622)
       Event emitted after the call(s):
       - PublicMintState(_colID,_tokenType,_state) (contracts/
          ↪ MADRouter1155.sol#206)
Reentrancy in MADRouter1155.setMintState(address,bool,uint8) (contracts/
   ↪ MADRouter1155.sol#194-218):
       External calls:
       - _stateType1(_tokenType,_token,_state) (contracts/MADRouter1155.
          ↪ sol#208)
              - ERC1155Whitelist(_token).setWhitelistMintState(_state) (
                 ↪ contracts/MADRouter1155.sol#637-639)
       Event emitted after the call(s):
       - WhitelistMintState(_colID,_tokenType,_state) (contracts/
          ↪ MADRouter1155.sol#209-213)
Reentrancy in MADRouter1155.setMintState(address,bool,uint8) (contracts/
   ↪ MADRouter1155.sol#194-218):
       External calls:
       - _stateType2(_tokenType,_token,_state) (contracts/MADRouter1155.
          ↪ sol#215)
              - ERC1155Whitelist(_token).setFreeClaimState(_state) (
                 ↪ contracts/MADRouter1155.sol#654-656)
       Event emitted after the call(s):
```

```
       - FreeClaimState(_colID,_tokenType,_state) (contracts/
          ↪ MADRouter1155.sol#216)
Reentrancy in MADRouter721.setMintState(address,bool,uint8) (contracts/
   ↪ MADRouter721.sol#179-203):
       External calls:
       - _stateType0(_tokenType,_token,_state) (contracts/MADRouter721.
          ↪ sol#190)
              - ERC721Minimal(_token).setPublicMintState(_state) (
                 ↪ contracts/MADRouter721.sol#503)
              - ERC721Basic(_token).setPublicMintState(_state) (
                 ↪ contracts/MADRouter721.sol#505)
              - ERC721Whitelist(_token).setPublicMintState(_state) (
                 ↪ contracts/MADRouter721.sol#507-509)
       Event emitted after the call(s):
       - PublicMintState(_colID,_tokenType,_state) (contracts/
          ↪ MADRouter721.sol#191)
Reentrancy in MADRouter721.setMintState(address,bool,uint8) (contracts/
   ↪ MADRouter721.sol#179-203):
       External calls:
       - _stateType1(_tokenType,_token,_state) (contracts/MADRouter721.
          ↪ sol#193)
              - ERC721Whitelist(_token).setWhitelistMintState(_state) (
                 ↪ contracts/MADRouter721.sol#524-526)
       Event emitted after the call(s):
       - WhitelistMintState(_colID,_tokenType,_state) (contracts/
          ↪ MADRouter721.sol#194-198)
Reentrancy in MADRouter721.setMintState(address,bool,uint8) (contracts/
   ↪ MADRouter721.sol#179-203):
       External calls:
       - _stateType2(_tokenType,_token,_state) (contracts/MADRouter721.
          ↪ sol#200)
              - ERC721Whitelist(_token).setFreeClaimState(_state) (
                 ↪ contracts/MADRouter721.sol#541)
       Event emitted after the call(s):
       - FreeClaimState(_colID,_tokenType,_state) (contracts/
          ↪ MADRouter721.sol#201)
Reentrancy in MADRouter1155.setURI(address,string) (contracts/
   ↪ MADRouter1155.sol#133-154):
       External calls:
       - ERC1155Basic(_token).setURI(_uri) (contracts/MADRouter1155.sol
          ↪ #143)
       Event emitted after the call(s):
       - BaseURI(_colID,_uri) (contracts/MADRouter1155.sol#144)
Reentrancy in MADRouter1155.setURI(address,string) (contracts/
   ↪ MADRouter1155.sol#133-154):
       External calls:
       - ERC1155Whitelist(_token).setURI(_uri) (contracts/MADRouter1155.
          ↪ sol#146)
       Event emitted after the call(s):
       - BaseURI(_colID,_uri) (contracts/MADRouter1155.sol#147)
Reentrancy in MADRouter1155.setURI(address,string) (contracts/
   ↪ MADRouter1155.sol#133-154):
       External calls:
       - ERC1155Lazy(_token).setURI(_uri) (contracts/MADRouter1155.sol
          ↪ #149)
       Event emitted after the call(s):
       - BaseURI(_colID,_uri) (contracts/MADRouter1155.sol#150)
Reentrancy in MADFactory1155.splitterCheck(string,address,address,
   ↪ uint256,uint256) (contracts/MADFactory1155.sol#130-310):
       External calls:
       - _splitter = SplitterDeployer._SplitterDeploy(_splitterSalt,
          ↪ _payees,_shares) (contracts/MADFactory1155.sol#152-156)
       Event emitted after the call(s):
       - SplitterCreated(tx.origin,_shares,_payees,_splitter,0) (
          ↪ contracts/MADFactory1155.sol#169-175)
Reentrancy in MADFactory1155.splitterCheck(string,address,address,
   ↪ uint256,uint256) (contracts/MADFactory1155.sol#130-310):
       External calls:
```

```
       - _splitter_scope_2 = SplitterDeployer._SplitterDeploy(
          ↪ _splitterSalt,_payees_scope_0,_shares_scope_1) (contracts/
          ↪ MADFactory1155.sol#191-195)
       Event emitted after the call(s):
       - SplitterCreated(tx.origin,_shares_scope_1,_payees_scope_0,
          ↪ _splitter_scope_2,1) (contracts/MADFactory1155.sol
          ↪ #208-214)
Reentrancy in MADFactory1155.splitterCheck(string,address,address,
   ↪ uint256,uint256) (contracts/MADFactory1155.sol#130-310):
       External calls:
       - _splitter_scope_5 = SplitterDeployer._SplitterDeploy(
          ↪ _splitterSalt,_payees_scope_3,_shares_scope_4) (contracts/
          ↪ MADFactory1155.sol#230-234)
       Event emitted after the call(s):
       - SplitterCreated(tx.origin,_shares_scope_4,_payees_scope_3,
          ↪ _splitter_scope_5,2) (contracts/MADFactory1155.sol
          ↪ #247-253)
Reentrancy in MADFactory1155.splitterCheck(string,address,address,
   ↪ uint256,uint256) (contracts/MADFactory1155.sol#130-310):
       External calls:
       - _splitter_scope_8 = SplitterDeployer._SplitterDeploy(
          ↪ _splitterSalt,_payees_scope_6,_shares_scope_7) (contracts/
          ↪ MADFactory1155.sol#277-281)
       Event emitted after the call(s):
       - SplitterCreated(tx.origin,_shares_scope_7,_payees_scope_6,
          ↪ _splitter_scope_8,3) (contracts/MADFactory1155.sol
          ↪ #294-300)
Reentrancy in MADFactory721.splitterCheck(string,address,address,uint256
   ↪ ,uint256) (contracts/MADFactory721.sol#134-314):
       External calls:
       - _splitter = SplitterDeployer._SplitterDeploy(_splitterSalt,
          ↪ _payees,_shares) (contracts/MADFactory721.sol#156-160)
       Event emitted after the call(s):
```

```
       - SplitterCreated(tx.origin,_shares,_payees,_splitter,0) (
          ↪ contracts/MADFactory721.sol#173-179)
Reentrancy in MADFactory721.splitterCheck(string,address,address,uint256
   ↪ ,uint256) (contracts/MADFactory721.sol#134-314):
       External calls:
       - _splitter_scope_2 = SplitterDeployer._SplitterDeploy(
          ↪ _splitterSalt,_payees_scope_0,_shares_scope_1) (contracts/
          ↪ MADFactory721.sol#195-199)
       Event emitted after the call(s):
       - SplitterCreated(tx.origin,_shares_scope_1,_payees_scope_0,
          ↪ _splitter_scope_2,1) (contracts/MADFactory721.sol#212-218)
Reentrancy in MADFactory721.splitterCheck(string,address,address,uint256
   ↪ ,uint256) (contracts/MADFactory721.sol#134-314):
       External calls:
       - _splitter_scope_5 = SplitterDeployer._SplitterDeploy(
          ↪ _splitterSalt,_payees_scope_3,_shares_scope_4) (contracts/
          ↪ MADFactory721.sol#234-238)
       Event emitted after the call(s):
       - SplitterCreated(tx.origin,_shares_scope_4,_payees_scope_3,
          ↪ _splitter_scope_5,2) (contracts/MADFactory721.sol#251-257)
Reentrancy in MADFactory721.splitterCheck(string,address,address,uint256
   ↪ ,uint256) (contracts/MADFactory721.sol#134-314):
       External calls:
       - _splitter_scope_8 = SplitterDeployer._SplitterDeploy(
          ↪ _splitterSalt,_payees_scope_6,_shares_scope_7) (contracts/
          ↪ MADFactory721.sol#281-285)
       Event emitted after the call(s):
       - SplitterCreated(tx.origin,_shares_scope_7,_payees_scope_6,
          ↪ _splitter_scope_8,3) (contracts/MADFactory721.sol#298-304)
Reentrancy in MADRouter1155.withdraw(address,ERC20) (contracts/
   ↪ MADRouter1155.sol#487-555):
       External calls:
       - ERC1155Minimal(_token).withdrawERC20(_erc20,recipient) (
          ↪ contracts/MADRouter1155.sol#497-502)
       - ERC1155Minimal(_token).withdraw(recipient) (contracts/
          ↪ MADRouter1155.sol#497-502)
       Event emitted after the call(s):
       - TokenFundsWithdrawn(_colID,_tokenType,msg.sender) (contracts/
          ↪ MADRouter1155.sol#504-508)
Reentrancy in MADRouter1155.withdraw(address,ERC20) (contracts/
   ↪ MADRouter1155.sol#487-555):
       External calls:
       - ERC1155Minimal(_token).withdrawERC20(_erc20,recipient) (
          ↪ contracts/MADRouter1155.sol#497-502)
       - ERC1155Basic(_token).withdrawERC20(_erc20,recipient) (contracts
          ↪ /MADRouter1155.sol#512-517)
       - ERC1155Minimal(_token).withdraw(recipient) (contracts/
          ↪ MADRouter1155.sol#497-502)
       - ERC1155Basic(_token).withdraw(recipient) (contracts/
          ↪ MADRouter1155.sol#512-517)
       Event emitted after the call(s):
       - TokenFundsWithdrawn(_colID,_tokenType,msg.sender) (contracts/
          ↪ MADRouter1155.sol#519-523)
Reentrancy in MADRouter1155.withdraw(address,ERC20) (contracts/
   ↪ MADRouter1155.sol#487-555):
       External calls:
       - ERC1155Minimal(_token).withdrawERC20(_erc20,recipient) (
          ↪ contracts/MADRouter1155.sol#497-502)
       - ERC1155Basic(_token).withdrawERC20(_erc20,recipient) (contracts
          ↪ /MADRouter1155.sol#512-517)
       - ERC1155Whitelist(_token).withdrawERC20(_erc20,recipient) (
          ↪ contracts/MADRouter1155.sol#527-532)
       - ERC1155Minimal(_token).withdraw(recipient) (contracts/
          ↪ MADRouter1155.sol#497-502)
       - ERC1155Basic(_token).withdraw(recipient) (contracts/
          ↪ MADRouter1155.sol#512-517)
       - ERC1155Whitelist(_token).withdraw(recipient) (contracts/
          ↪ MADRouter1155.sol#527-532)
```

