EtherAuthority

www.EtherAuthority.io
audit@etherauthority.io

SMART CONTRACT

Security Audit Report

Customer:	SaturnaNFT
Website:	https://www.saturna.co/
Platform:	Binance Smart Chain
Language:	Solidity
Date:	Aug 2nd, 2021

Table of contents

Introduction 4

Project Background 4

Audit Scope 4

Claimed Smart Contract Features 5

Audit Summary 6

Technical Quick Stats 7

Code Quality 8

Documentation 8

Use of Dependencies 8

AS-IS overview 9

Severity Definitions 13

Audit Findings 13

Conclusion 17

Our Methodology 18

Disclaimers 20

Appendix

- Code Flow Diagram 21
- Solidity static analysis 22
- Solhint Linter 52

THIS IS SECURITY AUDIT REPORT DOCUMENT AND WHICH MAY CONTAIN INFORMATION WHICH IS CONFIDENTIAL. WHICH INCLUDES ANY POTENTIAL VULNERABILITIES AND MALICIOUS CODES WHICH CAN BE USED TO EXPLOIT THE SOFTWARE. THIS MUST BE REFERRED INTERNALLY AND ONLY SHOULD BE MADE AVAILABLE TO PUBLIC AFTER ISSUES ARE RESOLVED.

Introduction

EtherAuthority was contracted by the SaturnaNFT team to perform the Security audit of the SaturnaNFT token smart contract code. The audit has been performed using manual analysis as well as using automated software tools. This report presents all the findings regarding the audit performed on Aug 2nd, 2021.

The purpose of this audit was to address the following:

- Ensure that all claimed functions exist and function correctly.
- Identify any security vulnerabilities that may be present in the smart contract.

Project Background

Saturna provides an opportunity to put away the past and focus on getting to the next moon. If those before have come and gone, then trust that history repeats itself and see where the future is with this next generation of tokens.

So break away from the worthless utility clones, stop worrying about APR's and Impermanent Loss, and stick with a tried-and-true smart contract code that has led prospectors to millions in realized and billions in unrealized funds, and trust one of the fastest growing movements on Binance Smart Chain.

Audit scope

Name	**Code Review and Security Analysis Report for SaturnaNFT Smart Contract**
Platform	**BSC / Solidity**
File	SaturnaNFT.sol
Smart Contract Online Code	https://docs.google.com/document/d/11-dKzQzg06rVnITzSnX-DBKuDekj3oZ7fJGTqvDxI28/edit?usp=sharing
File MD5 Hash	E1BBD92DDA39215EC932DC18E8D597EB
Audit Date	Aug 2nd, 2021

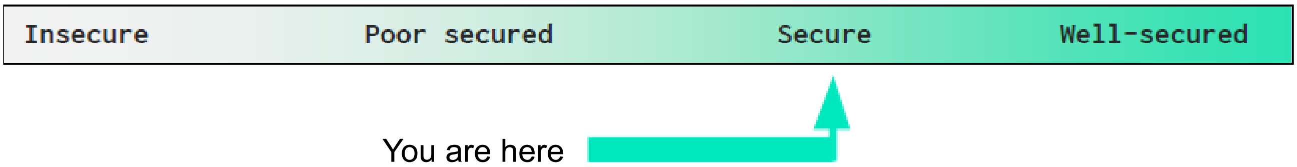
PS: There are 9 external imports from open zeppelin. These files are not included in the audit scope and thus they are not audited.

Claimed Smart Contract Features

Claimed Feature Detail	**Our Observation**
Name: Saturna NFT	**YES, This is valid.**
Symbol: SATURNA	**YES, This is valid.**
Percents Divider: 10%	**YES, This is valid.**
Max Fee Percentage: 3%	**YES, This is valid.**
● The Owner can access functions like addSupportedToken, setFeeAddress, removeSupportedToken, setFeePercent, setV1TokenAddress, setSATAddress, addAdmin, removeAdmin,etc.	**YES, This is valid. The smart contract owner controls these functions, so the owner must handle the private key of the owner's wallet very securely. Because if the private key is compromised, then it will create problems.**

Audit Summary

According to the standard audit assessment, Customer`s solidity smart contracts are **"Secured"**. These contracts also have owner functions (described in the centralization section below), which does not make everything 100% decentralized. Thus, the owner must execute those smart contract functions as per the business plan.



We used various tools like MythX, Slither and Remix IDE. At the same time this finding is based on critical analysis of the manual audit.

All issues found during automated analysis were manually reviewed and applicable vulnerabilities are presented in the Audit overview section. General overview is presented in AS-IS section and all identified issues can be found in the Audit overview section.

We found 0 critical, 0 high, 0 medium and 1 low and some very low level issues.

These issues are fixed/acknowledged in the revised smart contract code.

Investors Advice: Technical audit of the smart contract does not guarantee the ethical nature of the project. Any owner controlled functions should be executed by the owner with responsibility. All investors/users are advised to do their due diligence before investing in the project.

Technical Quick Stats

Main Category	Subcategory	Result
Contract Programming	Solidity version not specified	Passed
	Solidity version too old	Moderated
	Integer overflow/underflow	Passed
	Function input parameters lack of check	Moderated
	Function input parameters check bypass	Passed
	Function access control lacks management	Passed
	Critical operation lacks event log	Passed
	Human/contract checks bypass	Passed
	Random number generation/use vulnerability	Passed
	Fallback function misuse	Passed
	Race condition	Passed
	Logical vulnerability	Passed
	Features claimed	Passed
	Other programming issues	Passed
Code Specification	Function visibility not explicitly declared	Passed
	Var. storage location not explicitly declared	Passed
	Use keywords/functions to be deprecated	Passed
	Other code specification issues	Passed
Gas Optimization	“Out of Gas” Issue	Passed
	High consumption ‘for/while’ loop	Passed
	High consumption ‘storage’ storage	Passed
	Assert() misuse	Passed
Business Risk	The maximum limit for mintage not set	Passed
	“Short Address” Attack	Passed
	“Double Spend” Attack	Passed

Overall Audit Result: PASSED

Code Quality

This audit scope has 1 smart contract. This smart contract also contains Libraries, Smart contracts inherits and Interfaces. These are compact and well written contracts.

The libraries in SaturnaNFT are part of its logical algorithm. A library is a different type of smart contract that contains reusable code. Once deployed on the blockchain (only once), it is assigned a specific address and its properties / methods can be reused many times by other contracts in the SaturnaNFT token.

