**NFT Souq Smart Contracts Final Report**

[Scope of Audit](#_27qphk2hws9d) 3

[Check Vulnerabilities](#_fk11hakqdckb) 3

[Techniques and Methods](#_2et92p0) **4**

[Issue Categories](#_2p6a8gpqk15r) 5

[Number of security issues per severity.](#_2idbboidjlj7) 6

[**Introduction**](#_aqhg5kqr9l1u) **6**

[**Issues Found – Code Review / Manual Testing**](#_324c3n3m672b) **7**

[High Severity Issues](#_fpa7vs16wijg) 7

[A.1 Missing caller check before media contract update](#_yzmmo7wjipia) 7

[A.2 Anyone can update the ask for any tokenId](#_bwt5nc5e7eew) 8

[A.3 Missing authentication on admin and commission updates](#_767k9a3019d6) 8

[Medium Severity Issues](#_61duwpxcikki) 10

[A.4 Media contract retains allowance to move the tokens even after sale](#_vpqkbh6rakm5) 10

[A.5 Unnecessary payable methods](#_t6if3abnf6w4) 10

[Low Severity Issues](#_edbfqbhkxq1) 12

[A.6 getTotalNumberOfNFT returns number of all tokens minted](#_xg8p62iavmvd) 12

[A.7 Redundant code](#_j9db8neahdjh) 12

[Informational Issues](#_qaao2xn7xl8t) 14

[A.8 Calling transfer() function on behalf of onlyMediaCaller.](#_bnfy0uvce8ul) 14

[A.9 Unnecessary use of SafeMath](#_w0vntpono8zu) 14

[A.10 End Auction is called even for non auction sale](#_nhb4vqaltlj9) 15

[A.11 Unused Mappings](#_unuvo0w5fbua) 15

[A.12 State variable never updated should be constant](#_vkcliqxbe0rm) 15

[A.13 Repeated require checks](#_ia6jm4dnennx) 16

[A.14 Unnecessary storage read](#_6civuqcqtm6l) 16

[A.15 Redundant mathematical operation](#_1enr6l318a00) 17

[A.16 Dead Code](#_x7u68gwznmvz) 17

[A.17 Remove commented code](#_8y374hc7ev5e) 18

[A.18 Incorrect spelling](#_cq0ypdvj3ux2) 18

[A.19 Methods should be external is not being used internally](#_p2a0134xlr47) 19

[A.20 Missing Events for Significant Transactions](#_ja3savbl0xjh) 19

[B.21 Ambiguous code comments](#_t4yeqzvapwto) 20

[A.22 State Variable Default Visibility](#_6uq77nqssij) 20

[A.23 Use double quotes for string literals](#_shp9zxvylmce) 21

[**Functional Testing**](#_qca7jwpkvv6y) **22**

[ERC1155Factory.sol](#_ii1xm36lh7c2) 22

[ERC721Factory.sol](#_gu4x5n6mxc0z) 22

[Media.sol](#_wyv5yicm3qol) 22

[**Kovan Test Contracts**](#_t1tl3cno0wb3) **22**

[**Automated Tests  
  
Slither:**](#_mvahembg7xmp) **28**

[**Results**](#_nnxarl1y1et8) **33**

[**Closing Summary**](#_ledadd1qug2x) **34**

[**Disclaimer**](#_mvxk3jly36z5) **34**

## 

## **Scope of Audit**

The scope of this audit was to analyze and document the **NFT Souq** Token smart contract codebase for quality, security, and correctness.

## 

## **Check Vulnerabilities**

* Re-entrancy
* Timestamp Dependence
* Gas Limit and Loops
* DoS with Block Gas Limit
* Transaction-Ordering Dependence
* Use of tx.origin
* Exception disorder
* Gasless send
* Balance equality
* Byte array
* Transfer forwards all gas
* ERC20 API violation
* Malicious libraries
* Compiler version not fixed
* Redundant fallback function
* Send instead of transfer
* Style guide violation
* Unchecked external call
* Unchecked math
* Unsafe type inference
* Implicit visibility level

# **Techniques and Methods**

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

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

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

**Structural Analysis**

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

**Static Analysis**

A static Analysis of Smart Contracts was done to identify contract vulnerabilities. In this step, a series of automated tools are used to test the security of smart contracts.

**Code Review / Manual Analysis**

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

**Gas Consumption**

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

**Tools and Platforms used for Audit**

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

## **Issue Categories**

Every issue in this report has been assigned to a severity level. There are four levels of severity, and each of them has been explained below.

**High Severity Issues**

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

**Medium Severity Issues**

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

**Low Severity Issues**

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

**Informational** **Issues**

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

## 

## **Number of security issues per severity.**

# 

| **TYPE** | **HIGH** | **MEDIUM** | **LOW** | **INFORMATIONAL** |
| --- | --- | --- | --- | --- |
| **Open** | **0** | **0** | **0** | **0** |
| **Acknowledged** | **0** | **01** | **0** | **02** |
| **Closed** | **03** | **01** | **02** | **14** |

## **Introduction**

During the period of **December 06, 2021 to December 15, 2021**. The QuillAudits Team performed a security audit for **NFT Souq** smart contracts.

The code for the audit was taken from following the official link**:**

**https://github.com/Marhaba-DeFi/nft\_souq\_contracts**

| **Version Number** | **Date** | **Commit ID** |
| --- | --- | --- |
| 01 | Dec 06 | e36c6cb9c7bce8954b68a94c500d34634a124cb5 |
| 02 | Dec 14 | 792c918d998e18cdc2539c764e0982ca717aa241 |
| 03 | Dec 16 | f803b552c9b9b1e6b810bf48a5954d1afa657c20 |
| 04 | Dec 17 | 7aad8275137b32c5af9a83775889447bf0e31927 |
| 05 | Dec 29 | a3a969d3cd79d1c961acd01c9fb5ed10e53edbf6 |

# **Issues Found – Code Review / Manual Testing**

## **High Severity Issues**

### A.1 Missing caller check before media contract update

Contract: ERC721Factory

| **Line** | **Code/Function** |
| --- | --- |
| 15 | function configureMedia(address \_mediaContractAddress) |

Contract: ERC1155Factory

| **Line** | **Code/Function** |
| --- | --- |
| 26 | function configureMedia(address \_mediaContractAddress) |

Contract: Market

| **Line** | **Code/Function** |
| --- | --- |
| 85 | function configureMedia(address \_mediaContractAddress) |

**Description**

The **configureMedia** method can be called by anyone and has no modifier to check the authority of the caller

As a consequence anyone can create their on media contract and update media address to use any of these deployed contracts

**Recommendation**

Add modifier to check owner or admin before updating media contract address

**Status: Fixed** in version 02. Those aforementioned functions are only called by the owner.

### A.2 Anyone can update the ask for any tokenId

Contract: Media

| **Line** | **Code/Function** |
| --- | --- |
| 233 | function setAsk(uint256 \_tokenID, Iutils.Ask memory ask) public override {  require(  msg.sender == ask.\_sender,  "MEDIA: sender in ask tuple needs to be msg.sender"  );  IMarket(\_marketAddress).setAsk(\_tokenID, ask);  } |

**Description**

The require check within the method only compares the incoming sender value with the msg.sender and to pass the check the caller can simply pass the sender address in the ask object itself.

**Remediation**:

The require check should compare the msg.sender value with token owner address or some admin address.

**Status: Fixed** in version 02

### 

### A.3 Missing authentication on admin and commission updates

Contract: Media

| **Line** | **Code/Function** |
| --- | --- |
| 272 | function setAdminAddress(address \_adminAddress) |
| 281 | function setCommissionPercentage(uint8 \_newCommissionPercentage) |

**Description**

These functions have no check before updating the admin address or the commission percentage which can be easily updated by the attacker to divert all the transaction fee to their own address

**Recommendation**

Added modifier to check if the caller is an authorized user.

**Status: Fixed** in version 04

## 

## **Medium Severity Issues**

### A.4 Media contract retains allowance to move the tokens even after sale

Contract: ERC721Factory and ERC1155Factory

| **Line** | **Code/Function** |
| --- | --- |
| 63 and 56 respectively | function transferFrom(address \_sender, address \_recipient, uint256 tokenID) |

**Description**

All the token transfers use the transferFrom defined in ERC721Factory and ERC1155Factory which in the logic itself always allow the media contract to move the token.