```
        Event emitted after the call(s):
        - TokenFundsWithdrawn(_colID,_tokenType,msg.sender) (contracts/
          ↪ MADRouter1155.sol#534-538)
Reentrancy in MADRouter1155.withdraw(address,ERC20) (contracts/
   ↪ MADRouter1155.sol#487-555):
        External calls:
        - ERC1155Minimal(_token).withdrawERC20(_erc20,recipient) (
          ↪ contracts/MADRouter1155.sol#497-502)
        - ERC1155Basic(_token).withdrawERC20(_erc20,recipient) (contracts
          ↪ /MADRouter1155.sol#512-517)
        - ERC1155Whitelist(_token).withdrawERC20(_erc20,recipient) (
          ↪ contracts/MADRouter1155.sol#527-532)
        - ERC1155Lazy(_token).withdrawERC20(_erc20,recipient) (contracts/
          ↪ MADRouter1155.sol#542-547)
        - ERC1155Minimal(_token).withdraw(recipient) (contracts/
          ↪ MADRouter1155.sol#497-502)
        - ERC1155Basic(_token).withdraw(recipient) (contracts/
          ↪ MADRouter1155.sol#512-517)
        - ERC1155Whitelist(_token).withdraw(recipient) (contracts/
          ↪ MADRouter1155.sol#527-532)
        - ERC1155Lazy(_token).withdraw(recipient) (contracts/
          ↪ MADRouter1155.sol#542-547)
        Event emitted after the call(s):
        - TokenFundsWithdrawn(_colID,_tokenType,msg.sender) (contracts/
          ↪ MADRouter1155.sol#549-553)
Reentrancy in MADRouter721.withdraw(address,ERC20) (contracts/
   ↪ MADRouter721.sol#374-442):
        External calls:
        - ERC721Minimal(_token).withdrawERC20(_erc20,recipient) (
          ↪ contracts/MADRouter721.sol#384-389)
        - ERC721Minimal(_token).withdraw(recipient) (contracts/
          ↪ MADRouter721.sol#384-389)
        Event emitted after the call(s):
```

```
        - TokenFundsWithdrawn(_colID,_tokenType,msg.sender) (contracts/
          ↪ MADRouter721.sol#391-395)
Reentrancy in MADRouter721.withdraw(address,ERC20) (contracts/
    ↪ MADRouter721.sol#374-442):
        External calls:
        - ERC721Minimal(_token).withdrawERC20(_erc20,recipient) (
          ↪ contracts/MADRouter721.sol#384-389)
        - ERC721Basic(_token).withdrawERC20(_erc20,recipient) (contracts/
          ↪ MADRouter721.sol#399-404)
        - ERC721Minimal(_token).withdraw(recipient) (contracts/
          ↪ MADRouter721.sol#384-389)
        - ERC721Basic(_token).withdraw(recipient) (contracts/MADRouter721
          ↪ .sol#399-404)
        Event emitted after the call(s):
        - TokenFundsWithdrawn(_colID,_tokenType,msg.sender) (contracts/
          ↪ MADRouter721.sol#406-410)
Reentrancy in MADRouter721.withdraw(address,ERC20) (contracts/
    ↪ MADRouter721.sol#374-442):
        External calls:
        - ERC721Minimal(_token).withdrawERC20(_erc20,recipient) (
          ↪ contracts/MADRouter721.sol#384-389)
        - ERC721Basic(_token).withdrawERC20(_erc20,recipient) (contracts/
          ↪ MADRouter721.sol#399-404)
        - ERC721Whitelist(_token).withdrawERC20(_erc20,recipient) (
          ↪ contracts/MADRouter721.sol#414-419)
        - ERC721Minimal(_token).withdraw(recipient) (contracts/
          ↪ MADRouter721.sol#384-389)
        - ERC721Basic(_token).withdraw(recipient) (contracts/MADRouter721
          ↪ .sol#399-404)
        - ERC721Whitelist(_token).withdraw(recipient) (contracts/
          ↪ MADRouter721.sol#414-419)
        Event emitted after the call(s):
        - TokenFundsWithdrawn(_colID,_tokenType,msg.sender) (contracts/
          ↪ MADRouter721.sol#421-425)
```

```
Reentrancy in MADRouter721.withdraw(address,ERC20) (contracts/
    ↪ MADRouter721.sol#374-442):
        External calls:
        - ERC721Minimal(_token).withdrawERC20(_erc20,recipient) (
            ↪ contracts/MADRouter721.sol#384-389)
        - ERC721Basic(_token).withdrawERC20(_erc20,recipient) (contracts/
            ↪ MADRouter721.sol#399-404)
        - ERC721Whitelist(_token).withdrawERC20(_erc20,recipient) (
            ↪ contracts/MADRouter721.sol#414-419)
        - ERC721Lazy(_token).withdrawERC20(_erc20,recipient) (contracts/
            ↪ MADRouter721.sol#429-434)
        - ERC721Minimal(_token).withdraw(recipient) (contracts/
            ↪ MADRouter721.sol#384-389)
        - ERC721Basic(_token).withdraw(recipient) (contracts/MADRouter721
            ↪ .sol#399-404)
        - ERC721Whitelist(_token).withdraw(recipient) (contracts/
            ↪ MADRouter721.sol#414-419)
        - ERC721Lazy(_token).withdraw(recipient) (contracts/MADRouter721.
            ↪ sol#429-434)
        Event emitted after the call(s):
        - TokenFundsWithdrawn(_colID,_tokenType,msg.sender) (contracts/
            ↪ MADRouter721.sol#436-440)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation
    ↪ #reentrancy-vulnerabilities-3

MADFactory1155.name() (contracts/MADFactory1155.sol#46-57) uses assembly
        - INLINE ASM (contracts/MADFactory1155.sol#52-56)
MADFactory1155.splitterCheck(string,address,address,uint256,uint256) (
    ↪ contracts/MADFactory1155.sol#130-310) uses assembly
        - INLINE ASM (contracts/MADFactory1155.sol#304-308)
MADFactory1155.setOwner(address) (contracts/MADFactory1155.sol#492-504)
    ↪ uses assembly
        - INLINE ASM (contracts/MADFactory1155.sol#499-501)
```

```
MADFactory1155.setMarket(address) (contracts/MADFactory1155.sol#508-515)
    ↪  uses assembly
        - INLINE ASM (contracts/MADFactory1155.sol#510-512)
MADFactory1155.setRouter(address) (contracts/MADFactory1155.sol#519-527)
    ↪  uses assembly
        - INLINE ASM (contracts/MADFactory1155.sol#522-524)
MADFactory1155.setSigner(address) (contracts/MADFactory1155.sol#531-539)
    ↪  uses assembly
        - INLINE ASM (contracts/MADFactory1155.sol#534-536)
MADFactory1155.typeChecker(bytes32) (contracts/MADFactory1155.sol
    ↪ #577-585) uses assembly
        - INLINE ASM (contracts/MADFactory1155.sol#581-584)
MADFactory1155._payeesBuffer(address,address) (contracts/MADFactory1155.
    ↪ sol#588-635) uses assembly
        - INLINE ASM (contracts/MADFactory1155.sol#595-634)
MADFactory1155._sharesBuffer(uint256,uint256) (contracts/MADFactory1155.
    ↪ sol#638-683) uses assembly
        - INLINE ASM (contracts/MADFactory1155.sol#643-682)
MADFactory1155.creatorAuth(address,address) (contracts/MADFactory1155.
    ↪ sol#686-708) uses assembly
        - INLINE ASM (contracts/MADFactory1155.sol#693)
MADFactory1155.creatorCheck(bytes32) (contracts/MADFactory1155.sol
    ↪ #711-734) uses assembly
        - INLINE ASM (contracts/MADFactory1155.sol#720-733)
MADFactory1155._isRouter() (contracts/MADFactory1155.sol#738-748) uses
    ↪ assembly
        - INLINE ASM (contracts/MADFactory1155.sol#740-747)
MADFactory1155._isMarket() (contracts/MADFactory1155.sol#752-759) uses
    ↪ assembly
        - INLINE ASM (contracts/MADFactory1155.sol#753-758)
MADFactory1155._limiter(uint8,address) (contracts/MADFactory1155.sol
    ↪ #761-775) uses assembly
        - INLINE ASM (contracts/MADFactory1155.sol#769-774)
```

```
MADFactory1155._royaltyLocker(uint256) (contracts/MADFactory1155.sol
   ↪ #777-789) uses assembly
       - INLINE ASM (contracts/MADFactory1155.sol#781-788)
MADFactory1155._userRender(address) (contracts/MADFactory1155.sol
   ↪ #795-806) uses assembly
       - INLINE ASM (contracts/MADFactory1155.sol#796-805)
MADFactory721.name() (contracts/MADFactory721.sol#45-56) uses assembly
       - INLINE ASM (contracts/MADFactory721.sol#51-55)
MADFactory721.splitterCheck(string,address,address,uint256,uint256) (
   ↪ contracts/MADFactory721.sol#134-314) uses assembly
       - INLINE ASM (contracts/MADFactory721.sol#308-312)
MADFactory721.setOwner(address) (contracts/MADFactory721.sol#504-516)
   ↪ uses assembly
       - INLINE ASM (contracts/MADFactory721.sol#511-513)
MADFactory721.setMarket(address) (contracts/MADFactory721.sol#520-528)
   ↪ uses assembly
       - INLINE ASM (contracts/MADFactory721.sol#523-525)
MADFactory721.setRouter(address) (contracts/MADFactory721.sol#532-540)
   ↪ uses assembly
       - INLINE ASM (contracts/MADFactory721.sol#535-537)
MADFactory721.setSigner(address) (contracts/MADFactory721.sol#544-552)
   ↪ uses assembly
       - INLINE ASM (contracts/MADFactory721.sol#547-549)
MADFactory721.typeChecker(bytes32) (contracts/MADFactory721.sol#590-598)
   ↪  uses assembly
       - INLINE ASM (contracts/MADFactory721.sol#594-597)
MADFactory721._payeesBuffer(address,address) (contracts/MADFactory721.
   ↪ sol#601-648) uses assembly
       - INLINE ASM (contracts/MADFactory721.sol#608-647)
MADFactory721._sharesBuffer(uint256,uint256) (contracts/MADFactory721.
   ↪ sol#651-696) uses assembly
       - INLINE ASM (contracts/MADFactory721.sol#656-695)
MADFactory721.creatorAuth(address,address) (contracts/MADFactory721.sol
   ↪ #700-722) uses assembly
```