The SaturnaNFT team has **not** provided scenario and unit test scripts, which would have helped to determine the integrity of the code in an automated way.

Some code parts are **not well** commented on smart contracts.

Documentation

We were given a SaturnaNFT smart contract code in the form of a google document link. The hashes of that code are mentioned above in the table.

As mentioned above, some code parts are **not well** commented. So it is difficult to quickly understand the programming flow as well as complex code logic. Comments are very helpful in understanding the overall architecture of the protocol.

Another source of information was its official website https://www.saturna.co/ which provided rich information about the project architecture and tokenomics.

Use of Dependencies

As per our observation, the libraries are used in this smart contract infrastructure that are based on well known industry standard open source projects. And their core code blocks are written well.

Apart from libraries, its functions are used in external smart contract calls.

AS-IS overview

SaturnaNFT.sol

(1) Interface

(a) IERC20

(b) IERC165

(c) IERC721

(d) IERC721Enumerable

(e) IERC721Receiver

(f) IERC721Metadata

(2) Inherited contracts

(a) ERC721

(b) ERC721Enumerable

(c) ERC721URIStorage

(d) Ownable

(e) ReentrancyGuard

(3) Usages

(a) using Counters for Counters.Counter;

(b) using SafeMath for uint256;

(c) using EnumerableSet for EnumerableSet.AddressSet;

(4) Struct

(a) Item

(b) Condition

(c) Bid

(d) Info

(5) Events

(a) event ItemCreated(uint256 itemId);

(b) event ItemImported(uint256 itemId);

(c) event BidAddedToItem(address bidder , uint256 itemId , uint256 bidAmount);

(d) event AuctionCancelled(uint256 itemId);

(e) event ListCancelled(uint256 itemId);

(f) event ItemSold(address buyer , uint256 tokenID, uint256 itemId, uint256 amount);

(g) event UpdatedSupportCurrency(address currency , bool acceptable);

(6) Functions

Sl.	Functions	Type	Observation	Conclusion
1	supportsInterface	read	Passed	No Issue
2	_burn	internal	Passed	No Issue
3	tokenURI	read	Passed	No Issue
4	_beforeTokenTransfer	internal	Passed	No Issue
5	createItem	external	Passed	No Issue
6	importItem	external	Passed	No Issue
7	buyItem	external	Passed	No Issue
8	placeBid	external	Passed	No Issue
9	cancelAuction	write	access only ItemCreator	No Issue
10	cancelList	write	access only ItemCreator	No Issue
11	acceptWinningBid	external	access only ItemCreator	No Issue
12	_checkCanBuy	internal	Passed	No Issue
13	_processPayment	internal	Function input parameters lack of check	Refer Audit Findings
14	_refundLastBid	internal	Passed	No Issue
15	_payoutUser	internal	Function input parameters lack of check	Refer Audit Findings
16	calcBidAmount	internal	Function input parameters lack of check	Refer Audit Findings
17	lastBid	read	Passed	No Issue
18	itemCondition	read	Passed	No Issue
19	itemURI	read	Passed	No Issue
20	_safeTransferBNB	internal	Low level function calls used	Refer Audit Findings
21	receive	external	Passed	No Issue
22	addSupportedToken	write	access only Owner	No Issue
23	isSupportedToken	read	Passed	No Issue
24	removeSupportedToken	external	access only Owner	No Issue
25	supportedTokenAt	read	Passed	No Issue
26	supportedTokensLength	read	Passed	No Issue
27	setFeeAddress	external	access only Owner	No Issue
28	setFeePercent	external	access only Owner	No Issue
29	setV1TokenAddress	external	access only Owner	No Issue

30	setSATAddress	external	access only Owner	No Issue
31	addAdmin	write	access only Owner	No Issue
32	removeAdmin	write	access only Owner	No Issue
33	onlyAdmin	modifier	Passed	No Issue
34	onlyItemCreator	modifier	Passed	No Issue
35	supportsInterface	read	Passed	No Issue
36	balanceOf	read	Passed	No Issue
37	ownerOf	read	Passed	No Issue
38	name	read	Passed	No Issue
39	symbol	read	Passed	No Issue
40	tokenURI	read	Passed	No Issue
41	_baseURI	internal	Passed	No Issue
42	approve	write	Passed	No Issue
43	getApproved	read	Passed	No Issue
44	setApprovalForAll	write	Passed	No Issue
45	isApprovedForAll	read	Passed	No Issue
46	transferFrom	write	Passed	No Issue
47	safeTransferFrom	write	Passed	No Issue
48	safeTransferFrom	write	Passed	No Issue
49	_safeTransfer	internal	Passed	No Issue
50	_exists	internal	Passed	No Issue
51	_isApprovedOrOwner	internal	Passed	No Issue
52	_safeMint	internal	Passed	No Issue
53	_safeMint	internal	Passed	No Issue
54	_mint	internal	Passed	No Issue
55	_burn	internal	Passed	No Issue
56	_transfer	internal	Passed	No Issue
57	_approve	internal	Passed	No Issue
58	_checkOnERC721Received	write	Passed	No Issue
59	_beforeTokenTransfer	internal	Passed	No Issue
60	supportsInterface	read	Passed	No Issue
61	tokenOfOwnerByIndex	read	Passed	No Issue
62	totalSupply	read	Passed	No Issue
63	tokenByIndex	read	Passed	No Issue
64	_beforeTokenTransfer	internal	Passed	No Issue
65	_addTokenToOwnerEnumeration	write	Passed	No Issue
66	_addTokenToAllTokensEnumeration	write	Passed	No Issue
67	_removeTokenFromOwnerEnumeration	write	Passed	No Issue
68	_removeTokenFromAllTokensEnumeration	write	Passed	No Issue
69	tokenURI	read	Passed	No Issue
70	_setTokenURI	internal	Passed	No Issue
71	_burn	internal	Passed	No Issue
72	owner	read	Passed	No Issue
73	onlyOwner	modifier	Passed	No Issue

74	renounceOwnership	write	access only Owner	No Issue
75	transferOwnership	write	access only Owner	No Issue
76	nonReentrant	modifier	Passed	No Issue

Severity Definitions

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

Audit Findings

Critical

No Critical severity vulnerabilities were found.