This can mean even the final owner of the token can loose the token if media contract is compromised

**Recommendation**:

Only give approval to media contract when the bidding is ongoing

**Status: Acknowledged**

### A.5 Unnecessary payable methods

Contract: Media

| **Line** | **Code/Function** |
| --- | --- |
| 59 | function mintToken(MediaData memory data) external payable |
| 191 | function setBid(uint256 \_tokenID, IUtils Bid calldata bid) external payable |

**Description**

The methods are made payable but there is no logic associated with msg.value. These methods allow the contract to receive ether which will get locked in the contract since there is no withdraw function.

**Recommendation**:

Add a withdraw function if we want to receive either or simple remove payable from the method

**Status: Fixed** in version 02

## 

## **Low Severity Issues**

### A.6 getTotalNumberOfNFT returns number of all tokens minted

Contract: Media

| **Line** | **Code/Function** |
| --- | --- |
| 187 | function getTotalNumberOfNFT() external view returns (uint256) {  return \_tokenCounter;  } |

**Description**

The value of **\_tokenCounter** is incremented both when NFTs are minted and when FTs are minted, so method does not return the total number of NFTs as suggested the method name

**Recommendation**:

We can maintain two different counts for NFTs and FTs minted or update the method name

**Status: Fixed** in version 05

### A.7 Redundant code

Contract: Market

| **Line** | **Code/Function** |
| --- | --- |
| 24 | uint256 private \_adminPoints; |
| 405-408 | function addAdminCommission(uint256 \_amount) external override onlyMediaCaller returns (bool) {  \_adminPoints = \_adminPoints.add(\_amount);  return true;  } |

\_adminPoints variable was not used for any operation within the contract. We recommend removing this variable and the addAdminCommission() function that is used to update the variable.

**Status: Fixed** in version 02

## **Informational Issues**

### A.8 Calling transfer() function on behalf of onlyMediaCaller.

Contract: ERC1155, ERC721

**Description**

The transferFrom() in ERC1155Factory.sol is a user-restricted function that accepts only the onlyMediaCaller contract address. However, typical users can bypass this restriction by calling the safeTransferFrom() and setApprovalForAll() functions separately on behalf of the onlyMediaCaller. Because these 02 functions are set to public and have no access controls.

Similarly to the ERC721Factory.sol, the transferFrom() function allows only the onlyMediaCaller. However, other users can just call safeTransferFrom() and approve() on behalf of the onlyMediaCaller.

**Remediation**

We recommend removing the modifier onlyMediaCaller() or having further consideration on making other public functions listed above also restricted.

**Status: Acknowledged**

### A.9 Unnecessary use of SafeMath

**Description:**

We import and use SafeMath in contracts Market and Media whileSolidity 0.8 has an integrated SafeMath and new error handling for system errors

**Recommendation:**

Remove explicit inclusion of safemath

**Status: Fixed** in version 03

### A.10 End Auction is called even for non auction sale

Contract: Media

| **Line** | **Code/Function** |
| --- | --- |
| 249 | function endAuction(uint256 \_tokenID) |

**Description:**

The endAuction method can be called for any token Id which is not even put to auction

**Recommendation:**

Add a require check to see if the given token Id has been put to auction

**Status: Fixed** in version 02

### A.11 Unused Mappings

**Description:**

tokenBids, userRedeemPoints, tokenTransactionHistory, tokenBidders, \_newTokenBids and newTokenBidders are unused mappings in Market contract.

**Recommendation:**

Remove unused mappings

**Status: Fixed** in version 02

### A.12 State variable never updated should be constant

**Description:**

**minBidIncrementPercentage** is defined as a state variable and is never updated

**Recommendation:**

We recommend making it a constant or create a setter for the value

**Status: Fixed** in version 03. The setMinimumBidIncrementPercentage() was created to update the **minBidIncrementPercentage** variable.

### A.13 Repeated require checks

**Description:**

**\_handleAuction** being called through **setBid** performs a repeated allowance check which is already being checked in the setBid before **\_handleAuction** is called

**Recommendation:**

Remove the redundant check

**Status: Fixed** in version 02

### A.14 Unnecessary storage read

Contract: Market

| **Line** | **Code/Function** |
| --- | --- |
| 140 | Iutils.Bid storage existingBid = \_tokenBidders[\_tokenID][\_bidder]; |

Description

**existingBid** is used only conditionally but read every time.

**Recommendation**

Reading from storage is a very costly operation so it should only be done when required.

**Status: Acknowledged**

### A.15 Redundant mathematical operation

Contract: Market

| Line | Code |
| --- | --- |
| 391 | totalAmountTransferred = totalAmountTransferred.add(royaltyPoints.sub(totalAmountTransferred)) |

**Description**

We add **totalAmountTransferred** and subtract **totalAmountTransferred** from royaltyPoints

**Recommendation**

totalAmountTransferred = royaltyPoints;

**Status: Fixed** in version 02

### A.16 Dead Code

Contract: Market

| Line | Function |
| --- | --- |
| 405 | addAdminCommission |
| 301 | getAdminAddress |

**Description**

The methods are restricted by modifier **onlyMediaCaller** but media never tries to access these methods

**Recommendation**

Add call for **addAdminCommision** from Media contract and remove access restriction from **getAdminAddress**

**Status: Fixed** in version 02

### A.17 Remove commented code

**Description:**

There are pieces of commended code in the contracts

**Recommendation:**

Remove commented code

**Status: Fixed** in version 02

### A.18 Incorrect spelling

Contract: Media

**\_duration** is spelled as **\_duation** throughout the code, it should be **\_duration**

Contract: Market

**L25**: **storre** should be **stored**

**L88**: **Alredy** should be **Already**

**L252**: **potentally** should be **potentially**

**L364**: **royality** should be **royalty**

**L371**: **Collaboratoes** should be **Collaborators**

**L371**: **occuring** should be **occurring**

Entire code-base: **tokenColab** should be **tokenCollab**

**L385**: **transfering** should be **transferring**

**Status: Fixed** in version 04 and 02.

### A.19 Methods should be external is not being used internally

**Description:**

**setAsk** and **getToken** in Media contract are not being used internally but are made as public

**Recommendation:**

External contracts do not copy the parameters to memory unless required hence consume less gas than public methods so we should make methods external unless needed internally

**Status: Fixed** in version 03

### A.20 Missing Events for Significant Transactions

Contract: Media, Market

**Description:**

The critical settings are completely devoid of event definitions or emissions. This makes it very difficult for users or other interested parties to track important changes that take place in the system.

Thus, the missing event makes it difficult to track off-chain liquidity fee changes. An event should be emitted for significant transactions calling the following functions:

* setAdminAddress()
* setCommissionPercentage()
* setRoyaltyPoints()
* setCollaborators()
* configureMedia()

**Remediation:**

We recommend emitting an event to log the update of the above functions.

**Status: Fixed** in version 02

### B.21 Ambiguous code comments

Contract: Media

The comment within minToken() function “// verify sum of collaborators percentages needs to be less then or equals to 100” is inconsistent with the number in the require() function in line 79:

| require(  sumOfCollabRoyalty <= 10,  "Media: Sum of Collaborators Percentages can be maximum 10"  ); |
| --- |

In the comment it is stated that the sum of collaborators percentages needs to be less than or equal to 100. However, 10 was used to check for the sum or collaborators instead of 100. It is advised to make the comment more specific.

**Status: Fixed** in version 02

### A.22 State Variable Default Visibility

Contract: Media

**L27:** mapping(uint256 => address) nftToCreators;

**L30:** mapping(uint256 => MediaInfo) tokenIDToToken;

Contract: Market

**L60:** uint256 constant EXPO = 1e18;

**L62:** uint256 constant BASE = 100 \* EXPO;

**Description**

The Visibility of the aforementioned variable is not defined. Labeling the visibility explicitly makes it easier to catch incorrect assumptions about who can access the variable.

The default is internal for state variables, but it should be made explicit.

**Status: Fixed** in version 02

### A.23 Use double quotes for string literals

Contract: Market

**Description**

Single quote found in the entire code-base for string variables. Whilst, the double quotes are being utilized for other string literals.

**Remediation**

We recommend using double quotes for string literals in the entire codebase.