```
        - INLINE ASM (contracts/MADFactory721.sol#707)
MADFactory721.creatorCheck(bytes32) (contracts/MADFactory721.sol
    ↪ #725-748) uses assembly
        - INLINE ASM (contracts/MADFactory721.sol#734-747)
MADFactory721._isRouter() (contracts/MADFactory721.sol#752-762) uses
    ↪ assembly
        - INLINE ASM (contracts/MADFactory721.sol#754-761)
MADFactory721._isMarket() (contracts/MADFactory721.sol#766-773) uses
    ↪ assembly
        - INLINE ASM (contracts/MADFactory721.sol#767-772)
MADFactory721._limiter(uint8,address) (contracts/MADFactory721.sol
    ↪ #775-789) uses assembly
        - INLINE ASM (contracts/MADFactory721.sol#783-788)
MADFactory721._royaltyLocker(uint256) (contracts/MADFactory721.sol
    ↪ #791-803) uses assembly
        - INLINE ASM (contracts/MADFactory721.sol#795-802)
MADFactory721._userRender(address) (contracts/MADFactory721.sol#809-820)
    ↪  uses assembly
        - INLINE ASM (contracts/MADFactory721.sol#810-819)
MADMarketplace1155.name() (contracts/MADMarketplace1155.sol#25-36) uses
    ↪ assembly
        - INLINE ASM (contracts/MADMarketplace1155.sol#31-35)
MADMarketplace1155.bid(bytes32) (contracts/MADMarketplace1155.sol
    ↪ #190-276) uses assembly
        - INLINE ASM (contracts/MADMarketplace1155.sol#221-238)
MADMarketplace1155.setFactory(FactoryVerifier) (contracts/
    ↪ MADMarketplace1155.sol#471-480) uses assembly
        - INLINE ASM (contracts/MADMarketplace1155.sol#475-478)
MADMarketplace1155.setFees(uint256,uint256) (contracts/
    ↪ MADMarketplace1155.sol#482-498) uses assembly
        - INLINE ASM (contracts/MADMarketplace1155.sol#492-495)
MADMarketplace1155.updateSettings(uint256,uint256,uint256,uint256) (
    ↪ contracts/MADMarketplace1155.sol#507-541) uses assembly
        - INLINE ASM (contracts/MADMarketplace1155.sol#525-533)
```

```
MADMarketplace1155.setRecipient(address) (contracts/MADMarketplace1155.
    ↪ sol#571-586) uses assembly
        - INLINE ASM (contracts/MADMarketplace1155.sol#581-583)
MADMarketplace1155.setOwner(address) (contracts/MADMarketplace1155.sol
    ↪ #589-602) uses assembly
        - INLINE ASM (contracts/MADMarketplace1155.sol#597-599)
MADMarketplace1155.interfaceCheck(address,bytes4) (contracts/
    ↪ MADMarketplace1155.sol#856-881) uses assembly
        - INLINE ASM (contracts/MADMarketplace1155.sol#867-878)
MADMarketplace1155._feeResolver(uint256,uint256,uint256) (contracts/
    ↪ MADMarketplace1155.sol#1102-1126) uses assembly
        - INLINE ASM (contracts/MADMarketplace1155.sol#1107-1125)
MADMarketplace1155._exceedsMaxEP(uint256,uint256) (contracts/
    ↪ MADMarketplace1155.sol#1132-1150) uses assembly
        - INLINE ASM (contracts/MADMarketplace1155.sol#1136-1149)
MADMarketplace1155._isBidderOrSeller(address,address) (contracts/
    ↪ MADMarketplace1155.sol#1152-1168) uses assembly
        - INLINE ASM (contracts/MADMarketplace1155.sol#1156-1167)
MADMarketplace1155._makeOrderChecks(uint256,uint256) (contracts/
    ↪ MADMarketplace1155.sol#1170-1209) uses assembly
        - INLINE ASM (contracts/MADMarketplace1155.sol#1174-1208)
MADMarketplace1155._cancelOrderChecks(address,bool,uint256) (contracts/
    ↪ MADMarketplace1155.sol#1211-1233) uses assembly
        - INLINE ASM (contracts/MADMarketplace1155.sol#1216-1232)
MADMarketplace1155._bidChecks(uint8,uint256,address,uint256,uint256,
    ↪ uint256) (contracts/MADMarketplace1155.sol#1235-1291) uses
    ↪ assembly
        - INLINE ASM (contracts/MADMarketplace1155.sol#1243-1290)
MADMarketplace1155._buyChecks(uint256,uint8,bool) (contracts/
    ↪ MADMarketplace1155.sol#1293-1323) uses assembly
        - INLINE ASM (contracts/MADMarketplace1155.sol#1298-1322)
MADMarketplace1155._claimChecks(bool,uint8,uint256) (contracts/
    ↪ MADMarketplace1155.sol#1325-1350) uses assembly
        - INLINE ASM (contracts/MADMarketplace1155.sol#1330-1349)
```

```
MADMarketplace1155.getCurrentPrice(bytes32) (contracts/
    ↪ MADMarketplace1155.sol#1359-1425) uses assembly
        - INLINE ASM (contracts/MADMarketplace1155.sol#1366-1424)
MADMarketplace721.name() (contracts/MADMarketplace721.sol#25-36) uses
    ↪ assembly
        - INLINE ASM (contracts/MADMarketplace721.sol#31-35)
MADMarketplace721.bid(bytes32) (contracts/MADMarketplace721.sol#169-258)
    ↪  uses assembly
        - INLINE ASM (contracts/MADMarketplace721.sol#200-217)
MADMarketplace721.setFactory(FactoryVerifier) (contracts/
    ↪ MADMarketplace721.sol#451-460) uses assembly
        - INLINE ASM (contracts/MADMarketplace721.sol#455-458)
MADMarketplace721.setFees(uint256,uint256) (contracts/MADMarketplace721.
    ↪ sol#462-477) uses assembly
        - INLINE ASM (contracts/MADMarketplace721.sol#471-474)
MADMarketplace721.updateSettings(uint256,uint256,uint256,uint256) (
    ↪ contracts/MADMarketplace721.sol#485-519) uses assembly
        - INLINE ASM (contracts/MADMarketplace721.sol#503-511)
MADMarketplace721.setRecipient(address) (contracts/MADMarketplace721.sol
    ↪ #544-559) uses assembly
        - INLINE ASM (contracts/MADMarketplace721.sol#554-556)
MADMarketplace721.setOwner(address) (contracts/MADMarketplace721.sol
    ↪ #562-574) uses assembly
        - INLINE ASM (contracts/MADMarketplace721.sol#569-571)
MADMarketplace721.interfaceCheck(address,bytes4) (contracts/
    ↪ MADMarketplace721.sol#811-836) uses assembly
        - INLINE ASM (contracts/MADMarketplace721.sol#822-833)
MADMarketplace721._feeResolver(uint256,uint256) (contracts/
    ↪ MADMarketplace721.sol#1034-1054) uses assembly
        - INLINE ASM (contracts/MADMarketplace721.sol#1038-1053)
MADMarketplace721._exceedsMaxEP(uint256,uint256) (contracts/
    ↪ MADMarketplace721.sol#1060-1078) uses assembly
        - INLINE ASM (contracts/MADMarketplace721.sol#1064-1077)
```

```
MADMarketplace721._isBidderOrSeller(address,address) (contracts/
   ↪ MADMarketplace721.sol#1080-1096) uses assembly
      - INLINE ASM (contracts/MADMarketplace721.sol#1084-1095)
MADMarketplace721._makeOrderChecks(uint256,uint256) (contracts/
   ↪ MADMarketplace721.sol#1098-1137) uses assembly
      - INLINE ASM (contracts/MADMarketplace721.sol#1102-1136)
MADMarketplace721._cancelOrderChecks(address,bool,uint256) (contracts/
   ↪ MADMarketplace721.sol#1139-1161) uses assembly
      - INLINE ASM (contracts/MADMarketplace721.sol#1144-1160)
MADMarketplace721._bidChecks(uint8,uint256,address,uint256,uint256,
   ↪ uint256) (contracts/MADMarketplace721.sol#1163-1219) uses
   ↪ assembly
      - INLINE ASM (contracts/MADMarketplace721.sol#1171-1218)
MADMarketplace721._buyChecks(uint256,uint8,bool) (contracts/
   ↪ MADMarketplace721.sol#1221-1251) uses assembly
      - INLINE ASM (contracts/MADMarketplace721.sol#1226-1250)
MADMarketplace721._claimChecks(bool,uint8,uint256) (contracts/
   ↪ MADMarketplace721.sol#1253-1278) uses assembly
      - INLINE ASM (contracts/MADMarketplace721.sol#1258-1277)
MADMarketplace721.getCurrentPrice(bytes32) (contracts/MADMarketplace721.
   ↪ sol#1287-1353) uses assembly
      - INLINE ASM (contracts/MADMarketplace721.sol#1294-1352)
MADRouter1155.name() (contracts/MADRouter1155.sol#61-72) uses assembly
      - INLINE ASM (contracts/MADRouter1155.sol#67-71)
MADRouter1155.setRecipient(address) (contracts/MADRouter1155.sol
   ↪ #107-115) uses assembly
      - INLINE ASM (contracts/MADRouter1155.sol#110-112)
MADRouter1155.feeLookup(bytes4) (contracts/MADRouter1155.sol#564-588)
   ↪ uses assembly
      - INLINE ASM (contracts/MADRouter1155.sol#570-587)
MADRouter1155._paymentCheck(bytes4) (contracts/MADRouter1155.sol
   ↪ #664-684) uses assembly
      - INLINE ASM (contracts/MADRouter1155.sol#671-676)
```

```
MADRouter1155.setOwner(address) (contracts/MADRouter1155.sol#693-705)
    ↪ uses assembly
        - INLINE ASM (contracts/MADRouter1155.sol#700-702)
MADRouter1155.setFees(uint256,uint256) (contracts/MADRouter1155.sol
    ↪ #725-741) uses assembly
        - INLINE ASM (contracts/MADRouter1155.sol#735-738)
MADRouter721.name() (contracts/MADRouter721.sol#60-71) uses assembly
        - INLINE ASM (contracts/MADRouter721.sol#66-70)
MADRouter721.feeLookup(bytes4) (contracts/MADRouter721.sol#451-475) uses
    ↪  assembly
        - INLINE ASM (contracts/MADRouter721.sol#457-474)
MADRouter721._paymentCheck(bytes4) (contracts/MADRouter721.sol#549-569)
    ↪ uses assembly
        - INLINE ASM (contracts/MADRouter721.sol#556-561)
MADRouter721.setRecipient(address) (contracts/MADRouter721.sol#577-585)
    ↪ uses assembly
        - INLINE ASM (contracts/MADRouter721.sol#580-582)
MADRouter721.setOwner(address) (contracts/MADRouter721.sol#590-602) uses
    ↪  assembly
        - INLINE ASM (contracts/MADRouter721.sol#597-599)
MADRouter721.setFees(uint256,uint256) (contracts/MADRouter721.sol
    ↪ #622-633) uses assembly
        - INLINE ASM (contracts/MADRouter721.sol#627-630)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation
    ↪ #assembly-usage