High

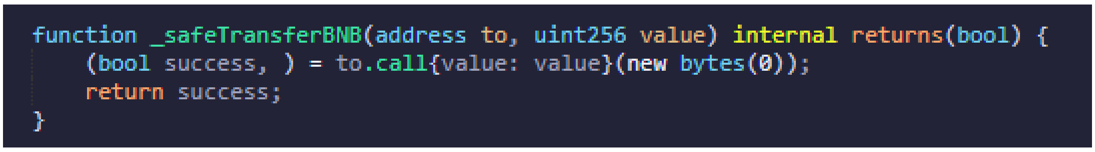
No High severity vulnerabilities were found.

Medium

No Medium severity vulnerabilities were found.

Low

(1) Low level function calls used:



```
function _safeTransferBNB(address to, uint256 value) internal returns(bool) {
    (bool success, ) = to.call{value: value}(new bytes(0));
    return success;
}
```

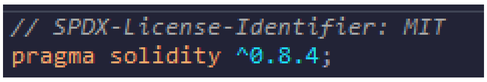
Usually an attacker can create an attack smart contract and use it to reenter. In line number #476, if the _to is an attack smart contract, then sending ether to that wallet will trigger that attack contract's fallback function and either he can reenter or can call any other function which is dependent on _safeTransferBNB function's state.

Resolution: To prevent such scenarios from rising, it is always better to use .transfer() to send the ether to any wallet as it only allocates 21,000 gas which is not enough to reenter.

Status: acknowledged.

Very Low / Discussion / Best practices:

(1) Use the latest solidity version:



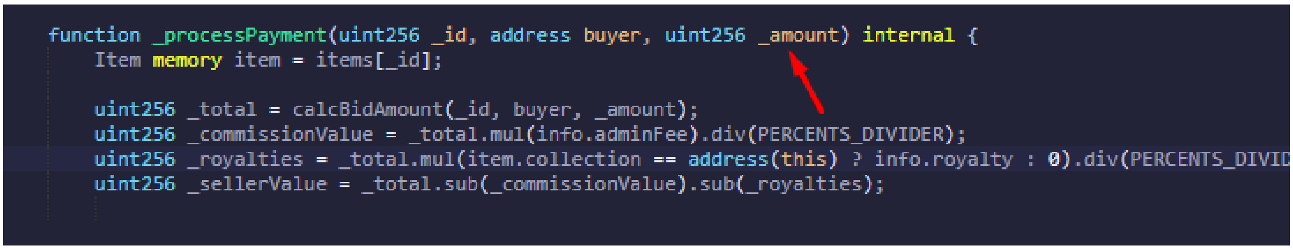
```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.4;
```

Using the latest solidity will prevent any compiler-level bugs.

Resolution: Please use 0.8.6 which is the latest version.

Status: acknowledged.

(2) Function input parameters lack of check:



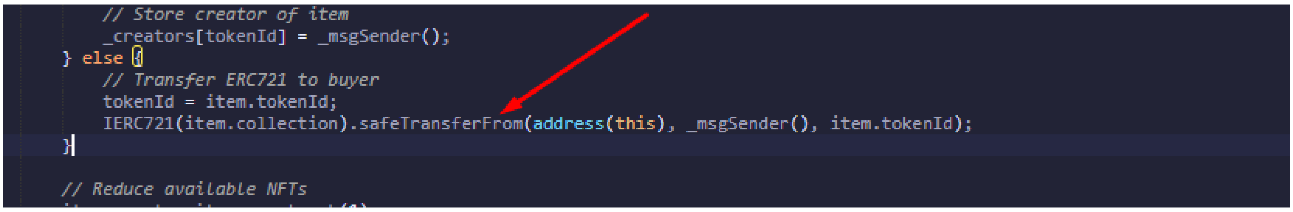
```
function _processPayment(uint256 _id, address buyer, uint256 _amount) internal {
    Item memory item = items[_id];

    uint256 _total = calcBidAmount(_id, buyer, _amount);
    uint256 _commissionValue = _total.mul(info.adminFee).div(PERCENTS_DIVIDER);
    uint256 _royalties = _total.mul(item.collection == address(this) ? info.royalty : 0).div(PERCENTS_DIVID
    uint256 _sellerValue = _total.sub(_commissionValue).sub(_royalties);
```

Variable validation is not performed in below functions : _processPayment , calcBidAmount , _payoutUser

Resolution: Put validation : variable _amount should not be empty and > 0

Status: acknowledged.

(3) Uses of safeTransferFrom when send token from contract/owner to any user:



```
        // Store creator of item
        _creators[tokenId] = _msgSender();
    } else {
        // Transfer ERC721 to buyer
        tokenId = item.tokenId;
        IERC721(item.collection).safeTransferFrom(address(this), _msgSender(), item.tokenId);
    }

    // Reduce available NFTs
```

If the asset is being transferred from contract (address this) to user, then there is no need to use safeTranseferFrom. This will cause it to issue additional approval. Best practice is to just use the safeTransfer() method.

Status: acknowledged.

(4) Internal instead of External:

All functions which are not called internally, must be declared as external. It is more efficient as sometimes it saves some gas.

https://ethereum.stackexchange.com/questions/19380/external-vs-public-best-practices

Status: acknowledged.

Centralization

These smart contracts have some functions which can be executed by Admin (Owner) only. If the admin wallet private key would be compromised, then it would create trouble. Following are Admin functions:

- addSupportedToken: The Owner can add supported token addresses.
- removeSupportedToken: The Owner can remove supported token addresses.
- setFeeAddress: The Owner can set Fee addresses.
- setFeePercent: The Owner can set a fee percentage.
- setV1TokenAddress: The Owner can check if the address is valid or not then set v1NFTAddress.
- setSATAddress: The Owner can check if the address is valid or not then set satTokenAddress.
- addAdmin: The Owner can check if the admin Address is the zero address or not then add a new the admin address.
- removeAdmin: The Owner can check if the admin Address is the zero address or not then remove the admin address.

Conclusion

We were given a contract code. And we have used all possible tests based on given objects as files. We observed some issues in the smart contracts and those issues are fixed in revised code. So, **it's good to go to production**.

Since possible test cases can be unlimited for such smart contracts protocol, we provide no such guarantee of future outcomes. We have used all the latest static tools and manual observations to cover maximum possible test cases to scan everything.

Smart contracts within the scope were manually reviewed and analyzed with static analysis tools. Smart Contract's high level description of functionality was presented in As-is overview section of the report.

Audit report contains all found security vulnerabilities and other issues in the reviewed code.