**Status: Fixed** in version 03

# 

# **Functional Testing**

Some of the test cases which were part of the functional testing are listed below:

## **ERC1155Factory.sol**

1. Set media contract address PASS

## **ERC721Factory.sol**

1. Set media contract address PASS

## **Media.sol**

1. Set Admin Address PASS
2. Set Admin Commision PASS

# **Kovan Test Contracts**

**ERC1155Factory:** 0x818470295B32a5e68a2eaD45DE8a35633031bC45

**ERC721Factory:** 0xAF0eE7ECaaeA555FD1A307783b0c6999f856F751

**Market**: 0xC9bEBB6354385d269a2Cea78aAD1fB975dfd6FF3

**Media:** 0x207d45f251566d0DD2AD618Fae8Ca7b74229ad33

**BSC Test Contracts**

**ERC1155Factory:** 0x9f207f91090eE09cd5a262ce592d8B6A44D645D3

**ERC721Factory:** 0xa455B435f57Cf2E2A3B6C0952E2D58d25584Bf7c

**Market**: 0x623F87B1270Da736D68a05F88d634625E14985b0

**Media:** 0x6d8595B7eAE21fcf461C192181b1242a6903b6e1

**Market**

| **Function name** | **Input** | **Output** | **TX** | **Status** | **Fix Status** |
| --- | --- | --- | --- | --- | --- |
| configureMedia | 0xDFE32E79A9afda8744C52980dF6461C38B9fde22 | true | 0xc4f073707f5e6feb586d25132d555d4d0fbdd978566d75f7af9614dd7b585b3e | FAILED (Anyone can call this function) | Passed |
| addAdminCommission | 1000 | execution reverted: Market: Unauthorized Access | N/A | Passed | N/A |
| setBid | "1","0x60b6D91cB698F41E1eD928f9631cEC6b8Ff8F6cC",[2, "1000", "0x50073629613AAe78b215F0f42Cd9D9Fe56210a87", "0x60b6D91cB698F41E1eD928f9631cEC6b8Ff8F6cC", "0x153b057d5d7262dC92099B59c975255ecE66784F", 1],"0x1dd64394E29c5988f04A8E074D0DBACd4D614729","0x1dd64394E29c5988f04A8E074D0DBACd4D614729" | execution reverted: Market: Unauthorized Access | N/A | Passed | N/A |
| setAsk | 1,["0x1dd64394E29c5988f04A8E074D0DBACd4D614729", 1000, 10000, 1000,"0x50073629613AAe78b215F0f42Cd9D9Fe56210a87", 1, 600, 1638948722, "0x153b057d5d7262dC92099B59c975255ecE66784F",500] | execution reverted: Market: Unauthorized Access | N/A | Passed | N/A |
| setCollaborators | 1, [["0x1dd64394E29c5988f04A8E074D0DBACd4D614729"],[90],true] | execution reverted: Market: Unauthorized Access | N/A | Passed | N/A |
| setRoyaltyPoints | 1, 90 | execution reverted: Market: Unauthorized Access | N/A | Passed | N/A |
| setCommissionPercentage | 90 | execution reverted: Market: Unauthorized Access | N/A | Passed | N/A |
| setAdminAddress | 0x1dd64394E29c5988f04A8E074D0DBACd4D614729 | execution reverted: Market: Unauthorized Access | N/A | Passed | N/A |
| removeBid | 0xDFE32E79A9afda8744C52980dF6461C38B9fde22 | execution reverted: Market: Unauthorized Access | N/A | Passed | N/A |
| cancelAuction | 1 | execution reverted: Market: Unauthorized Access | N/A | Passed | N/A |
| addAdminCommission | 90 | execution reverted: Market: Unauthorized Access | N/A | Passed | N/A |

**Media**

| **Function name** | **Input** | **Output** | **TX** | **Status** | **Fix Status** |
| --- | --- | --- | --- | --- | --- |
| setAdminAddress | 0x1dd64394E29c5988f04A8E074D0DBACd4D614729 | true | 0x483f40957c0b0bf0bc29f3f6c87c77a16fe709769cd2e9334d88c0cd40533618 | FAILED (everyone can call this function) | Passed |
| setCommissionPercentage | 10 | true | 0x3a7335972b6bb7c81d78311cdde06cf9c66fd986691a36a91006ee1ae06a304c | FAILED (everyone can call this function) | Passed |
| mintToken (mint ERC1155) | ["http://mytoken.com","mytoken",10000000,10, ["0x60b6D91cB698F41E1eD928f9631cEC6b8Ff8F6cC","0x153b057d5d7262dC92099B59c975255ecE66784F"],[3,3],1,1000000,1000000, "0x8cF95C8750FB8E83Cc45F44232da9Dd022037e05", 3600] | true | 0x7bcb767684fff4d4dac48f0fb0831116a08d59ffec2d6a54fc40f70c9273ded7 | Passed | N/A |
| mintToken (mint ERC721) | ["http://mytoken.copm","mytoken",1,10, ["0x60b6D91cB698F41E1eD928f9631cEC6b8Ff8F6cC","0x153b057d5d7262dC92099B59c975255ecE66784F"],[3,3],1,1000000,1000000, "0x8cF95C8750FB8E83Cc45F44232da9Dd022037e05", 120] | true | 0x37da80744719b4d1a42ff7f7ffba7a51e6241e8a1e3b7c01bc124be4258cb33e | Passed | N/A |
| cancelAuction | 1 | true | 0xf857df1a843f60d8b853ae94e1bf3d8848bf80f3457787256cf4aaa195d984c4 | Passed | N/A |
| getCommissionPercentage | N/A | true | N/A | Passed | N/A |
| setBid | "4",[1,"2000000", "0xD9972082c296e76f4BC10C8BD20998c61026c483", "0x8cF95C8750FB8E83Cc45F44232da9Dd022037e05","0x60b6D91cB698F41E1eD928f9631cEC6b8Ff8F6cC", 0] | true | 0x31e4bfba4d0d21e3146ae528953511502471fea3f5deac8183a5a5c4639a99b0 | Passed | N/A |