MADMarketplace1155.buy(bytes32) (contracts/MADMarketplace1155.sol
    ↪ #281-361) compares to a boolean constant:
        -ERC165Check(address(order.token)) && interfaceCheck(address(
            ↪ order.token),0x2a55205a) == true (contracts/
            ↪ MADMarketplace1155.sol#337-342)
MADMarketplace1155.claim(bytes32) (contracts/MADMarketplace1155.sol
    ↪ #366-427) compares to a boolean constant:
        -ERC165Check(address(order.token)) && interfaceCheck(address(
           ↪ order.token),0x2a55205a) == true (contracts/
           ↪ MADMarketplace1155.sol#403-408)
MADMarketplace1155.claim(bytes32) (contracts/MADMarketplace1155.sol
    ↪ #366-427) compares to a boolean constant:
        -! feeSelector[key][order.tokenId][order.amount] &&
           ↪ MADFactory1155.creatorAuth(address(order.token),order.
           ↪ seller) == true (contracts/MADMarketplace1155.sol#384-389)
MADMarketplace721.buy(bytes32) (contracts/MADMarketplace721.sol#263-348)
    ↪  compares to a boolean constant:
        -! feeSelector[key][order.tokenId] && MADFactory721.creatorAuth(
           ↪ address(order.token),order.seller) == true (contracts/
           ↪ MADMarketplace721.sol#303-308)
MADMarketplace721.buy(bytes32) (contracts/MADMarketplace721.sol#263-348)
    ↪  compares to a boolean constant:
        -ERC165Check(address(order.token)) && interfaceCheck(address(
           ↪ order.token),0x2a55205a) == true (contracts/
           ↪ MADMarketplace721.sol#322-327)
MADMarketplace721.claim(bytes32) (contracts/MADMarketplace721.sol
    ↪ #353-416) compares to a boolean constant:
        -ERC165Check(address(order.token)) && interfaceCheck(address(
           ↪ order.token),0x2a55205a) == true (contracts/
           ↪ MADMarketplace721.sol#390-395)
MADMarketplace721.claim(bytes32) (contracts/MADMarketplace721.sol
    ↪ #353-416) compares to a boolean constant:
        -! feeSelector[key][order.tokenId] && MADFactory721.creatorAuth(
           ↪ address(order.token),order.seller) == true (contracts/
           ↪ MADMarketplace721.sol#371-376)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation
    ↪ #boolean-equality

MADMarketplace1155.autoTransferFunds(address[]) (contracts/
    ↪ MADMarketplace1155.sol#630-653) has costly operations inside a
    ↪ loop:
        - totalOutbid = totalOutbid - outbid (contracts/
            ↪ MADMarketplace1155.sol#641)
MADMarketplace1155._withdrawOutbid(address,ERC20,uint256,uint160) (
    ↪ contracts/MADMarketplace1155.sol#688-740) has costly operations
    ↪ inside a loop:
        - totalOutbid -= amountIn (contracts/MADMarketplace1155.sol#706)
MADMarketplace721.autoTransferFunds(address[]) (contracts/
    ↪ MADMarketplace721.sol#607-632) has costly operations inside a
    ↪ loop:
        - totalOutbid = totalOutbid - outbid (contracts/MADMarketplace721
            ↪ .sol#617)
MADMarketplace721._withdrawOutbid(address,ERC20,uint256,uint160) (
    ↪ contracts/MADMarketplace721.sol#667-719) has costly operations
    ↪ inside a loop:
        - totalOutbid -= amountIn (contracts/MADMarketplace721.sol#685)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation
    ↪ #costly-operations-inside-a-loop

Pragma version>=0.5.0 (node_modules/@uniswap/v3-core/contracts/
    ↪ interfaces/callback/IUniswapV3SwapCallback.sol#2) allows old
    ↪ versions
Pragma version0.8.16 (contracts/EventsAndErrors.sol#3) necessitates a
    ↪ version too recent to be trusted. Consider deploying with
    ↪ 0.6.12/0.7.6/0.8.7
Pragma version0.8.16 (contracts/MAD.sol#3) necessitates a version too
    ↪ recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.16 (contracts/MADFactory1155.sol#3) necessitates a
    ↪ version too recent to be trusted. Consider deploying with
    ↪ 0.6.12/0.7.6/0.8.7
Pragma version0.8.16 (contracts/MADFactory721.sol#3) necessitates a
    ↪ version too recent to be trusted. Consider deploying with
    ↪ 0.6.12/0.7.6/0.8.7
Pragma version0.8.16 (contracts/MADMarketplace1155.sol#3) necessitates a
    ↪  version too recent to be trusted. Consider deploying with
    ↪ 0.6.12/0.7.6/0.8.7
Pragma version0.8.16 (contracts/MADMarketplace721.sol#3) necessitates a
    ↪ version too recent to be trusted. Consider deploying with
    ↪ 0.6.12/0.7.6/0.8.7
Pragma version0.8.16 (contracts/MADRouter1155.sol#3) necessitates a
    ↪ version too recent to be trusted. Consider deploying with
    ↪ 0.6.12/0.7.6/0.8.7
Pragma version0.8.16 (contracts/MADRouter721.sol#3) necessitates a
    ↪ version too recent to be trusted. Consider deploying with
    ↪ 0.6.12/0.7.6/0.8.7
Pragma version0.8.16 (contracts/Types.sol#3) necessitates a version too
    ↪ recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
solc-0.8.16 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation
    ↪ #incorrect-versions-of-solidity

Parameter MADFactory1155.splitterCheck(string,address,address,uint256,
    ↪ uint256)._splitterSalt (contracts/MADFactory1155.sol#131) is not
    ↪ in mixedCase
Parameter MADFactory1155.splitterCheck(string,address,address,uint256,
    ↪ uint256)._ambassador (contracts/MADFactory1155.sol#132) is not in
    ↪  mixedCase
Parameter MADFactory1155.splitterCheck(string,address,address,uint256,
    ↪ uint256)._project (contracts/MADFactory1155.sol#133) is not in
    ↪ mixedCase
Parameter MADFactory1155.splitterCheck(string,address,address,uint256,
    ↪ uint256)._ambShare (contracts/MADFactory1155.sol#134) is not in
    ↪ mixedCase
Parameter MADFactory1155.splitterCheck(string,address,address,uint256,
    ↪ uint256)._projectShare (contracts/MADFactory1155.sol#135) is not
    ↪ in mixedCase
Parameter MADFactory1155.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256)._tokenType (contracts/
    ↪ MADFactory1155.sol#331) is not in mixedCase
```

```
Parameter MADFactory1155.createCollection(uint8,string,string,string,
   ↪ uint256,uint256,string,address,uint256)._tokenSalt (contracts/
   ↪ MADFactory1155.sol#332) is not in mixedCase
Parameter MADFactory1155.createCollection(uint8,string,string,string,
   ↪ uint256,uint256,string,address,uint256)._name (contracts/
   ↪ MADFactory1155.sol#333) is not in mixedCase
Parameter MADFactory1155.createCollection(uint8,string,string,string,
   ↪ uint256,uint256,string,address,uint256)._symbol (contracts/
   ↪ MADFactory1155.sol#334) is not in mixedCase
Parameter MADFactory1155.createCollection(uint8,string,string,string,
   ↪ uint256,uint256,string,address,uint256)._price (contracts/
   ↪ MADFactory1155.sol#335) is not in mixedCase
Parameter MADFactory1155.createCollection(uint8,string,string,string,
   ↪ uint256,uint256,string,address,uint256)._maxSupply (contracts/
   ↪ MADFactory1155.sol#336) is not in mixedCase
Parameter MADFactory1155.createCollection(uint8,string,string,string,
   ↪ uint256,uint256,string,address,uint256)._uri (contracts/
   ↪ MADFactory1155.sol#337) is not in mixedCase
Parameter MADFactory1155.createCollection(uint8,string,string,string,
   ↪ uint256,uint256,string,address,uint256)._splitter (contracts/
   ↪ MADFactory1155.sol#338) is not in mixedCase
Parameter MADFactory1155.createCollection(uint8,string,string,string,
   ↪ uint256,uint256,string,address,uint256)._royalty (contracts/
   ↪ MADFactory1155.sol#339) is not in mixedCase
Parameter MADFactory1155.setMarket(address)._market (contracts/
   ↪ MADFactory1155.sol#508) is not in mixedCase
Parameter MADFactory1155.setRouter(address)._router (contracts/
   ↪ MADFactory1155.sol#519) is not in mixedCase
Parameter MADFactory1155.setSigner(address)._signer (contracts/
   ↪ MADFactory1155.sol#531) is not in mixedCase
Parameter MADFactory1155.getIDsLength(address)._user (contracts/
   ↪ MADFactory1155.sol#562) is not in mixedCase
Parameter MADFactory1155.getColID(address)._colAddress (contracts/
   ↪ MADFactory1155.sol#571) is not in mixedCase
```

```
Parameter MADFactory1155.typeChecker(bytes32)._colID (contracts/
    ↪ MADFactory1155.sol#577) is not in mixedCase
Parameter MADFactory1155.creatorAuth(address,address)._token (contracts/
    ↪ MADFactory1155.sol#686) is not in mixedCase
Parameter MADFactory1155.creatorAuth(address,address)._user (contracts/
    ↪ MADFactory1155.sol#686) is not in mixedCase
Parameter MADFactory1155.creatorCheck(bytes32)._colID (contracts/
    ↪ MADFactory1155.sol#711) is not in mixedCase
Parameter MADFactory1155.getDeployedAddr(string)._salt (contracts/
    ↪ MADFactory1155.sol#808) is not in mixedCase
Parameter MADFactory721.splitterCheck(string,address,address,uint256,
    ↪ uint256)._splitterSalt (contracts/MADFactory721.sol#135) is not
    ↪ in mixedCase
Parameter MADFactory721.splitterCheck(string,address,address,uint256,
    ↪ uint256)._ambassador (contracts/MADFactory721.sol#136) is not in
    ↪ mixedCase
Parameter MADFactory721.splitterCheck(string,address,address,uint256,
    ↪ uint256)._project (contracts/MADFactory721.sol#137) is not in
    ↪ mixedCase
Parameter MADFactory721.splitterCheck(string,address,address,uint256,
    ↪ uint256)._ambShare (contracts/MADFactory721.sol#138) is not in
    ↪ mixedCase
Parameter MADFactory721.splitterCheck(string,address,address,uint256,
    ↪ uint256)._projectShare (contracts/MADFactory721.sol#139) is not
    ↪ in mixedCase
Parameter MADFactory721.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256)._tokenType (contracts/
    ↪ MADFactory721.sol#335) is not in mixedCase
Parameter MADFactory721.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256)._tokenSalt (contracts/
    ↪ MADFactory721.sol#336) is not in mixedCase
Parameter MADFactory721.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256)._name (contracts/
    ↪ MADFactory721.sol#337) is not in mixedCase
```

```
Parameter MADFactory721.createCollection(uint8,string,string,string,
   ↪ uint256,uint256,string,address,uint256)._symbol (contracts/
   ↪ MADFactory721.sol#338) is not in mixedCase
Parameter MADFactory721.createCollection(uint8,string,string,string,
   ↪ uint256,uint256,string,address,uint256)._price (contracts/
   ↪ MADFactory721.sol#339) is not in mixedCase
Parameter MADFactory721.createCollection(uint8,string,string,string,
   ↪ uint256,uint256,string,address,uint256)._maxSupply (contracts/
   ↪ MADFactory721.sol#340) is not in mixedCase
Parameter MADFactory721.createCollection(uint8,string,string,string,
   ↪ uint256,uint256,string,address,uint256)._baseURI (contracts/
   ↪ MADFactory721.sol#341) is not in mixedCase
Parameter MADFactory721.createCollection(uint8,string,string,string,
   ↪ uint256,uint256,string,address,uint256)._splitter (contracts/
   ↪ MADFactory721.sol#342) is not in mixedCase
Parameter MADFactory721.createCollection(uint8,string,string,string,
   ↪ uint256,uint256,string,address,uint256)._royalty (contracts/
   ↪ MADFactory721.sol#343) is not in mixedCase
Parameter MADFactory721.setMarket(address)._market (contracts/
   ↪ MADFactory721.sol#520) is not in mixedCase
Parameter MADFactory721.setRouter(address)._router (contracts/
   ↪ MADFactory721.sol#532) is not in mixedCase
Parameter MADFactory721.setSigner(address)._signer (contracts/
   ↪ MADFactory721.sol#544) is not in mixedCase
Parameter MADFactory721.getIDsLength(address)._user (contracts/
   ↪ MADFactory721.sol#575) is not in mixedCase
Parameter MADFactory721.getColID(address)._colAddress (contracts/
   ↪ MADFactory721.sol#584) is not in mixedCase
Parameter MADFactory721.typeChecker(bytes32)._colID (contracts/
   ↪ MADFactory721.sol#590) is not in mixedCase
Parameter MADFactory721.creatorAuth(address,address)._token (contracts/
   ↪ MADFactory721.sol#700) is not in mixedCase
Parameter MADFactory721.creatorAuth(address,address)._user (contracts/
   ↪ MADFactory721.sol#700) is not in mixedCase
```