Security state of the reviewed contract, based on standard audit procedure scope, is **"Secured".**

Our Methodology

We like to work with a transparent process and make our reviews a collaborative effort. The goals of our security audits are to improve the quality of systems we review and aim for sufficient remediation to help protect users. The following is the methodology we use in our security audit process.

Manual Code Review:

In manually reviewing all of the code, we look for any potential issues with code logic, error handling, protocol and header parsing, cryptographic errors, and random number generators. We also watch for areas where more defensive programming could reduce the risk of future mistakes and speed up future audits. Although our primary focus is on the in-scope code, we examine dependency code and behavior when it is relevant to a particular line of investigation.

Vulnerability Analysis:

Our audit techniques included manual code analysis, user interface interaction, and whitebox penetration testing. We look at the project's web site to get a high level understanding of what functionality the software under review provides. We then meet with the developers to gain an appreciation of their vision of the software. We install and use the relevant software, exploring the user interactions and roles. While we do this, we brainstorm threat models and attack surfaces. We read design documentation, review other audit results, search for similar projects, examine source code dependencies, skim open issue tickets, and generally investigate details other than the implementation.

Documenting Results:

We follow a conservative, transparent process for analyzing potential security vulnerabilities and seeing them through successful remediation. Whenever a potential issue is discovered, we immediately create an Issue entry for it in this document, even though we have not yet verified the feasibility and impact of the issue. This process is conservative because we document our suspicions early even if they are later shown to not represent exploitable vulnerabilities. We generally follow a process of first documenting the suspicion with unresolved questions, then confirming the issue through code analysis, live experimentation, or automated tests. Code analysis is the most tentative, and we strive to provide test code, log captures, or screenshots demonstrating our confirmation. After this we analyze the feasibility of an attack in a live system.

Suggested Solutions:

We search for immediate mitigations that live deployments can take, and finally we suggest the requirements for remediation engineering for future releases. The mitigation and remediation recommendations should be scrutinized by the developers and deployment engineers, and successful mitigation and remediation is an ongoing collaborative process after we deliver our report, and before the details are made public.

Disclaimers

EtherAuthority.io Disclaimer

EtherAuthority team has analyzed this smart contract in accordance with the best industry practices at the date of this report, in relation to: cybersecurity vulnerabilities and issues in smart contract source code, the details of which are disclosed in this report, (Source Code); the Source Code compilation, deployment and functionality (performing the intended functions).

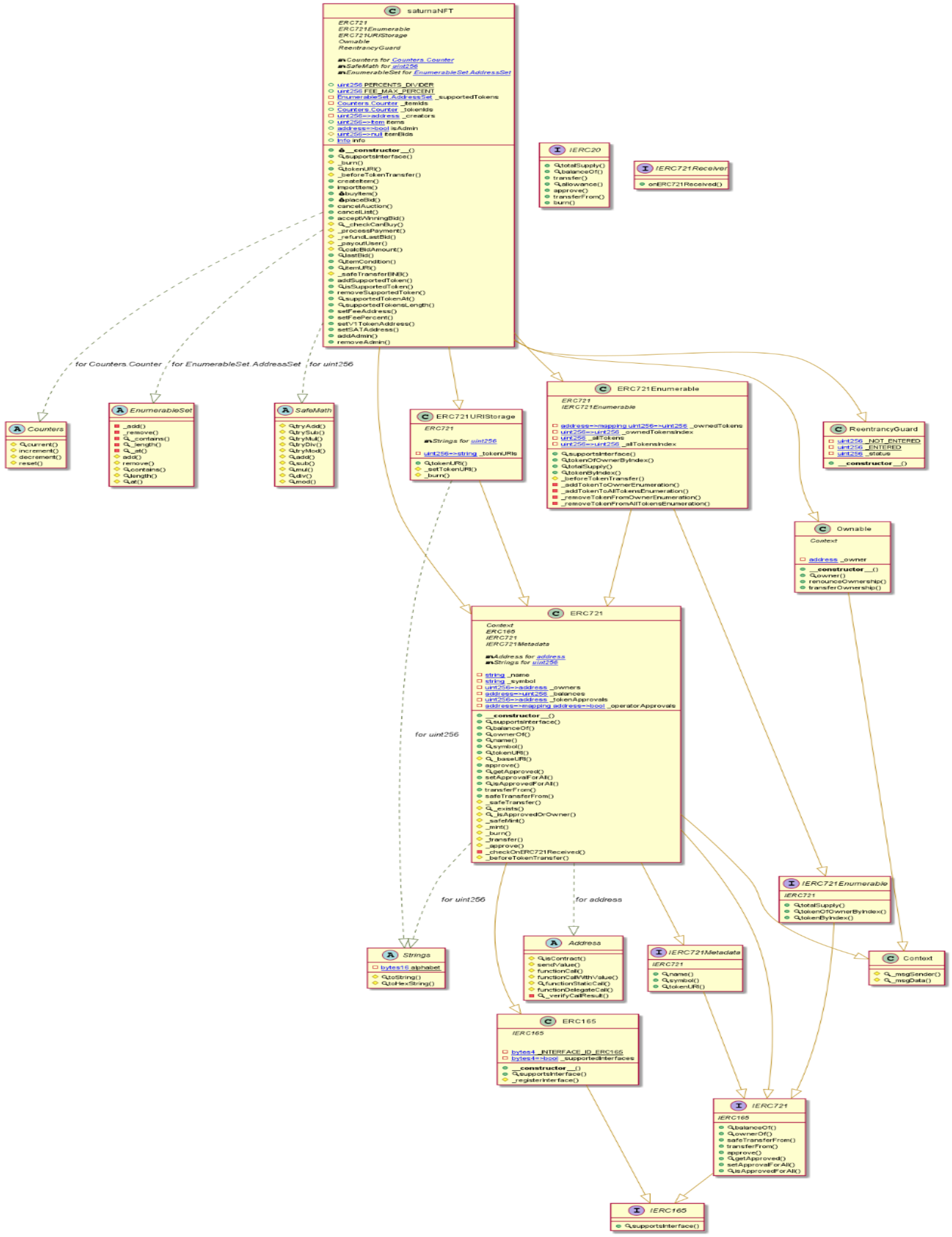
Due to the fact that the total number of test cases are unlimited, the audit makes no statements or warranties on security of the code. It also cannot be considered as a sufficient assessment regarding the utility and safety of the code, bugfree status or any other statements of the contract. While we have done our best in conducting the analysis and producing this report, it is important to note that you should not rely on this report only. We also suggest conducting a bug bounty program to confirm the high level of security of this smart contract.