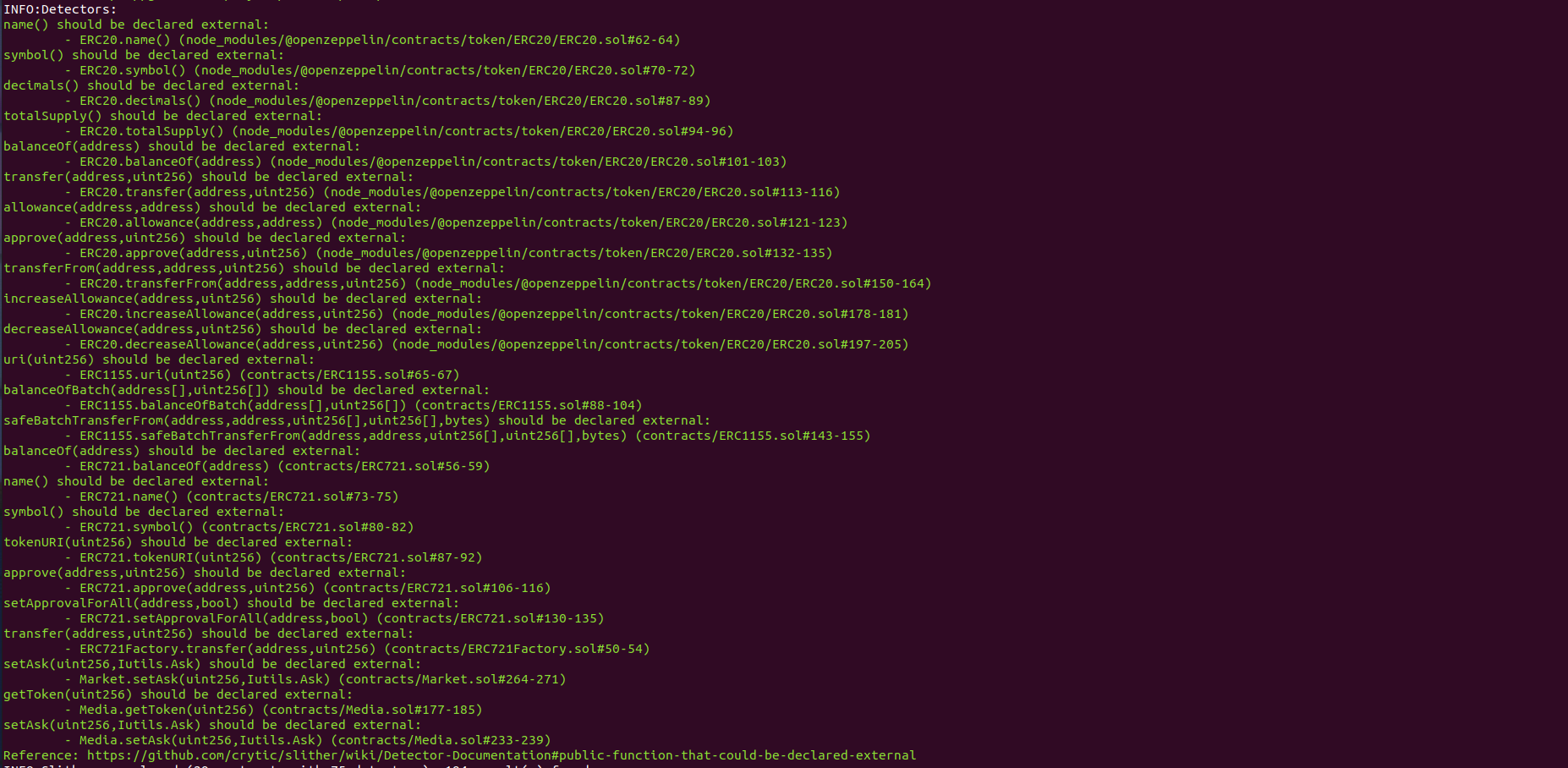
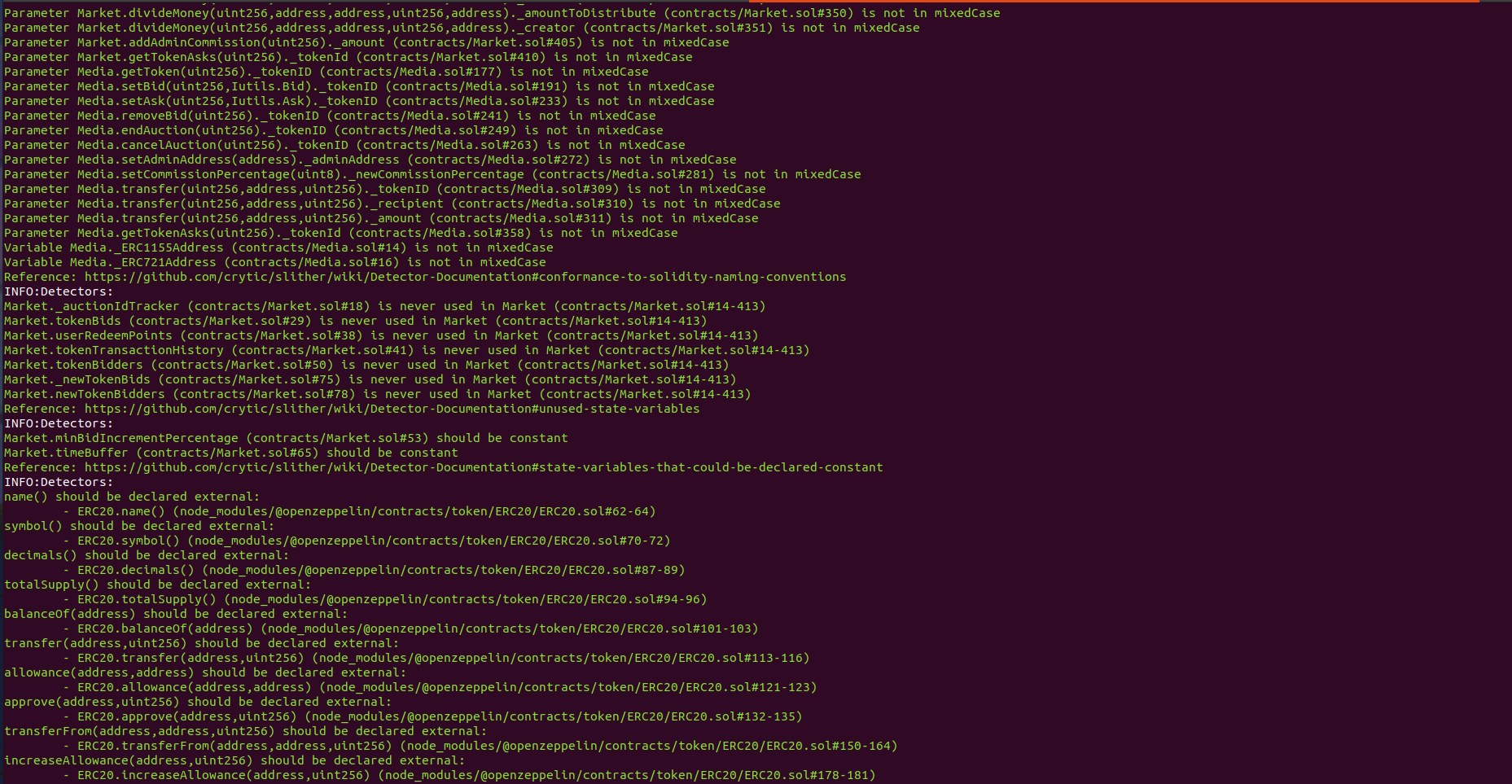
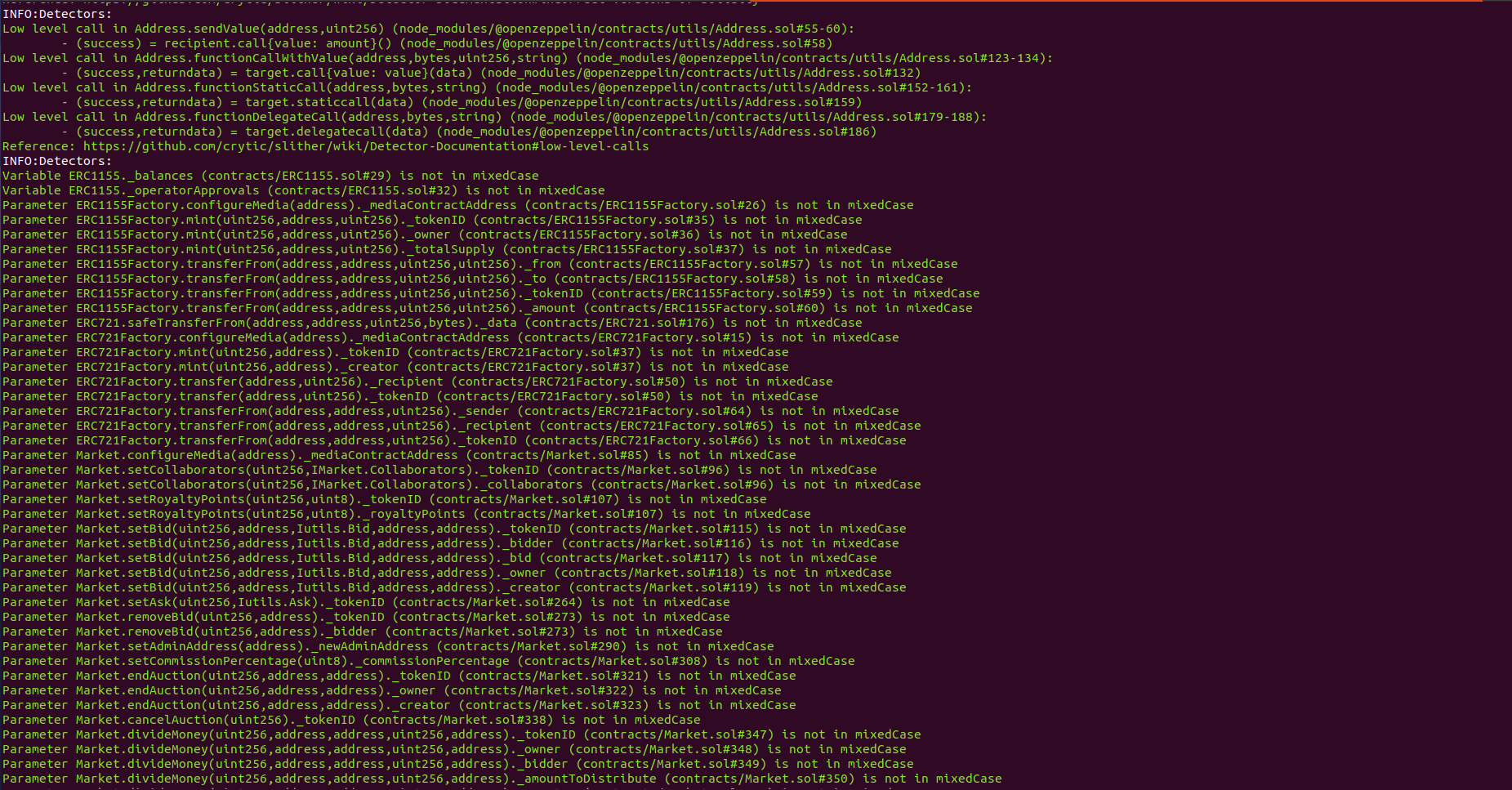
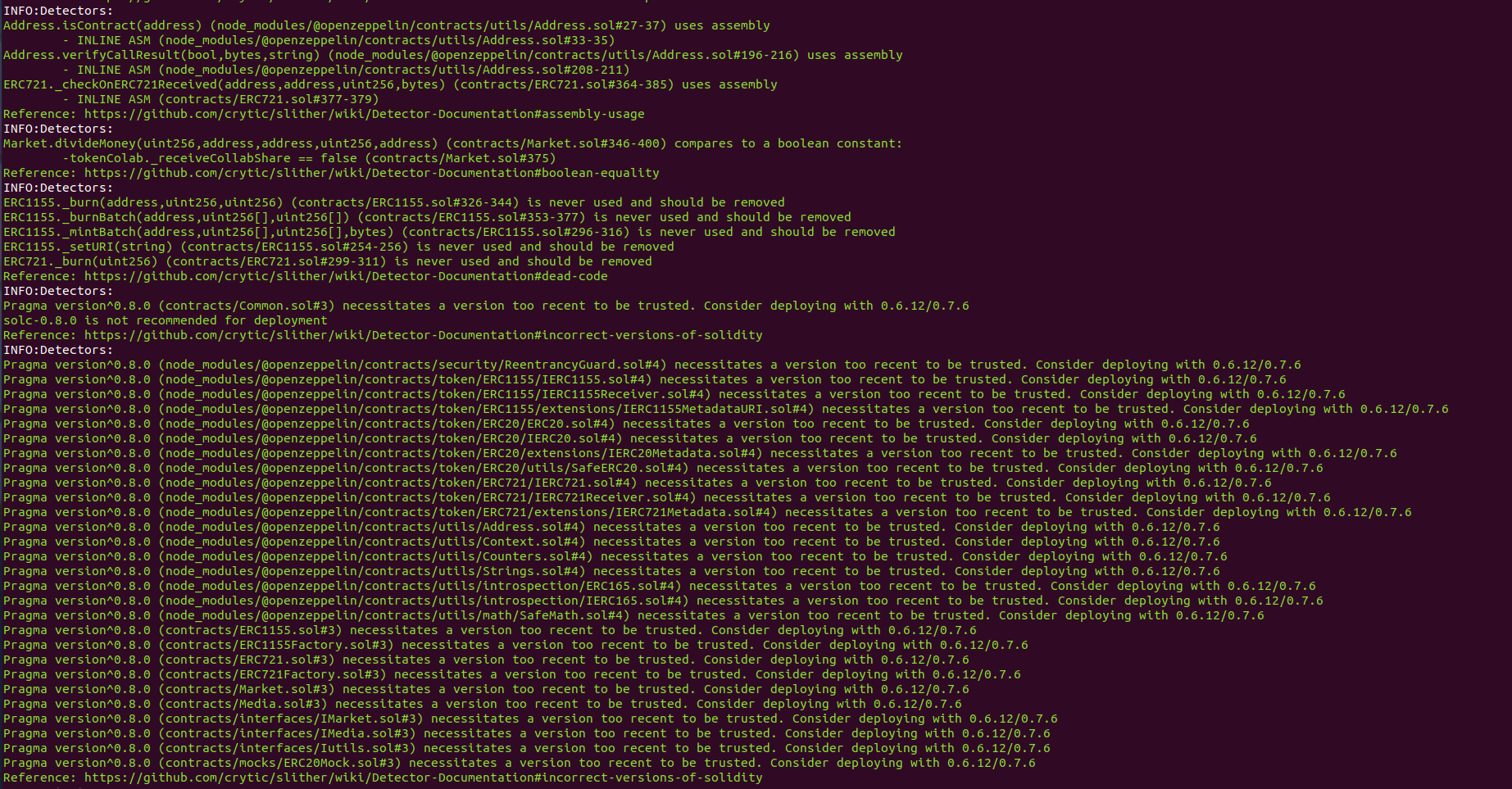