```
Parameter MADFactory721.creatorCheck(bytes32)._colID (contracts/
    ↪ MADFactory721.sol#725) is not in mixedCase
Parameter MADFactory721.getDeployedAddr(string)._salt (contracts/
    ↪ MADFactory721.sol#822) is not in mixedCase
Parameter MADMarketplace1155.fixedPrice(IERC1155,uint256,uint256,uint256
    ↪ ,uint256)._token (contracts/MADMarketplace1155.sol#126) is not in
    ↪  mixedCase
Parameter MADMarketplace1155.fixedPrice(IERC1155,uint256,uint256,uint256
    ↪ ,uint256)._id (contracts/MADMarketplace1155.sol#127) is not in
    ↪ mixedCase
Parameter MADMarketplace1155.fixedPrice(IERC1155,uint256,uint256,uint256
    ↪ ,uint256)._amount (contracts/MADMarketplace1155.sol#128) is not
    ↪ in mixedCase
Parameter MADMarketplace1155.fixedPrice(IERC1155,uint256,uint256,uint256
    ↪ ,uint256)._price (contracts/MADMarketplace1155.sol#129) is not in
    ↪  mixedCase
Parameter MADMarketplace1155.fixedPrice(IERC1155,uint256,uint256,uint256
    ↪ ,uint256)._endTime (contracts/MADMarketplace1155.sol#130) is not
    ↪ in mixedCase
Parameter MADMarketplace1155.dutchAuction(IERC1155,uint256,uint256,
    ↪ uint256,uint256,uint256)._token (contracts/MADMarketplace1155.sol
    ↪ #146) is not in mixedCase
Parameter MADMarketplace1155.dutchAuction(IERC1155,uint256,uint256,
    ↪ uint256,uint256,uint256)._id (contracts/MADMarketplace1155.sol
    ↪ #147) is not in mixedCase
Parameter MADMarketplace1155.dutchAuction(IERC1155,uint256,uint256,
    ↪ uint256,uint256,uint256)._amount (contracts/MADMarketplace1155.
    ↪ sol#148) is not in mixedCase
Parameter MADMarketplace1155.dutchAuction(IERC1155,uint256,uint256,
    ↪ uint256,uint256,uint256)._startPrice (contracts/
    ↪ MADMarketplace1155.sol#149) is not in mixedCase
Parameter MADMarketplace1155.dutchAuction(IERC1155,uint256,uint256,
    ↪ uint256,uint256,uint256)._endPrice (contracts/MADMarketplace1155.
    ↪ sol#150) is not in mixedCase
```

```
Parameter MADMarketplace1155.dutchAuction(IERC1155,uint256,uint256,
    ↪ uint256,uint256,uint256)._endTime (contracts/MADMarketplace1155.
    ↪ sol#151) is not in mixedCase
Parameter MADMarketplace1155.englishAuction(IERC1155,uint256,uint256,
    ↪ uint256,uint256)._token (contracts/MADMarketplace1155.sol#168) is
    ↪  not in mixedCase
Parameter MADMarketplace1155.englishAuction(IERC1155,uint256,uint256,
    ↪ uint256,uint256)._id (contracts/MADMarketplace1155.sol#169) is
    ↪ not in mixedCase
Parameter MADMarketplace1155.englishAuction(IERC1155,uint256,uint256,
    ↪ uint256,uint256)._amount (contracts/MADMarketplace1155.sol#170)
    ↪ is not in mixedCase
Parameter MADMarketplace1155.englishAuction(IERC1155,uint256,uint256,
    ↪ uint256,uint256)._startPrice (contracts/MADMarketplace1155.sol
    ↪ #171) is not in mixedCase
Parameter MADMarketplace1155.englishAuction(IERC1155,uint256,uint256,
    ↪ uint256,uint256)._endTime (contracts/MADMarketplace1155.sol#172)
    ↪ is not in mixedCase
Parameter MADMarketplace1155.bid(bytes32)._order (contracts/
    ↪ MADMarketplace1155.sol#190) is not in mixedCase
Parameter MADMarketplace1155.buy(bytes32)._order (contracts/
    ↪ MADMarketplace1155.sol#281) is not in mixedCase
Parameter MADMarketplace1155.claim(bytes32)._order (contracts/
    ↪ MADMarketplace1155.sol#366) is not in mixedCase
Parameter MADMarketplace1155.cancelOrder(bytes32)._order (contracts/
    ↪ MADMarketplace1155.sol#432) is not in mixedCase
Parameter MADMarketplace1155.setFactory(FactoryVerifier)._factory (
    ↪ contracts/MADMarketplace1155.sol#471) is not in mixedCase
Parameter MADMarketplace1155.setFees(uint256,uint256)._feeVal2 (
    ↪ contracts/MADMarketplace1155.sol#482) is not in mixedCase
Parameter MADMarketplace1155.setFees(uint256,uint256)._feeVal3 (
    ↪ contracts/MADMarketplace1155.sol#482) is not in mixedCase
Parameter MADMarketplace1155.updateSettings(uint256,uint256,uint256,
    ↪ uint256)._minAuctionIncrement (contracts/MADMarketplace1155.sol
    ↪ #508) is not in mixedCase
Parameter MADMarketplace1155.updateSettings(uint256,uint256,uint256,
    ↪ uint256)._minOrderDuration (contracts/MADMarketplace1155.sol#509)
    ↪  is not in mixedCase
Parameter MADMarketplace1155.updateSettings(uint256,uint256,uint256,
    ↪ uint256)._minBidValue (contracts/MADMarketplace1155.sol#510) is
    ↪ not in mixedCase
Parameter MADMarketplace1155.updateSettings(uint256,uint256,uint256,
    ↪ uint256)._maxOrderDuration (contracts/MADMarketplace1155.sol#511)
    ↪  is not in mixedCase
Parameter MADMarketplace1155.setRecipient(address)._recipient (contracts
    ↪ /MADMarketplace1155.sol#571) is not in mixedCase
Parameter MADMarketplace1155.withdrawERC20(ERC20)._token (contracts/
    ↪ MADMarketplace1155.sol#613) is not in mixedCase
Parameter MADMarketplace1155.withdrawOutbid(ERC20,uint256,uint160).
    ↪ _token (contracts/MADMarketplace1155.sol#676) is not in mixedCase
Parameter MADMarketplace1155.delOrder(bytes32,IERC1155,uint256,uint256,
    ↪ address)._token (contracts/MADMarketplace1155.sol#755) is not in
    ↪ mixedCase
Parameter MADMarketplace1155.delOrder(bytes32,IERC1155,uint256,uint256,
    ↪ address)._id (contracts/MADMarketplace1155.sol#756) is not in
    ↪ mixedCase
Parameter MADMarketplace1155.delOrder(bytes32,IERC1155,uint256,uint256,
    ↪ address)._amount (contracts/MADMarketplace1155.sol#757) is not in
    ↪  mixedCase
Parameter MADMarketplace1155.delOrder(bytes32,IERC1155,uint256,uint256,
    ↪ address)._seller (contracts/MADMarketplace1155.sol#758) is not in
    ↪  mixedCase
Function MADMarketplace1155.ERC165Check(address) (contracts/
    ↪ MADMarketplace1155.sol#886-894) is not in mixedCase
Parameter MADMarketplace1155.getCurrentPrice(bytes32)._order (contracts/
    ↪ MADMarketplace1155.sol#1359) is not in mixedCase
Parameter MADMarketplace1155.tokenOrderLength(IERC1155,uint256,uint256).
    ↪ _token (contracts/MADMarketplace1155.sol#1433) is not in
    ↪ mixedCase
Parameter MADMarketplace1155.tokenOrderLength(IERC1155,uint256,uint256).
    ↪ _id (contracts/MADMarketplace1155.sol#1434) is not in mixedCase
Parameter MADMarketplace1155.tokenOrderLength(IERC1155,uint256,uint256).
    ↪ _amount (contracts/MADMarketplace1155.sol#1435) is not in
    ↪ mixedCase
Parameter MADMarketplace1155.sellerOrderLength(address)._seller (
    ↪ contracts/MADMarketplace1155.sol#1445) is not in mixedCase
Constant MADMarketplace1155.feeTier (contracts/MADMarketplace1155.sol
    ↪ #42) is not in UPPER_CASE_WITH_UNDERSCORES
Constant MADMarketplace1155.basisPoints (contracts/MADMarketplace1155.
    ↪ sol#51) is not in UPPER_CASE_WITH_UNDERSCORES
Variable MADMarketplace1155.MADFactory1155 (contracts/MADMarketplace1155
    ↪ .sol#76) is not in mixedCase
Parameter MADMarketplace721.fixedPrice(IERC721,uint256,uint256,uint256).
    ↪ _token (contracts/MADMarketplace721.sol#125) is not in mixedCase
Parameter MADMarketplace721.fixedPrice(IERC721,uint256,uint256,uint256).
    ↪ _id (contracts/MADMarketplace721.sol#126) is not in mixedCase
Parameter MADMarketplace721.fixedPrice(IERC721,uint256,uint256,uint256).
    ↪ _price (contracts/MADMarketplace721.sol#127) is not in mixedCase
Parameter MADMarketplace721.fixedPrice(IERC721,uint256,uint256,uint256).
    ↪ _endTime (contracts/MADMarketplace721.sol#128) is not in
    ↪ mixedCase
Parameter MADMarketplace721.dutchAuction(IERC721,uint256,uint256,uint256
    ↪ ,uint256)._token (contracts/MADMarketplace721.sol#136) is not in
    ↪ mixedCase
Parameter MADMarketplace721.dutchAuction(IERC721,uint256,uint256,uint256
    ↪ ,uint256)._id (contracts/MADMarketplace721.sol#137) is not in
    ↪ mixedCase
Parameter MADMarketplace721.dutchAuction(IERC721,uint256,uint256,uint256
    ↪ ,uint256)._startPrice (contracts/MADMarketplace721.sol#138) is
    ↪ not in mixedCase
Parameter MADMarketplace721.dutchAuction(IERC721,uint256,uint256,uint256
    ↪ ,uint256)._endPrice (contracts/MADMarketplace721.sol#139) is not
    ↪ in mixedCase
Parameter MADMarketplace721.dutchAuction(IERC721,uint256,uint256,uint256
    ↪ ,uint256)._endTime (contracts/MADMarketplace721.sol#140) is not
    ↪ in mixedCase
Parameter MADMarketplace721.englishAuction(IERC721,uint256,uint256,
    ↪ uint256)._token (contracts/MADMarketplace721.sol#156) is not in
    ↪ mixedCase
Parameter MADMarketplace721.englishAuction(IERC721,uint256,uint256,
    ↪ uint256)._id (contracts/MADMarketplace721.sol#157) is not in
    ↪ mixedCase
Parameter MADMarketplace721.englishAuction(IERC721,uint256,uint256,
    ↪ uint256)._startPrice (contracts/MADMarketplace721.sol#158) is not
    ↪  in mixedCase
Parameter MADMarketplace721.englishAuction(IERC721,uint256,uint256,
    ↪ uint256)._endTime (contracts/MADMarketplace721.sol#159) is not in
    ↪  mixedCase
Parameter MADMarketplace721.bid(bytes32)._order (contracts/
    ↪ MADMarketplace721.sol#169) is not in mixedCase
Parameter MADMarketplace721.buy(bytes32)._order (contracts/
    ↪ MADMarketplace721.sol#263) is not in mixedCase
Parameter MADMarketplace721.claim(bytes32)._order (contracts/
    ↪ MADMarketplace721.sol#353) is not in mixedCase
Parameter MADMarketplace721.cancelOrder(bytes32)._order (contracts/
    ↪ MADMarketplace721.sol#421) is not in mixedCase
Parameter MADMarketplace721.setFactory(FactoryVerifier)._factory (
    ↪ contracts/MADMarketplace721.sol#451) is not in mixedCase
Parameter MADMarketplace721.setFees(uint256,uint256)._feeVal2 (contracts
    ↪ /MADMarketplace721.sol#462) is not in mixedCase
Parameter MADMarketplace721.setFees(uint256,uint256)._feeVal3 (contracts
    ↪ /MADMarketplace721.sol#462) is not in mixedCase
Parameter MADMarketplace721.updateSettings(uint256,uint256,uint256,
    ↪ uint256)._minAuctionIncrement (contracts/MADMarketplace721.sol
    ↪ #486) is not in mixedCase
```