Technical Disclaimer

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

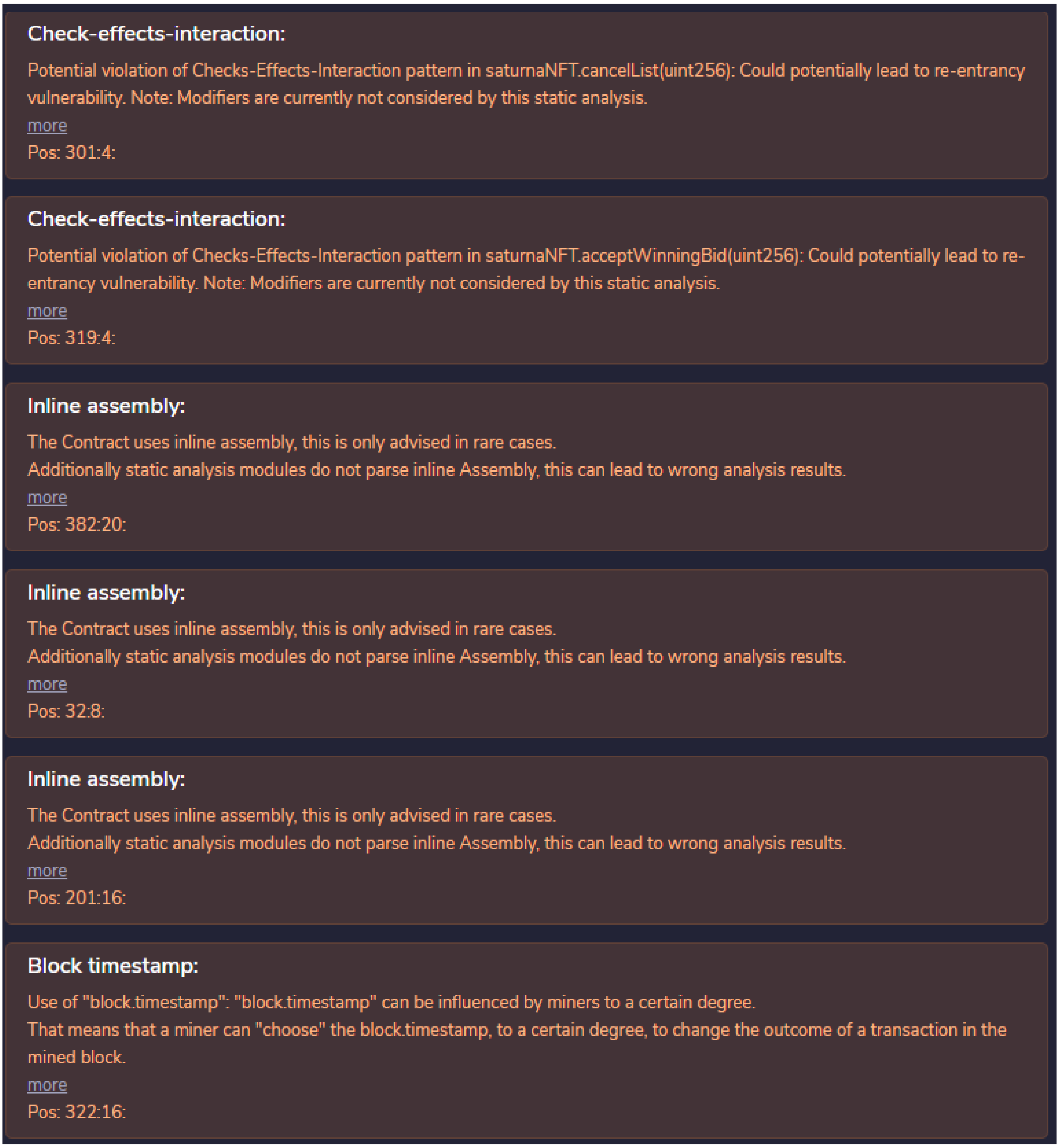
Appendix

Code Flow Diagram - SaturnaNFT



Solidity static analysis

SaturnaNFT.sol



Check-effects-interaction:

Potential violation of Checks-Effects-Interaction pattern in saturnaNFT.cancelList(uint256): Could potentially lead to re-entrancy vulnerability. Note: Modifiers are currently not considered by this static analysis.

more

Pos: 301:4:

Check-effects-interaction:

Potential violation of Checks-Effects-Interaction pattern in saturnaNFT.acceptWinningBid(uint256): Could potentially lead to re-entrancy vulnerability. Note: Modifiers are currently not considered by this static analysis.

more

Pos: 319:4:

Inline assembly:

The Contract uses inline assembly, this is only advised in rare cases.
Additionally static analysis modules do not parse inline Assembly, this can lead to wrong analysis results.

more

Pos: 382:20:

Inline assembly:

The Contract uses inline assembly, this is only advised in rare cases.
Additionally static analysis modules do not parse inline Assembly, this can lead to wrong analysis results.

more

Pos: 32:8:

Inline assembly:

The Contract uses inline assembly, this is only advised in rare cases.
Additionally static analysis modules do not parse inline Assembly, this can lead to wrong analysis results.