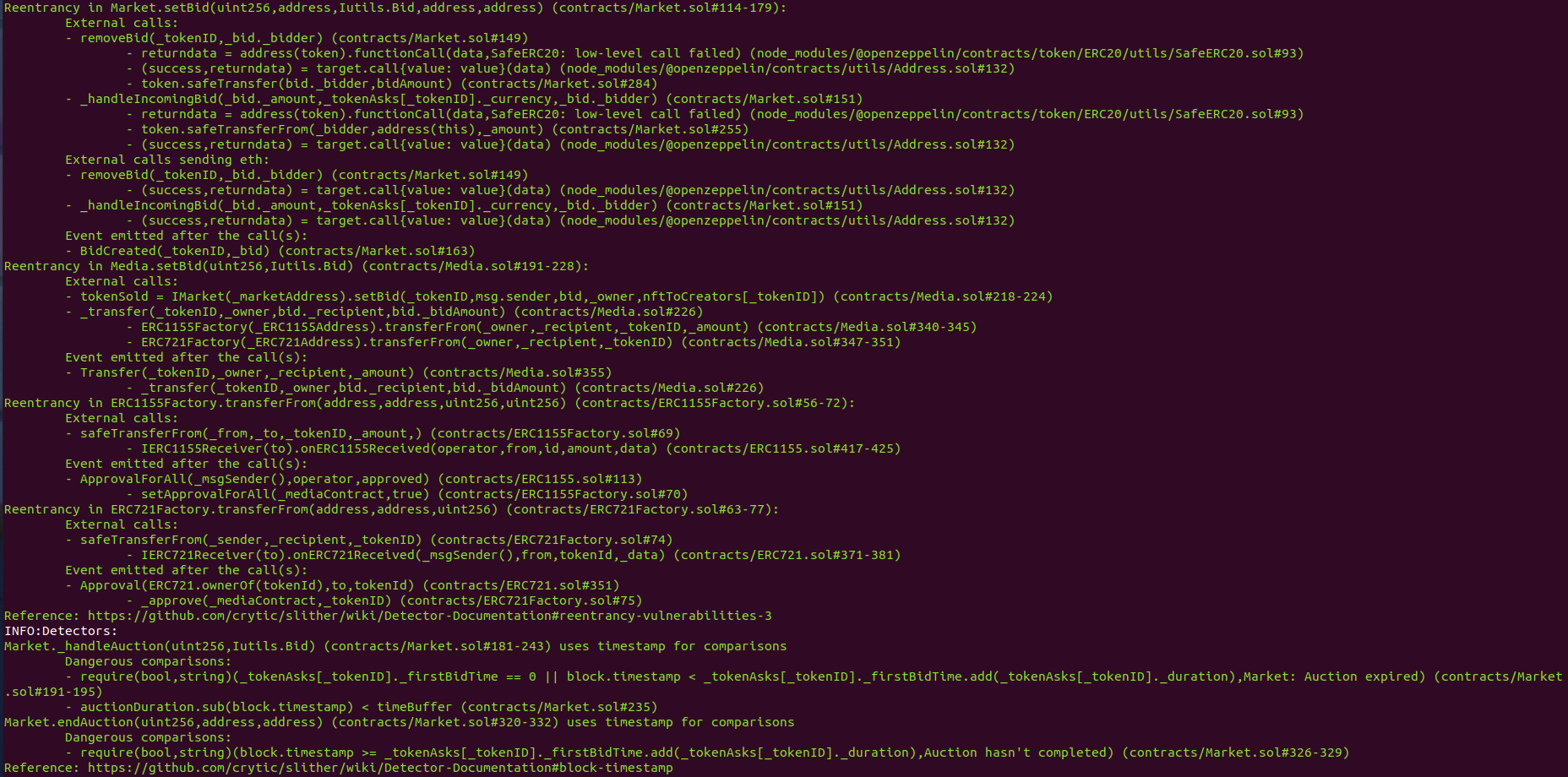
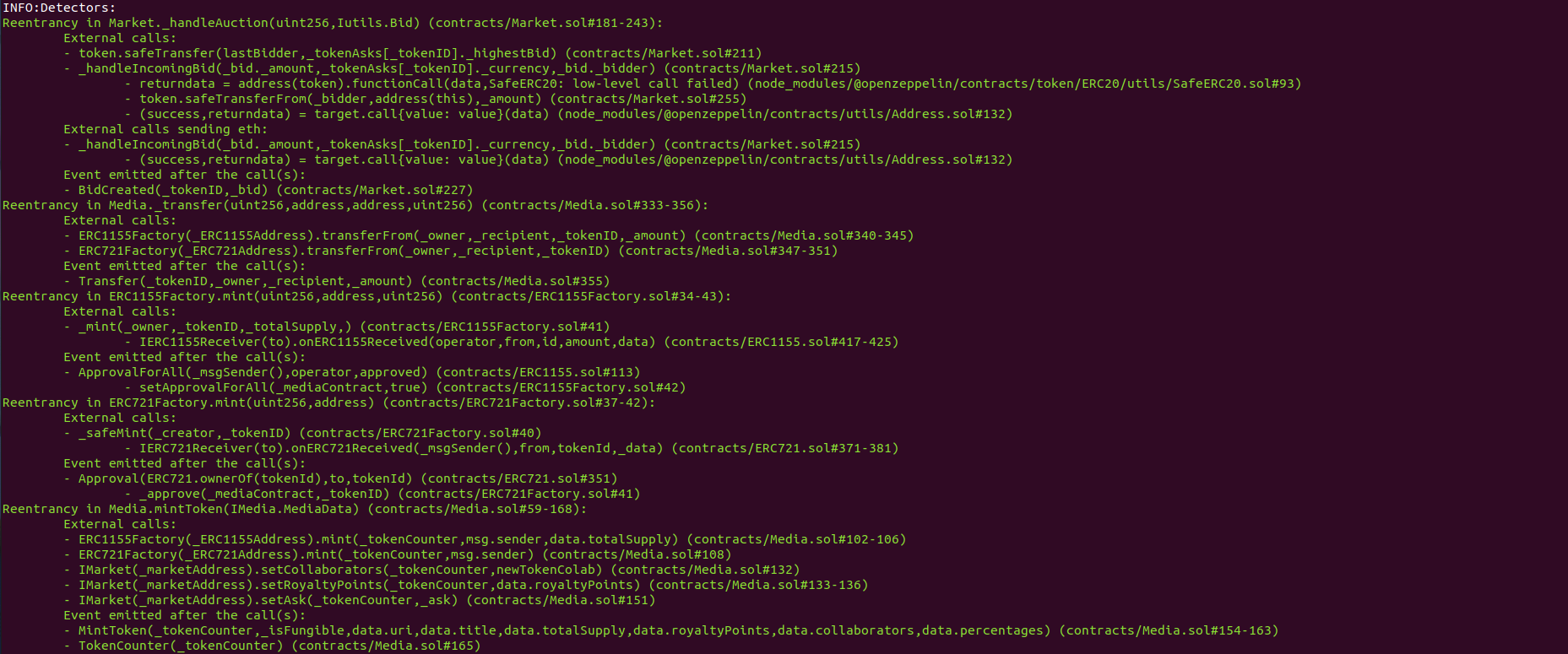
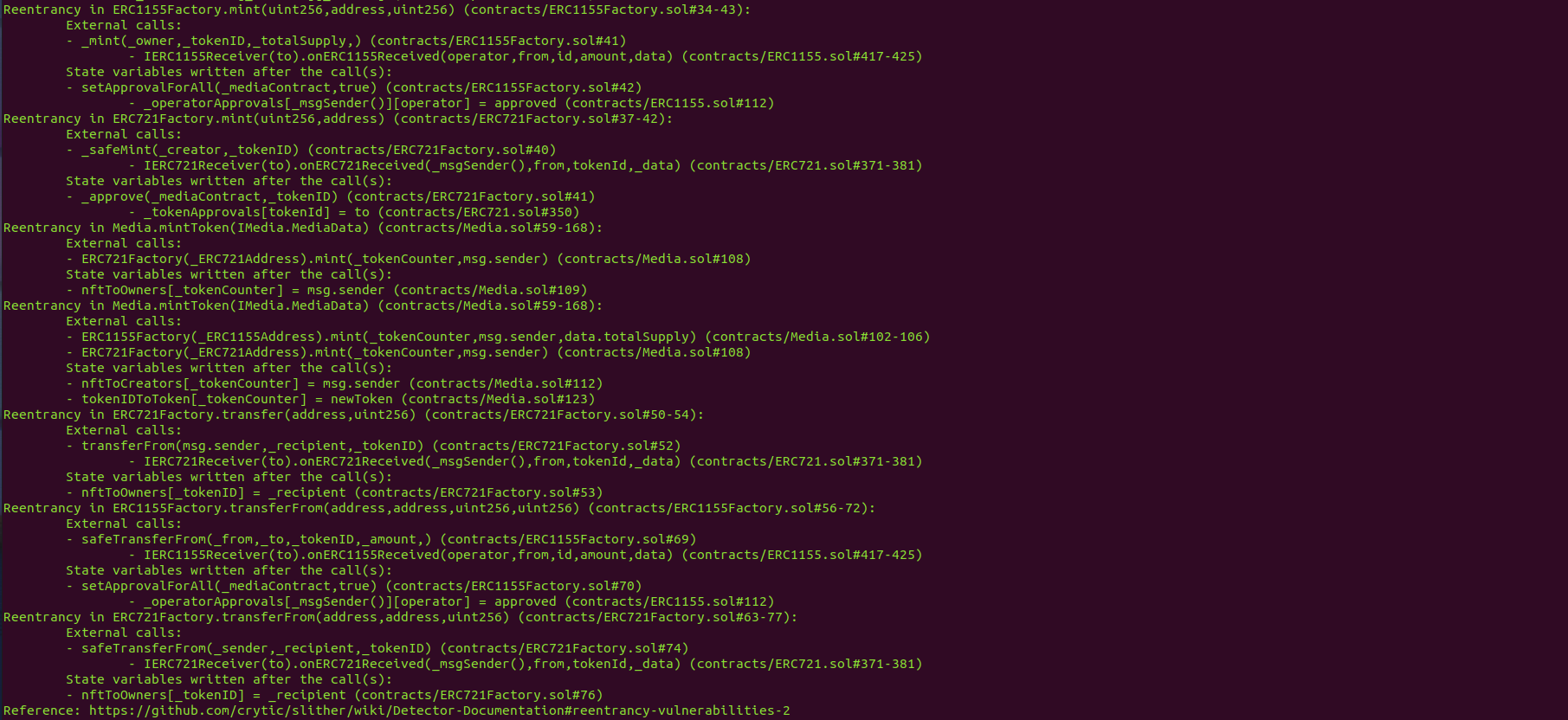
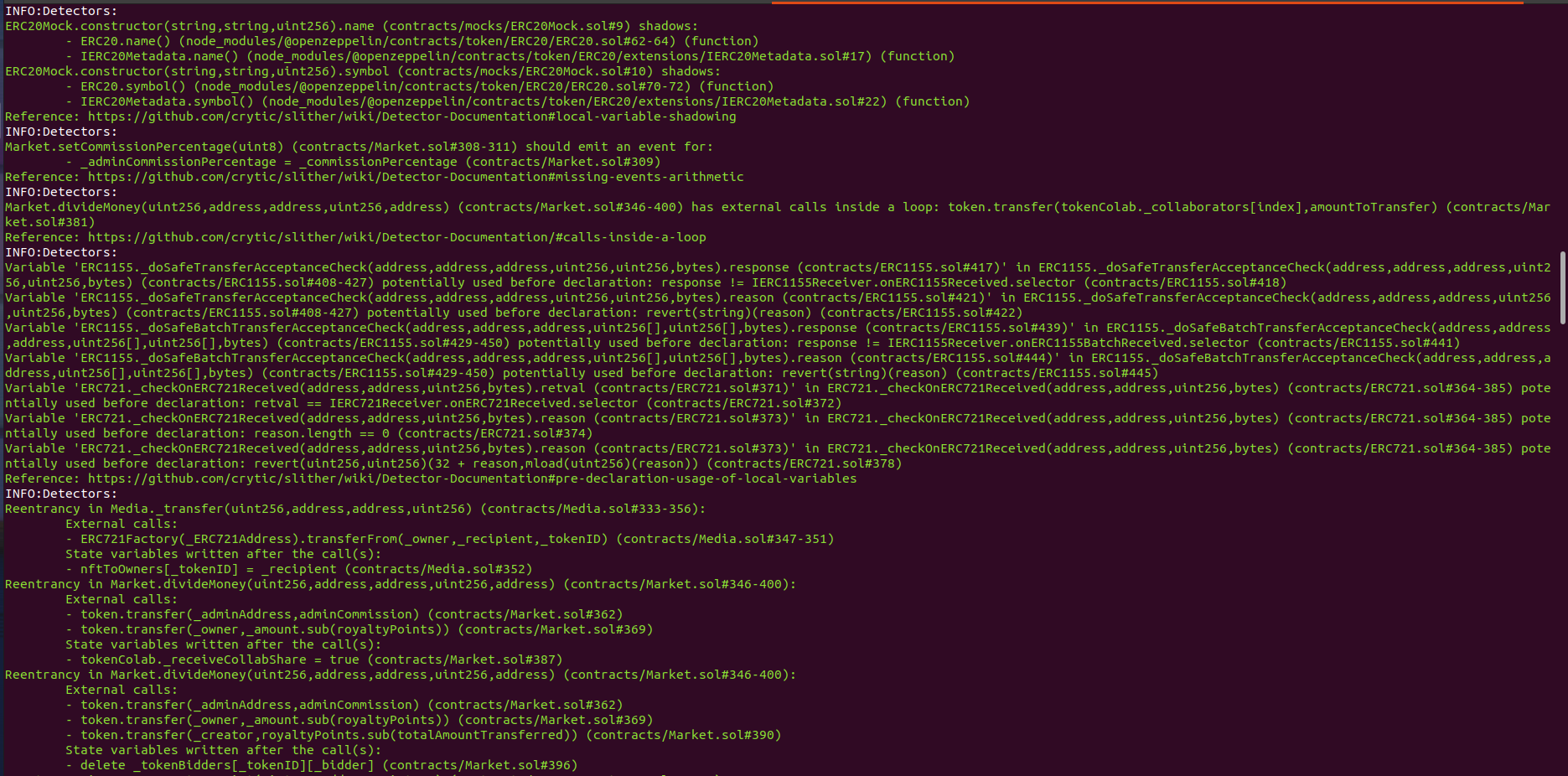