```
Parameter MADMarketplace721.updateSettings(uint256,uint256,uint256,
    ↪ uint256)._minOrderDuration (contracts/MADMarketplace721.sol#487)
    ↪ is not in mixedCase
Parameter MADMarketplace721.updateSettings(uint256,uint256,uint256,
    ↪ uint256)._minBidValue (contracts/MADMarketplace721.sol#488) is
    ↪ not in mixedCase
Parameter MADMarketplace721.updateSettings(uint256,uint256,uint256,
    ↪ uint256)._maxOrderDuration (contracts/MADMarketplace721.sol#489)
    ↪ is not in mixedCase
Parameter MADMarketplace721.setRecipient(address)._recipient (contracts/
    ↪ MADMarketplace721.sol#544) is not in mixedCase
Parameter MADMarketplace721.withdrawERC20(ERC20)._token (contracts/
    ↪ MADMarketplace721.sol#588) is not in mixedCase
Parameter MADMarketplace721.withdrawOutbid(ERC20,uint256,uint160)._token
    ↪  (contracts/MADMarketplace721.sol#655) is not in mixedCase
Parameter MADMarketplace721.delOrder(bytes32,IERC721,uint256,address).
    ↪ _token (contracts/MADMarketplace721.sol#734) is not in mixedCase
Parameter MADMarketplace721.delOrder(bytes32,IERC721,uint256,address).
    ↪ _id (contracts/MADMarketplace721.sol#735) is not in mixedCase
Parameter MADMarketplace721.delOrder(bytes32,IERC721,uint256,address).
    ↪ _seller (contracts/MADMarketplace721.sol#736) is not in mixedCase
Function MADMarketplace721.ERC165Check(address) (contracts/
    ↪ MADMarketplace721.sol#841-849) is not in mixedCase
Parameter MADMarketplace721.getCurrentPrice(bytes32)._order (contracts/
    ↪ MADMarketplace721.sol#1287) is not in mixedCase
Parameter MADMarketplace721.tokenOrderLength(IERC721,uint256)._token (
    ↪ contracts/MADMarketplace721.sol#1360) is not in mixedCase
Parameter MADMarketplace721.tokenOrderLength(IERC721,uint256)._id (
    ↪ contracts/MADMarketplace721.sol#1360) is not in mixedCase
Parameter MADMarketplace721.sellerOrderLength(address)._seller (
    ↪ contracts/MADMarketplace721.sol#1373) is not in mixedCase
Constant MADMarketplace721.feeTier (contracts/MADMarketplace721.sol#42)
    ↪ is not in UPPER_CASE_WITH_UNDERSCORES
```

```
Constant MADMarketplace721.basisPoints (contracts/MADMarketplace721.sol
   ↪ #51) is not in UPPER_CASE_WITH_UNDERSCORES
Variable MADMarketplace721.MADFactory721 (contracts/MADMarketplace721.
   ↪ sol#76) is not in mixedCase
Parameter MADRouter1155.setRecipient(address)._recipient (contracts/
   ↪ MADRouter1155.sol#107) is not in mixedCase
Parameter MADRouter1155.setURI(address,string)._token (contracts/
   ↪ MADRouter1155.sol#133) is not in mixedCase
Parameter MADRouter1155.setURI(address,string)._uri (contracts/
   ↪ MADRouter1155.sol#133) is not in mixedCase
Parameter MADRouter1155.setURILock(address)._token (contracts/
   ↪ MADRouter1155.sol#164) is not in mixedCase
Parameter MADRouter1155.setMintState(address,bool,uint8)._token (
   ↪ contracts/MADRouter1155.sol#195) is not in mixedCase
Parameter MADRouter1155.setMintState(address,bool,uint8)._state (
   ↪ contracts/MADRouter1155.sol#196) is not in mixedCase
Parameter MADRouter1155.setMintState(address,bool,uint8)._stateType (
   ↪ contracts/MADRouter1155.sol#197) is not in mixedCase
Parameter MADRouter1155.whitelistSettings(address,uint256,uint256,
   ↪ bytes32)._token (contracts/MADRouter1155.sol#228) is not in
   ↪ mixedCase
Parameter MADRouter1155.whitelistSettings(address,uint256,uint256,
   ↪ bytes32)._price (contracts/MADRouter1155.sol#229) is not in
   ↪ mixedCase
Parameter MADRouter1155.whitelistSettings(address,uint256,uint256,
   ↪ bytes32)._supply (contracts/MADRouter1155.sol#230) is not in
   ↪ mixedCase
Parameter MADRouter1155.whitelistSettings(address,uint256,uint256,
   ↪ bytes32)._root (contracts/MADRouter1155.sol#231) is not in
   ↪ mixedCase
Parameter MADRouter1155.freeSettings(address,uint256,uint256,bytes32).
   ↪ _token (contracts/MADRouter1155.sol#251) is not in mixedCase
Parameter MADRouter1155.freeSettings(address,uint256,uint256,bytes32).
   ↪ _freeAmount (contracts/MADRouter1155.sol#252) is not in mixedCase
```

```
Parameter MADRouter1155.freeSettings(address,uint256,uint256,bytes32).
    ↪ _maxFree (contracts/MADRouter1155.sol#253) is not in mixedCase
Parameter MADRouter1155.freeSettings(address,uint256,uint256,bytes32).
    ↪ _claimRoot (contracts/MADRouter1155.sol#254) is not in mixedCase
Parameter MADRouter1155.minimalSafeMint(address,address,uint256)._token
    ↪ (contracts/MADRouter1155.sol#276) is not in mixedCase
Parameter MADRouter1155.minimalSafeMint(address,address,uint256)._to (
    ↪ contracts/MADRouter1155.sol#277) is not in mixedCase
Parameter MADRouter1155.basicMintTo(address,address,uint256,uint256[]).
    ↪ _token (contracts/MADRouter1155.sol#297) is not in mixedCase
Parameter MADRouter1155.basicMintTo(address,address,uint256,uint256[]).
    ↪ _to (contracts/MADRouter1155.sol#298) is not in mixedCase
Parameter MADRouter1155.basicMintTo(address,address,uint256,uint256[]).
    ↪ _amount (contracts/MADRouter1155.sol#299) is not in mixedCase
Parameter MADRouter1155.basicMintTo(address,address,uint256,uint256[]).
    ↪ _balances (contracts/MADRouter1155.sol#300) is not in mixedCase
Parameter MADRouter1155.basicMintBatchTo(address,address,uint256[],
    ↪ uint256[])._token (contracts/MADRouter1155.sol#319) is not in
    ↪ mixedCase
Parameter MADRouter1155.basicMintBatchTo(address,address,uint256[],
    ↪ uint256[])._to (contracts/MADRouter1155.sol#320) is not in
    ↪ mixedCase
Parameter MADRouter1155.basicMintBatchTo(address,address,uint256[],
    ↪ uint256[])._ids (contracts/MADRouter1155.sol#321) is not in
    ↪ mixedCase
Parameter MADRouter1155.basicMintBatchTo(address,address,uint256[],
    ↪ uint256[])._balances (contracts/MADRouter1155.sol#322) is not in
    ↪ mixedCase
Parameter MADRouter1155.creatorMint(address,uint256,uint256[],uint256).
    ↪ _token (contracts/MADRouter1155.sol#342) is not in mixedCase
Parameter MADRouter1155.creatorMint(address,uint256,uint256[],uint256).
    ↪ _amount (contracts/MADRouter1155.sol#343) is not in mixedCase
Parameter MADRouter1155.creatorMint(address,uint256,uint256[],uint256).
    ↪ _balances (contracts/MADRouter1155.sol#344) is not in mixedCase
```

```
Parameter MADRouter1155.creatorBatchMint(address,uint256[],uint256[],
   ↪ uint256)._token (contracts/MADRouter1155.sol#363) is not in
   ↪ mixedCase
Parameter MADRouter1155.creatorBatchMint(address,uint256[],uint256[],
   ↪ uint256)._ids (contracts/MADRouter1155.sol#364) is not in
   ↪ mixedCase
Parameter MADRouter1155.creatorBatchMint(address,uint256[],uint256[],
   ↪ uint256)._balances (contracts/MADRouter1155.sol#365) is not in
   ↪ mixedCase
Parameter MADRouter1155.gift(address,address[],uint256[],uint256)._token
   ↪  (contracts/MADRouter1155.sol#384) is not in mixedCase
Parameter MADRouter1155.gift(address,address[],uint256[],uint256).
   ↪ _addresses (contracts/MADRouter1155.sol#385) is not in mixedCase
Parameter MADRouter1155.gift(address,address[],uint256[],uint256).
   ↪ _balances (contracts/MADRouter1155.sol#386) is not in mixedCase
Parameter MADRouter1155.burn(address,uint256[],address[],uint256[]).
   ↪ _token (contracts/MADRouter1155.sol#411) is not in mixedCase
Parameter MADRouter1155.burn(address,uint256[],address[],uint256[])._ids
   ↪  (contracts/MADRouter1155.sol#412) is not in mixedCase
Parameter MADRouter1155.burn(address,uint256[],address[],uint256[]).
   ↪ _amount (contracts/MADRouter1155.sol#414) is not in mixedCase
Parameter MADRouter1155.batchBurn(address,address,uint256[],uint256[]).
   ↪ _token (contracts/MADRouter1155.sol#454) is not in mixedCase
Parameter MADRouter1155.batchBurn(address,address,uint256[],uint256[]).
   ↪ _from (contracts/MADRouter1155.sol#455) is not in mixedCase
Parameter MADRouter1155.batchBurn(address,address,uint256[],uint256[]).
   ↪ _ids (contracts/MADRouter1155.sol#456) is not in mixedCase
Parameter MADRouter1155.batchBurn(address,address,uint256[],uint256[]).
   ↪ _balances (contracts/MADRouter1155.sol#457) is not in mixedCase
Parameter MADRouter1155.withdraw(address,ERC20)._token (contracts/
   ↪ MADRouter1155.sol#487) is not in mixedCase
Parameter MADRouter1155.withdraw(address,ERC20)._erc20 (contracts/
   ↪ MADRouter1155.sol#487) is not in mixedCase
```