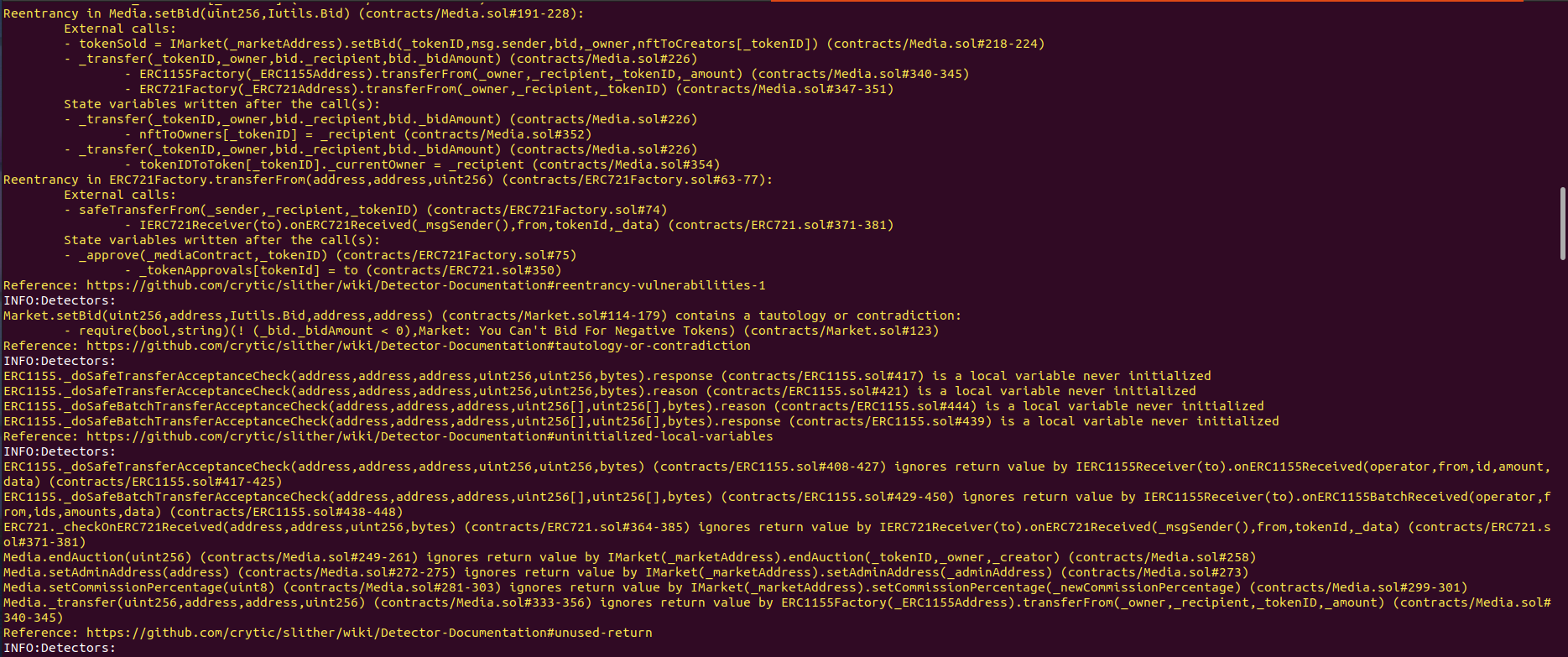
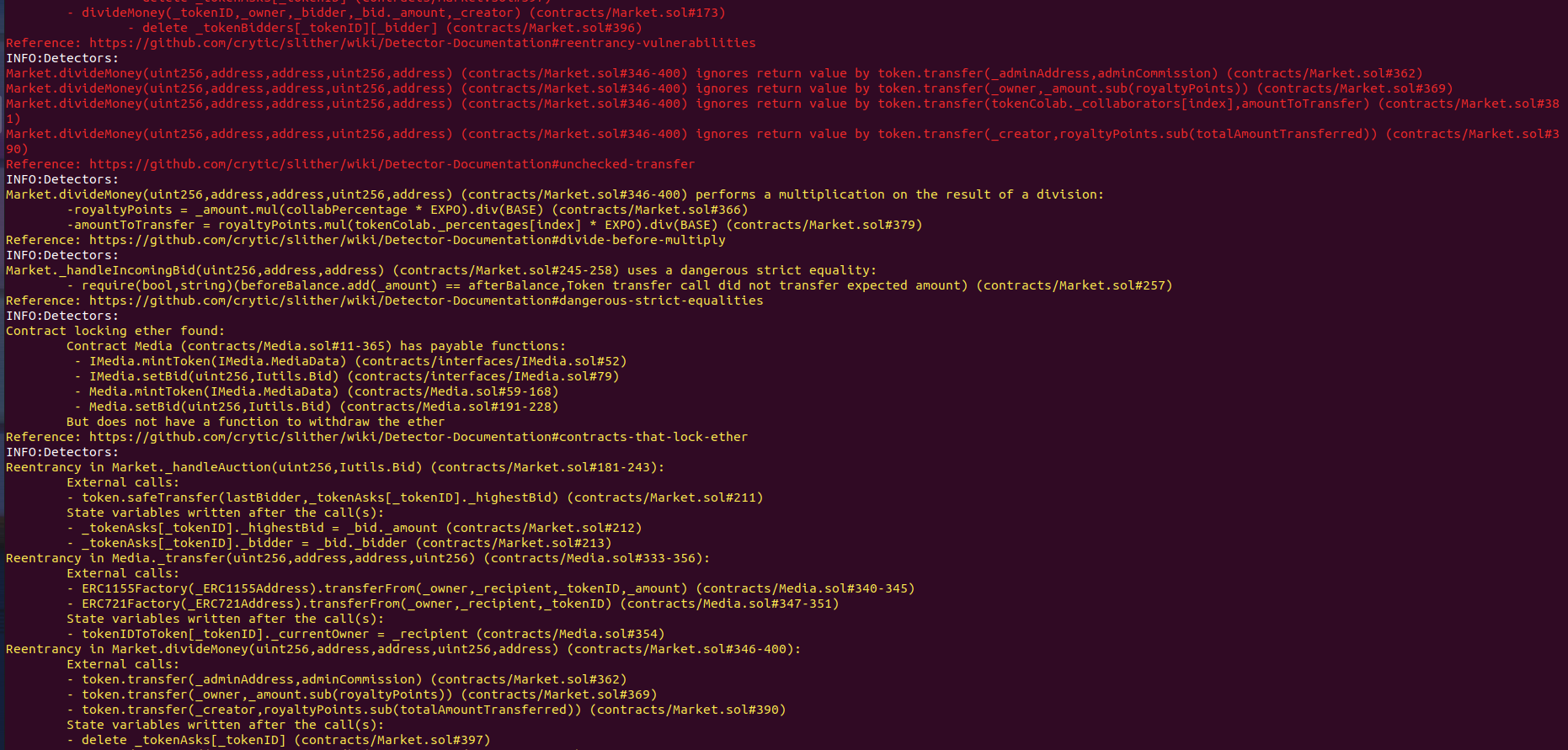
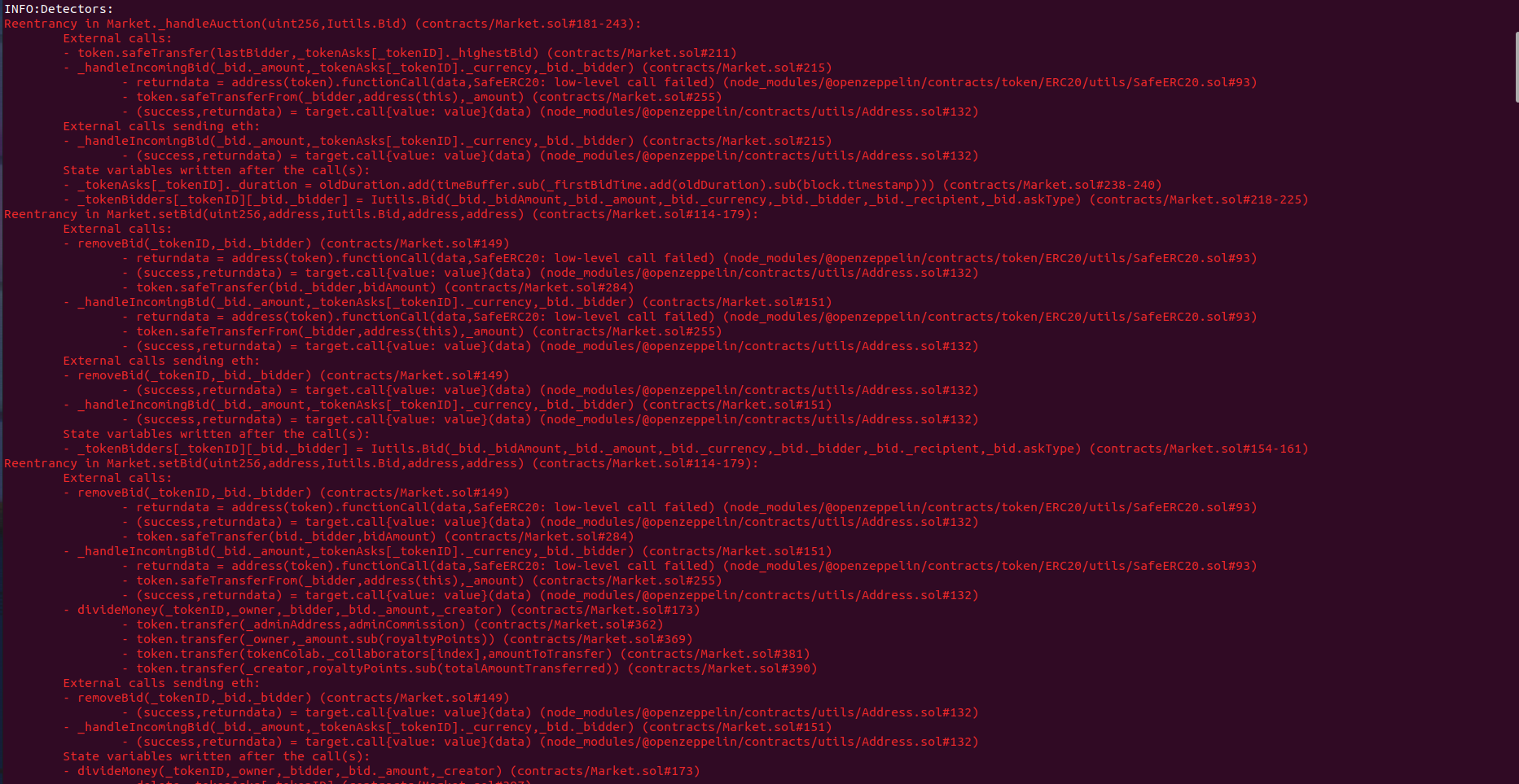
**ERC1155**

| **Function name** | **Input** | **Output** | **TX** | **Status** | **Fix Status** |
| --- | --- | --- | --- | --- | --- |
| safeTransferFrom | "0x1dd64394E29c5988f04A8E074D0DBACd4D614729","0x60b6D91cB698F41E1eD928f9631cEC6b8Ff8F6cC","1","1000",0x0000000000000000000000000000000000000000000000000000000000000000 | true | 0x765f651230b1afe737d353fb88d7456aaa84641835dfbf97785e7a7593eef68e | FAILED (this function should be restricted or set to internal) | Passed |
| safeBatchTransferFrom | "0x1dd64394E29c5988f04A8E074D0DBACd4D614729","0x60b6D91cB698F41E1eD928f9631cEC6b8Ff8F6cC",["1"],["1000"],"0x0000000000000000000000000000000000000000000000000000000000000000" | true | 0x3d56a377a0b37eee3a294d6dc84437cdfd43844c2f53eef9d1d63e45968d2d14 | FAILED (this function should be restricted or set to internal) | Passed |
| setApprovalForAll | 0x6d8595B7eAE21fcf461C192181b1242a6903b6e1, true | true | 0x00da969006b7aa5f3d45cf4fb47305fa6ced5103582ca21c56544383d7fce9a5 | FAILED (this function should be restricted or set to internal) | Passed |