```
Parameter MADRouter1155.setSigner(address,address)._token (contracts/
   ↪ MADRouter1155.sol#712) is not in mixedCase
Parameter MADRouter1155.setSigner(address,address)._signer (contracts/
   ↪ MADRouter1155.sol#712) is not in mixedCase
Parameter MADRouter1155.setFees(uint256,uint256)._feeMint (contracts/
   ↪ MADRouter1155.sol#725) is not in mixedCase
Parameter MADRouter1155.setFees(uint256,uint256)._feeBurn (contracts/
   ↪ MADRouter1155.sol#725) is not in mixedCase
Variable MADRouter1155.MADFactory1155 (contracts/MADRouter1155.sol#32)
   ↪ is not in mixedCase
Parameter MADRouter721.setBase(address,string)._token (contracts/
   ↪ MADRouter721.sol#120) is not in mixedCase
Parameter MADRouter721.setBase(address,string)._baseURI (contracts/
   ↪ MADRouter721.sol#120) is not in mixedCase
Parameter MADRouter721.setBaseLock(address)._token (contracts/
   ↪ MADRouter721.sol#149) is not in mixedCase
Parameter MADRouter721.setMintState(address,bool,uint8)._token (
   ↪ contracts/MADRouter721.sol#180) is not in mixedCase
Parameter MADRouter721.setMintState(address,bool,uint8)._state (
   ↪ contracts/MADRouter721.sol#181) is not in mixedCase
Parameter MADRouter721.setMintState(address,bool,uint8)._stateType (
   ↪ contracts/MADRouter721.sol#182) is not in mixedCase
Parameter MADRouter721.whitelistSettings(address,uint256,uint256,bytes32
   ↪ )._token (contracts/MADRouter721.sol#213) is not in mixedCase
Parameter MADRouter721.whitelistSettings(address,uint256,uint256,bytes32
   ↪ )._price (contracts/MADRouter721.sol#214) is not in mixedCase
Parameter MADRouter721.whitelistSettings(address,uint256,uint256,bytes32
   ↪ )._supply (contracts/MADRouter721.sol#215) is not in mixedCase
Parameter MADRouter721.whitelistSettings(address,uint256,uint256,bytes32
   ↪ )._root (contracts/MADRouter721.sol#216) is not in mixedCase
Parameter MADRouter721.freeSettings(address,uint256,uint256,bytes32).
   ↪ _token (contracts/MADRouter721.sol#236) is not in mixedCase
Parameter MADRouter721.freeSettings(address,uint256,uint256,bytes32).
   ↪ _freeAmount (contracts/MADRouter721.sol#237) is not in mixedCase
```

```
Parameter MADRouter721.freeSettings(address,uint256,uint256,bytes32).
   ↪ _maxFree (contracts/MADRouter721.sol#238) is not in mixedCase
Parameter MADRouter721.freeSettings(address,uint256,uint256,bytes32).
   ↪ _claimRoot (contracts/MADRouter721.sol#239) is not in mixedCase
Parameter MADRouter721.minimalSafeMint(address,address)._token (
   ↪ contracts/MADRouter721.sol#259) is not in mixedCase
Parameter MADRouter721.minimalSafeMint(address,address)._to (contracts/
   ↪ MADRouter721.sol#259) is not in mixedCase
Parameter MADRouter721.basicMintTo(address,address,uint256)._token (
   ↪ contracts/MADRouter721.sol#280) is not in mixedCase
Parameter MADRouter721.basicMintTo(address,address,uint256)._to (
   ↪ contracts/MADRouter721.sol#281) is not in mixedCase
Parameter MADRouter721.basicMintTo(address,address,uint256)._amount (
   ↪ contracts/MADRouter721.sol#282) is not in mixedCase
Parameter MADRouter721.creatorMint(address,uint256)._token (contracts/
   ↪ MADRouter721.sol#298) is not in mixedCase
Parameter MADRouter721.creatorMint(address,uint256)._amount (contracts/
   ↪ MADRouter721.sol#298) is not in mixedCase
Parameter MADRouter721.gift(address,address[])._token (contracts/
   ↪ MADRouter721.sol#318) is not in mixedCase
Parameter MADRouter721.gift(address,address[])._addresses (contracts/
   ↪ MADRouter721.sol#319) is not in mixedCase
Parameter MADRouter721.burn(address,uint256[])._token (contracts/
   ↪ MADRouter721.sol#335) is not in mixedCase
Parameter MADRouter721.burn(address,uint256[])._ids (contracts/
   ↪ MADRouter721.sol#335) is not in mixedCase
Parameter MADRouter721.withdraw(address,ERC20)._token (contracts/
   ↪ MADRouter721.sol#374) is not in mixedCase
Parameter MADRouter721.withdraw(address,ERC20)._erc20 (contracts/
   ↪ MADRouter721.sol#374) is not in mixedCase
Parameter MADRouter721.setRecipient(address)._recipient (contracts/
   ↪ MADRouter721.sol#577) is not in mixedCase
Parameter MADRouter721.setSigner(address,address)._token (contracts/
   ↪ MADRouter721.sol#609) is not in mixedCase
```

```
Parameter MADRouter721.setSigner(address,address)._signer (contracts/
    ↪ MADRouter721.sol#609) is not in mixedCase
Parameter MADRouter721.setFees(uint256,uint256)._feeMint (contracts/
    ↪ MADRouter721.sol#622) is not in mixedCase
Parameter MADRouter721.setFees(uint256,uint256)._feeBurn (contracts/
    ↪ MADRouter721.sol#622) is not in mixedCase
Variable MADRouter721.MADFactory721 (contracts/MADRouter721.sol#32) is
    ↪ not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation
    ↪ #conformance-to-solidity-naming-conventions

Variable IUniswapV3SwapCallback.uniswapV3SwapCallback(int256,int256,
    ↪ bytes).amount0Delta (node_modules/@uniswap/v3-core/contracts/
    ↪ interfaces/callback/IUniswapV3SwapCallback.sol#17) is too similar
    ↪  to IUniswapV3SwapCallback.uniswapV3SwapCallback(int256,int256,
    ↪ bytes).amount1Delta (node_modules/@uniswap/v3-core/contracts/
    ↪ interfaces/callback/IUniswapV3SwapCallback.sol#18)
Variable MADFactory1155.splitterCheck(string,address,address,uint256,
    ↪ uint256)._payees_scope_0 (contracts/MADFactory1155.sol#184-187)
    ↪ is too similar to MADFactory1155.splitterCheck(string,address,
    ↪ address,uint256,uint256)._payees_scope_3 (contracts/
    ↪ MADFactory1155.sol#223-226)
Variable MADFactory1155.splitterCheck(string,address,address,uint256,
    ↪ uint256)._payees_scope_0 (contracts/MADFactory1155.sol#184-187)
    ↪ is too similar to MADFactory1155.splitterCheck(string,address,
    ↪ address,uint256,uint256)._payees_scope_6 (contracts/
    ↪ MADFactory1155.sol#264-267)
Variable MADFactory1155.splitterCheck(string,address,address,uint256,
    ↪ uint256)._payees_scope_3 (contracts/MADFactory1155.sol#223-226)
    ↪ is too similar to MADFactory1155.splitterCheck(string,address,
    ↪ address,uint256,uint256)._payees_scope_6 (contracts/
    ↪ MADFactory1155.sol#264-267)
Variable MADFactory1155.splitterCheck(string,address,address,uint256,
    ↪ uint256)._shares_scope_1 (contracts/MADFactory1155.sol#189) is
    ↪ too similar to MADFactory1155.splitterCheck(string,address,
    ↪ address,uint256,uint256)._shares_scope_4 (contracts/
    ↪ MADFactory1155.sol#228)
Variable MADFactory1155.splitterCheck(string,address,address,uint256,
    ↪ uint256)._shares_scope_1 (contracts/MADFactory1155.sol#189) is
    ↪ too similar to MADFactory1155.splitterCheck(string,address,
    ↪ address,uint256,uint256)._shares_scope_7 (contracts/
    ↪ MADFactory1155.sol#275)
Variable MADFactory1155.splitterCheck(string,address,address,uint256,
    ↪ uint256)._shares_scope_4 (contracts/MADFactory1155.sol#228) is
    ↪ too similar to MADFactory1155.splitterCheck(string,address,
    ↪ address,uint256,uint256)._shares_scope_7 (contracts/
    ↪ MADFactory1155.sol#275)
Variable MADFactory1155.splitterCheck(string,address,address,uint256,
    ↪ uint256)._splitter_scope_2 (contracts/MADFactory1155.sol#191-195)
    ↪  is too similar to MADFactory1155.splitterCheck(string,address,
    ↪ address,uint256,uint256)._splitter_scope_5 (contracts/
    ↪ MADFactory1155.sol#230-234)
Variable MADFactory1155.splitterCheck(string,address,address,uint256,
    ↪ uint256)._splitter_scope_2 (contracts/MADFactory1155.sol#191-195)
    ↪  is too similar to MADFactory1155.splitterCheck(string,address,
    ↪ address,uint256,uint256)._splitter_scope_8 (contracts/
    ↪ MADFactory1155.sol#277-281)
Variable MADFactory1155.splitterCheck(string,address,address,uint256,
    ↪ uint256)._splitter_scope_5 (contracts/MADFactory1155.sol#230-234)
    ↪  is too similar to MADFactory1155.splitterCheck(string,address,
    ↪ address,uint256,uint256)._splitter_scope_8 (contracts/
    ↪ MADFactory1155.sol#277-281)
Variable MADFactory1155.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).colId_scope_2 (contracts/
    ↪ MADFactory1155.sol#397) is too similar to MADFactory1155.
    ↪ createCollection(uint8,string,string,string,uint256,uint256,
    ↪ string,address,uint256).colId_scope_5 (contracts/MADFactory1155.
    ↪ sol#431)
```