**ERC721**

| **Function name** | **Input** | **Output** | **TX** | **Status** | **Fix Status** |
| --- | --- | --- | --- | --- | --- |
| safeTransferFrom | "0x1dd64394E29c5988f04A8E074D0DBACd4D614729","0x60b6D91cB698F41E1eD928f9631cEC6b8Ff8F6cC","2","0x0000000000000000000000000000000000000000000000000000000000000000" | true | 0xa7120972be522bff7a9c29221d7e23f846f25690d5c28b30f9818298e1c52271 | FAILED (this function should be restricted or set to internal) | Passed |
| safeTransferFrom | "0x60b6D91cB698F41E1eD928f9631cEC6b8Ff8F6cC","0x1dd64394E29c5988f04A8E074D0DBACd4D614729",2 | true | 0x3841662f7c4dc59127caafefa3f5b95d74114641187cf6c45840b0a619bc4572 | FAILED (this function should be restricted or set to internal) | Passed |
| approve | 0x60b6D91cB698F41E1eD928f9631cEC6b8Ff8F6cC, 2 | true | 0x14b6418fe4d90b0c07fb5439d1c261c06a2e53bc6621f80b8df494b617a5ca90 | FAILED (this function should be restricted or set to internal) | Passed |
| setApprovalForAll | 0x60b6D91cB698F41E1eD928f9631cEC6b8Ff8F6cC, true | true | 0xe29474d625326c565efd0063e8aa0e705b22c1c7a56310c295705a5e3235a708 | FAILED (this function should be restricted or set to internal) | Passed |

# Automated Tests **Slither:**

****

# Results

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

# 

# **Closing Summary**

Overall, smart contracts are very well written and adhere to guidelines.

The audit showed several high, medium, low, and informational severity issues. In the end, the majority of the issues were fixed or acknowledged by the Auditee. .

# Disclaimer

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