```
Variable MADFactory1155.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).colId_scope_5 (contracts/
    ↪ MADFactory1155.sol#431) is too similar to MADFactory1155.
    ↪ createCollection(uint8,string,string,string,uint256,uint256,
    ↪ string,address,uint256).colId_scope_8 (contracts/MADFactory1155.
    ↪ sol#464)
Variable MADFactory1155.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).colId_scope_2 (contracts/
    ↪ MADFactory1155.sol#397) is too similar to MADFactory1155.
    ↪ createCollection(uint8,string,string,string,uint256,uint256,
    ↪ string,address,uint256).colId_scope_8 (contracts/MADFactory1155.
    ↪ sol#464)
Variable MADFactory1155.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).deployed_scope_1 (
    ↪ contracts/MADFactory1155.sol#385) is too similar to
    ↪ MADFactory1155.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).deployed_scope_4 (
    ↪ contracts/MADFactory1155.sol#419)
Variable MADFactory1155.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).deployed_scope_4 (
    ↪ contracts/MADFactory1155.sol#419) is too similar to
    ↪ MADFactory1155.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).deployed_scope_7 (
    ↪ contracts/MADFactory1155.sol#453)
Variable MADFactory1155.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).deployed_scope_1 (
    ↪ contracts/MADFactory1155.sol#385) is too similar to
    ↪ MADFactory1155.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).deployed_scope_7 (
    ↪ contracts/MADFactory1155.sol#453)
Variable MADFactory1155.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).tokenSalt_scope_0 (
    ↪ contracts/MADFactory1155.sol#385) is too similar to
    ↪ MADFactory1155.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).tokenSalt_scope_3 (
    ↪ contracts/MADFactory1155.sol#419)
Variable MADFactory1155.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).tokenSalt_scope_3 (
    ↪ contracts/MADFactory1155.sol#419) is too similar to
    ↪ MADFactory1155.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).tokenSalt_scope_6 (
    ↪ contracts/MADFactory1155.sol#453)
Variable MADFactory1155.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).tokenSalt_scope_0 (
    ↪ contracts/MADFactory1155.sol#385) is too similar to
    ↪ MADFactory1155.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).tokenSalt_scope_6 (
    ↪ contracts/MADFactory1155.sol#453)
Variable MADFactory721.splitterCheck(string,address,address,uint256,
    ↪ uint256)._payees_scope_0 (contracts/MADFactory721.sol#188-191) is
    ↪  too similar to MADFactory721.splitterCheck(string,address,
    ↪ address,uint256,uint256)._payees_scope_3 (contracts/MADFactory721
    ↪ .sol#227-230)
Variable MADFactory721.splitterCheck(string,address,address,uint256,
    ↪ uint256)._payees_scope_0 (contracts/MADFactory721.sol#188-191) is
    ↪  too similar to MADFactory721.splitterCheck(string,address,
    ↪ address,uint256,uint256)._payees_scope_6 (contracts/MADFactory721
    ↪ .sol#268-271)
Variable MADFactory721.splitterCheck(string,address,address,uint256,
    ↪ uint256)._payees_scope_3 (contracts/MADFactory721.sol#227-230) is
    ↪  too similar to MADFactory721.splitterCheck(string,address,
    ↪ address,uint256,uint256)._payees_scope_6 (contracts/MADFactory721
    ↪ .sol#268-271)
Variable MADFactory721.splitterCheck(string,address,address,uint256,
    ↪ uint256)._shares_scope_1 (contracts/MADFactory721.sol#193) is too
    ↪  similar to MADFactory721.splitterCheck(string,address,address,
    ↪ uint256,uint256)._shares_scope_4 (contracts/MADFactory721.sol
    ↪ #232)
```

```
Variable MADFactory721.splitterCheck(string,address,address,uint256,
   ↪ uint256)._shares_scope_1 (contracts/MADFactory721.sol#193) is too
   ↪  similar to MADFactory721.splitterCheck(string,address,address,
   ↪ uint256,uint256)._shares_scope_7 (contracts/MADFactory721.sol
   ↪ #279)
Variable MADFactory721.splitterCheck(string,address,address,uint256,
   ↪ uint256)._shares_scope_4 (contracts/MADFactory721.sol#232) is too
   ↪  similar to MADFactory721.splitterCheck(string,address,address,
   ↪ uint256,uint256)._shares_scope_7 (contracts/MADFactory721.sol
   ↪ #279)
Variable MADFactory721.splitterCheck(string,address,address,uint256,
   ↪ uint256)._splitter_scope_2 (contracts/MADFactory721.sol#195-199)
   ↪ is too similar to MADFactory721.splitterCheck(string,address,
   ↪ address,uint256,uint256)._splitter_scope_5 (contracts/
   ↪ MADFactory721.sol#234-238)
Variable MADFactory721.splitterCheck(string,address,address,uint256,
   ↪ uint256)._splitter_scope_2 (contracts/MADFactory721.sol#195-199)
   ↪ is too similar to MADFactory721.splitterCheck(string,address,
   ↪ address,uint256,uint256)._splitter_scope_8 (contracts/
   ↪ MADFactory721.sol#281-285)
Variable MADFactory721.splitterCheck(string,address,address,uint256,
   ↪ uint256)._splitter_scope_5 (contracts/MADFactory721.sol#234-238)
   ↪ is too similar to MADFactory721.splitterCheck(string,address,
   ↪ address,uint256,uint256)._splitter_scope_8 (contracts/
   ↪ MADFactory721.sol#281-285)
Variable MADFactory721.createCollection(uint8,string,string,string,
   ↪ uint256,uint256,string,address,uint256).colId_scope_2 (contracts/
   ↪ MADFactory721.sol#405) is too similar to MADFactory721.
   ↪ createCollection(uint8,string,string,string,uint256,uint256,
   ↪ string,address,uint256).colId_scope_5 (contracts/MADFactory721.
   ↪ sol#441)
Variable MADFactory721.createCollection(uint8,string,string,string,
   ↪ uint256,uint256,string,address,uint256).colId_scope_5 (contracts/
   ↪ MADFactory721.sol#441) is too similar to MADFactory721.
```

```
    ↪ createCollection(uint8,string,string,string,uint256,uint256,
    ↪ string,address,uint256).colId_scope_8 (contracts/MADFactory721.
    ↪ sol#476)
Variable MADFactory721.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).colId_scope_2 (contracts/
    ↪ MADFactory721.sol#405) is too similar to MADFactory721.
    ↪ createCollection(uint8,string,string,string,uint256,uint256,
    ↪ string,address,uint256).colId_scope_8 (contracts/MADFactory721.
    ↪ sol#476)
Variable MADFactory721.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).deployed_scope_1 (
    ↪ contracts/MADFactory721.sol#391) is too similar to MADFactory721.
    ↪ createCollection(uint8,string,string,string,uint256,uint256,
    ↪ string,address,uint256).deployed_scope_4 (contracts/MADFactory721
    ↪ .sol#427)
Variable MADFactory721.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).deployed_scope_4 (
    ↪ contracts/MADFactory721.sol#427) is too similar to MADFactory721.
    ↪ createCollection(uint8,string,string,string,uint256,uint256,
    ↪ string,address,uint256).deployed_scope_7 (contracts/MADFactory721
    ↪ .sol#463)
Variable MADFactory721.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).deployed_scope_1 (
    ↪ contracts/MADFactory721.sol#391) is too similar to MADFactory721.
    ↪ createCollection(uint8,string,string,string,uint256,uint256,
    ↪ string,address,uint256).deployed_scope_7 (contracts/MADFactory721
    ↪ .sol#463)
Variable MADFactory721.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).tokenSalt_scope_0 (
    ↪ contracts/MADFactory721.sol#391) is too similar to MADFactory721.
    ↪ createCollection(uint8,string,string,string,uint256,uint256,
    ↪ string,address,uint256).tokenSalt_scope_3 (contracts/
    ↪ MADFactory721.sol#427)
```

```
Variable MADFactory721.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).tokenSalt_scope_3 (
    ↪ contracts/MADFactory721.sol#427) is too similar to MADFactory721.
    ↪ createCollection(uint8,string,string,string,uint256,uint256,
    ↪ string,address,uint256).tokenSalt_scope_6 (contracts/
    ↪ MADFactory721.sol#463)
Variable MADFactory721.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).tokenSalt_scope_0 (
    ↪ contracts/MADFactory721.sol#391) is too similar to MADFactory721.
    ↪ createCollection(uint8,string,string,string,uint256,uint256,
    ↪ string,address,uint256).tokenSalt_scope_6 (contracts/
    ↪ MADFactory721.sol#463)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation
    ↪ #variable-names-are-too-similar

MADFactory1155._limiter(uint8,address) (contracts/MADFactory1155.sol
    ↪ #761-775) uses literals with too many digits:
        - mstore(uint256,uint256)(0x00,0
            ↪ x4ca8886700000000000000000000000000000000000000000000000000000000
            ↪ ) (contracts/MADFactory1155.sol#771)
MADFactory721._limiter(uint8,address) (contracts/MADFactory721.sol
    ↪ #775-789) uses literals with too many digits:
        - mstore(uint256,uint256)(0x00,0
            ↪ x4ca8886700000000000000000000000000000000000000000000000000000000
            ↪ ) (contracts/MADFactory721.sol#785)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation
    ↪ #too-many-digits

MADFactory1155.market (contracts/MADFactory1155.sol#79) should be
    ↪ constant
MADFactory1155.signer (contracts/MADFactory1155.sol#82) should be
    ↪ constant
MADFactory721.market (contracts/MADFactory721.sol#79) should be constant
MADFactory721.signer (contracts/MADFactory721.sol#82) should be constant
```

```
MADMarketplace1155.feeVal2 (contracts/MADMarketplace1155.sol#46) should
    ↪ be constant
MADMarketplace1155.feeVal3 (contracts/MADMarketplace1155.sol#47) should
    ↪ be constant
MADMarketplace1155.maxOrderDuration (contracts/MADMarketplace1155.sol
    ↪ #73) should be constant
MADMarketplace1155.minAuctionIncrement (contracts/MADMarketplace1155.sol
    ↪ #71) should be constant
MADMarketplace1155.minBidValue (contracts/MADMarketplace1155.sol#72)
    ↪ should be constant
MADMarketplace1155.minOrderDuration (contracts/MADMarketplace1155.sol
    ↪ #70) should be constant
MADMarketplace1155.recipient (contracts/MADMarketplace1155.sol#75)
    ↪ should be constant
MADMarketplace721.feeVal2 (contracts/MADMarketplace721.sol#46) should be
    ↪  constant
MADMarketplace721.feeVal3 (contracts/MADMarketplace721.sol#47) should be
    ↪  constant
MADMarketplace721.maxOrderDuration (contracts/MADMarketplace721.sol#71)
    ↪ should be constant
MADMarketplace721.minAuctionIncrement (contracts/MADMarketplace721.sol
    ↪ #72) should be constant
MADMarketplace721.minBidValue (contracts/MADMarketplace721.sol#73)
    ↪ should be constant
MADMarketplace721.minOrderDuration (contracts/MADMarketplace721.sol#70)
    ↪ should be constant
MADMarketplace721.recipient (contracts/MADMarketplace721.sol#75) should
    ↪ be constant
MADRouter1155.feeBurn (contracts/MADRouter1155.sol#47) should be
    ↪ constant
MADRouter1155.feeMint (contracts/MADRouter1155.sol#44) should be
    ↪ constant
MADRouter1155.recipient (contracts/MADRouter1155.sol#50) should be
    ↪ constant
```

```
MADRouter721.feeBurn (contracts/MADRouter721.sol#47) should be constant
MADRouter721.feeMint (contracts/MADRouter721.sol#44) should be constant
MADRouter721.recipient (contracts/MADRouter721.sol#56) should be
   ↪ constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation
   ↪ #state-variables-that-could-be-declared-constant

bid(bytes32) should be declared external:
       - MADMarketplace721.bid(bytes32) (contracts/MADMarketplace721.sol
          ↪ #169-258)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation
   ↪ #public-function-that-could-be-declared-external
. analyzed (75 contracts with 78 detectors), 495 result(s) found
```

Conclusion:

Most of the vulnerabilities found by the analysis have already been addressed by the smart contract code review.

7 Conclusion

In this audit, we examined the design and implementation of MADNFT contract and discovered several issues of varying severity. Jacob Clay team addressed all the issues raised in the initial report and implemented the necessary fixes.

The present code base is well-structured and ready for the mainnet.

BLOCKHAT

Security