more

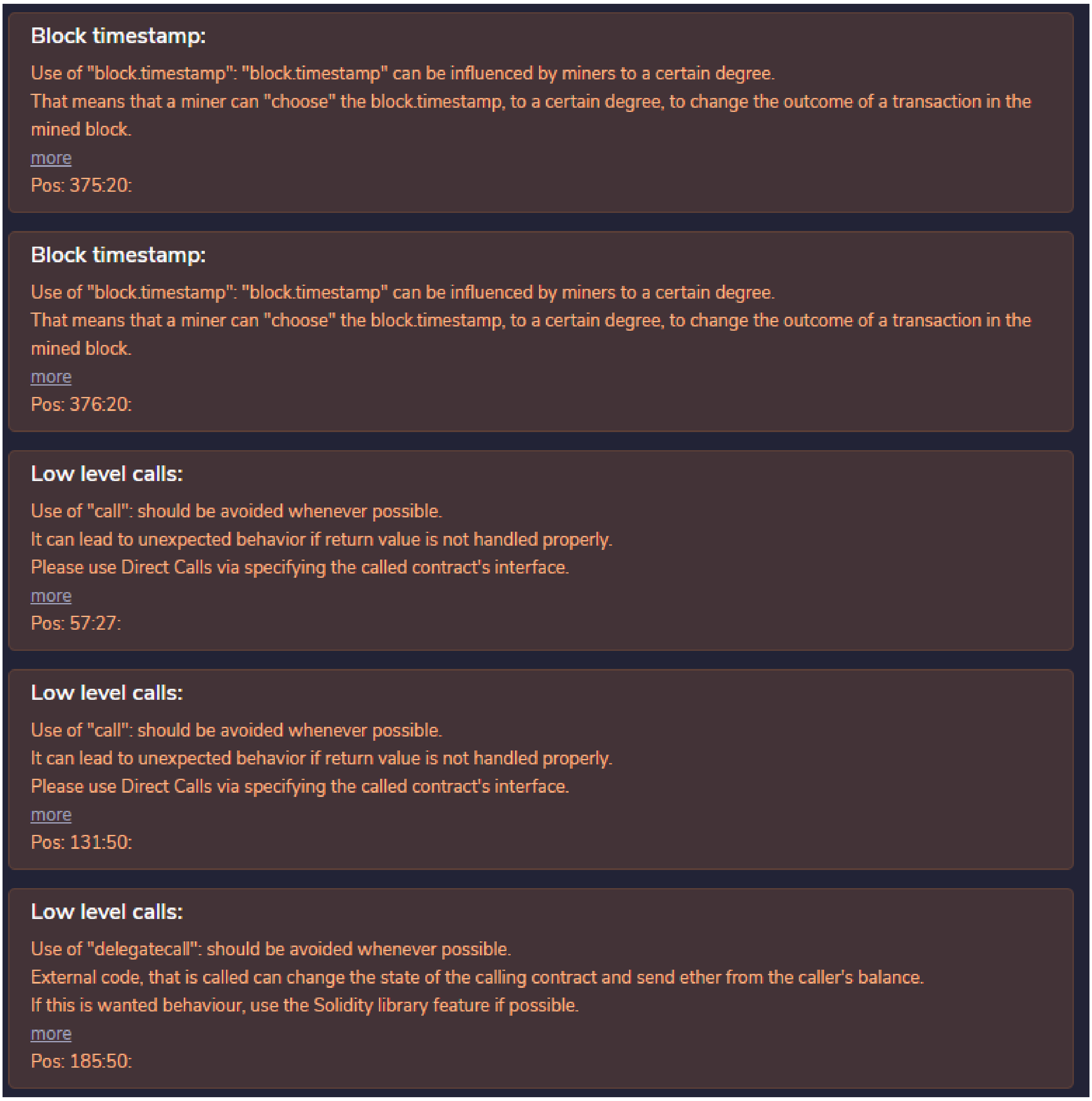
Pos: 201:16:

Block timestamp:

Use of "block.timestamp": "block.timestamp" can be influenced by miners to a certain degree.
That means that a miner can "choose" the block.timestamp, to a certain degree, to change the outcome of a transaction in the mined block.

more

Pos: 322:16:



Block timestamp:

Use of "block.timestamp": "block.timestamp" can be influenced by miners to a certain degree.
That means that a miner can "choose" the block.timestamp, to a certain degree, to change the outcome of a transaction in the mined block.
more
Pos: 375:20:

Block timestamp:

Use of "block.timestamp": "block.timestamp" can be influenced by miners to a certain degree.
That means that a miner can "choose" the block.timestamp, to a certain degree, to change the outcome of a transaction in the mined block.
more
Pos: 376:20:

Low level calls:

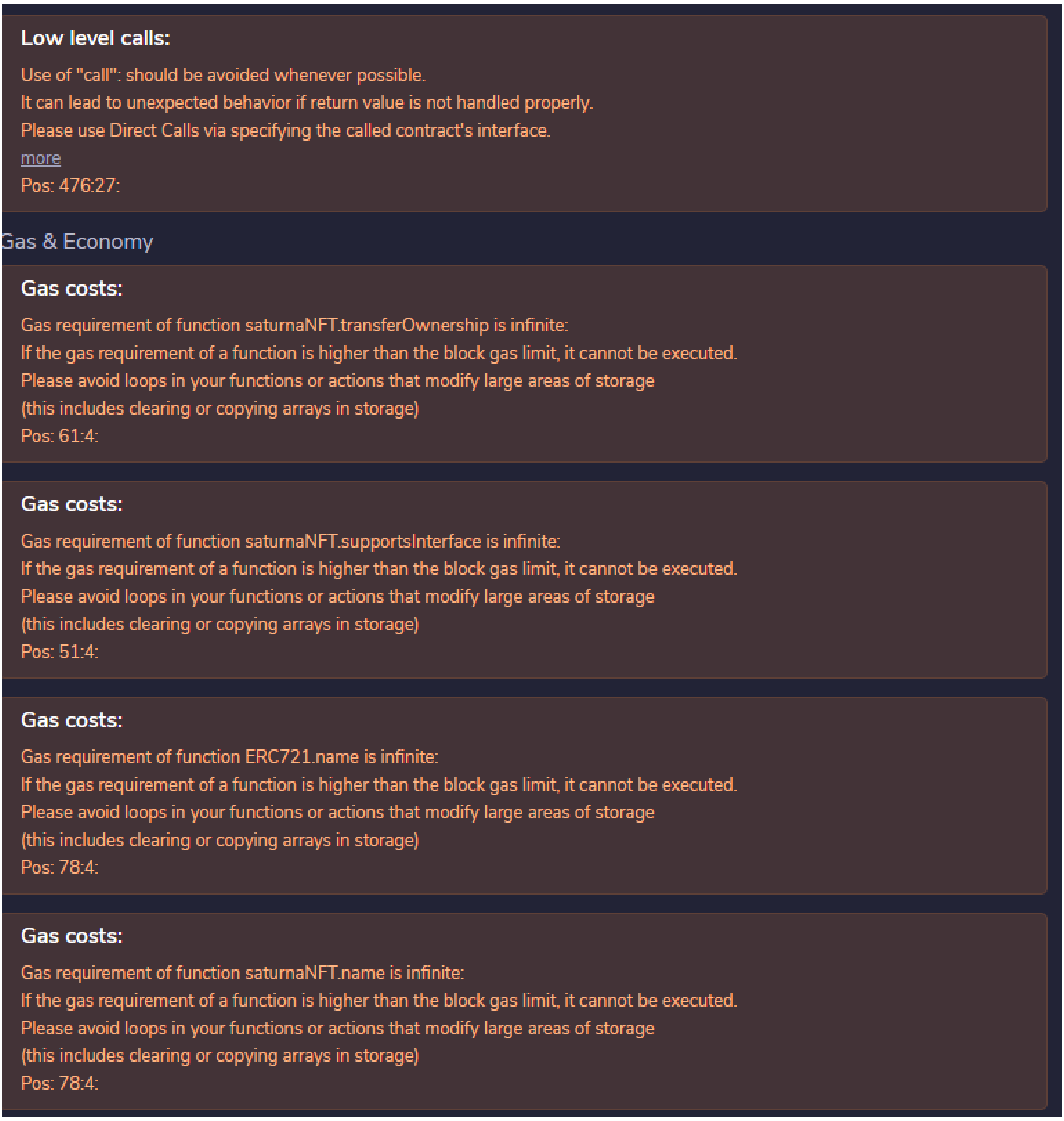
Use of "call": should be avoided whenever possible.
It can lead to unexpected behavior if return value is not handled properly.
Please use Direct Calls via specifying the called contract's interface.
more
Pos: 57:27:

Low level calls:

Use of "call": should be avoided whenever possible.
It can lead to unexpected behavior if return value is not handled properly.
Please use Direct Calls via specifying the called contract's interface.
more
Pos: 131:50:

Low level calls:

Use of "delegatecall": should be avoided whenever possible.
External code, that is called can change the state of the calling contract and send ether from the caller's balance.
If this is wanted behaviour, use the Solidity library feature if possible.
more
Pos: 185:50:



Low level calls:

Use of "call": should be avoided whenever possible.
It can lead to unexpected behavior if return value is not handled properly.
Please use Direct Calls via specifying the called contract's interface.
more
Pos: 476:27:

Gas & Economy

Gas costs:

Gas requirement of function saturnaNFT.transferOwnership is infinite:
If the gas requirement of a function is higher than the block gas limit, it cannot be executed.
Please avoid loops in your functions or actions that modify large areas of storage
(this includes clearing or copying arrays in storage)
Pos: 61:4:

Gas costs:

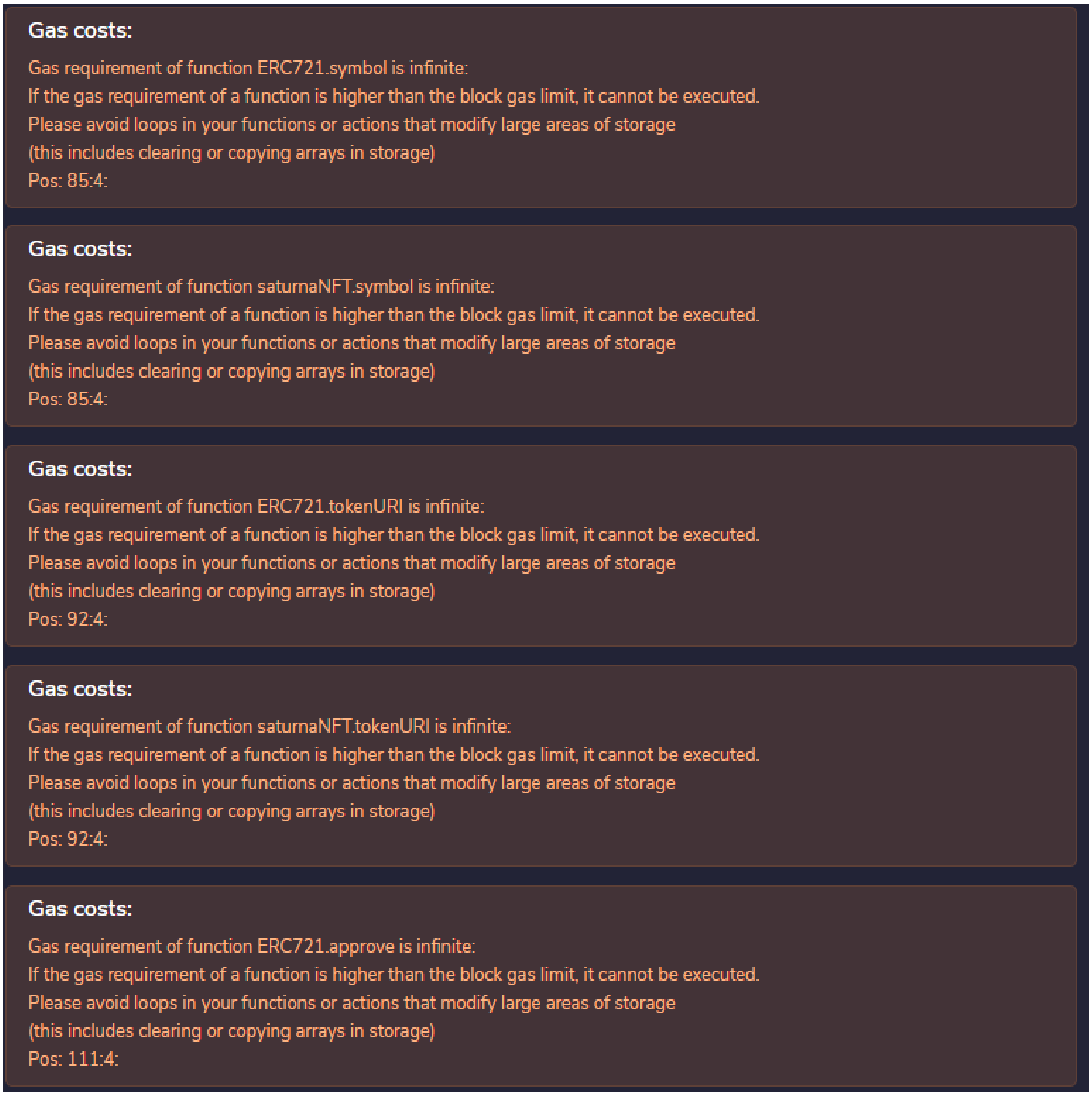
Gas requirement of function saturnaNFT.supportsInterface is infinite:
If the gas requirement of a function is higher than the block gas limit, it cannot be executed.
Please avoid loops in your functions or actions that modify large areas of storage
(this includes clearing or copying arrays in storage)
Pos: 51:4:

Gas costs:

Gas requirement of function ERC721.name is infinite:
If the gas requirement of a function is higher than the block gas limit, it cannot be executed.
Please avoid loops in your functions or actions that modify large areas of storage
(this includes clearing or copying arrays in storage)
Pos: 78:4:

Gas costs:

Gas requirement of function saturnaNFT.name is infinite:
If the gas requirement of a function is higher than the block gas limit, it cannot be executed.
Please avoid loops in your functions or actions that modify large areas of storage
(this includes clearing or copying arrays in storage)
Pos: 78:4:



Gas costs:

Gas requirement of function ERC721.symbol is infinite:
If the gas requirement of a function is higher than the block gas limit, it cannot be executed.
Please avoid loops in your functions or actions that modify large areas of storage
(this includes clearing or copying arrays in storage)
Pos: 85:4:

Gas costs:

Gas requirement of function saturnaNFT.symbol is infinite:
If the gas requirement of a function is higher than the block gas limit, it cannot be executed.
Please avoid loops in your functions or actions that modify large areas of storage
(this includes clearing or copying arrays in storage)
Pos: 85:4:

Gas costs:

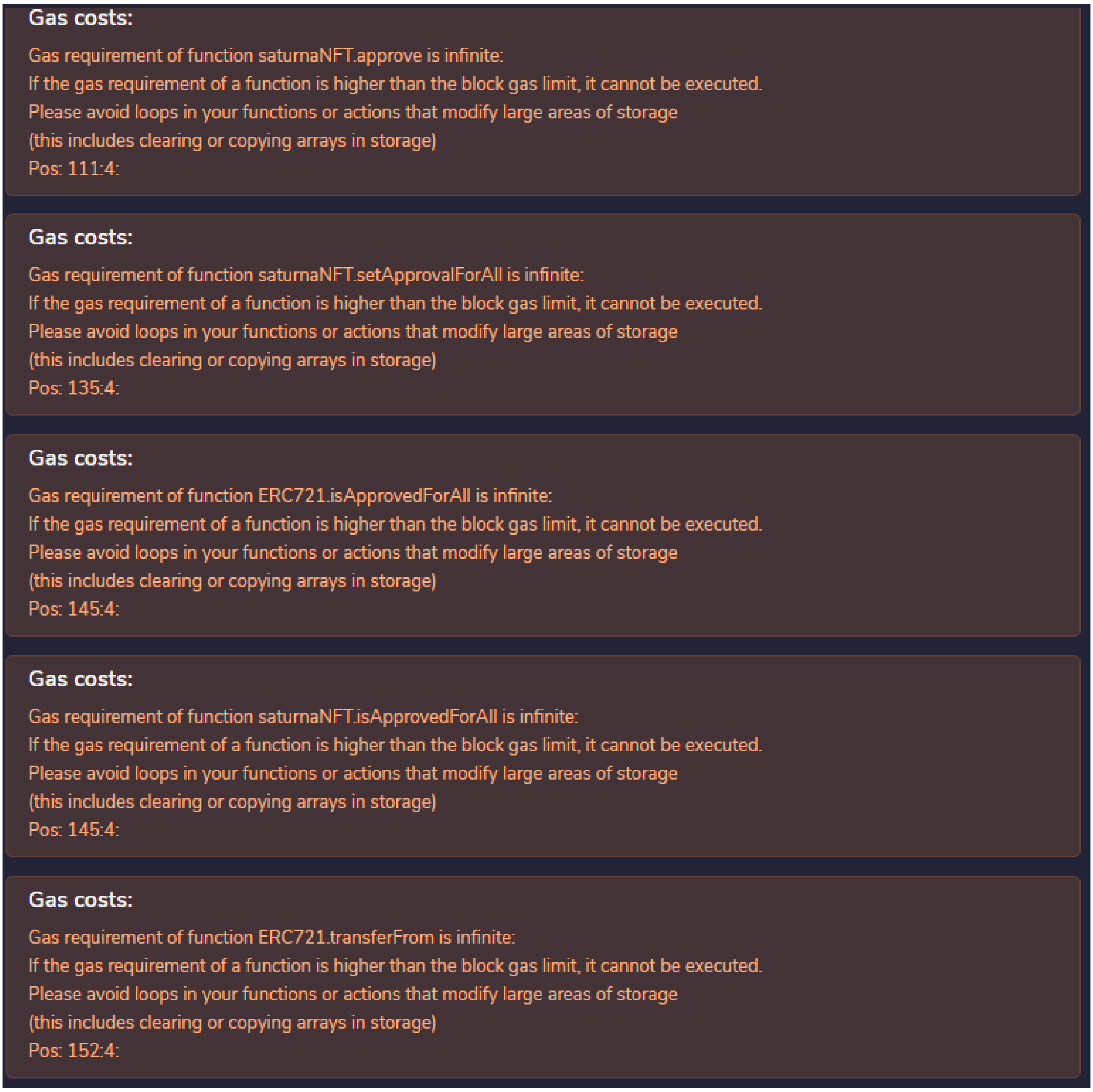
Gas requirement of function ERC721.tokenURI is infinite:
If the gas requirement of a function is higher than the block gas limit, it cannot be executed.
Please avoid loops in your functions or actions that modify large areas of storage
(this includes clearing or copying arrays in storage)
Pos: 92:4:

Gas costs:

Gas requirement of function saturnaNFT.tokenURI is infinite:
If the gas requirement of a function is higher than the block gas limit, it cannot be executed.
Please avoid loops in your functions or actions that modify large areas of storage
(this includes clearing or copying arrays in storage)
Pos: 92:4:

Gas costs:

Gas requirement of function ERC721.approve is infinite:
If the gas requirement of a function is higher than the block gas limit, it cannot be executed.
Please avoid loops in your functions or actions that modify large areas of storage
(this includes clearing or copying arrays in storage)
Pos: 111:4:



Gas costs:

Gas requirement of function saturnaNFT.approve is infinite:
If the gas requirement of a function is higher than the block gas limit, it cannot be executed.
Please avoid loops in your functions or actions that modify large areas of storage
(this includes clearing or copying arrays in storage)
Pos: 111:4:

Gas costs:

Gas requirement of function saturnaNFT.setApprovalForAll is infinite:
If the gas requirement of a function is higher than the block gas limit, it cannot be executed.
Please avoid loops in your functions or actions that modify large areas of storage
(this includes clearing or copying arrays in storage)
Pos: 135:4:

Gas costs:

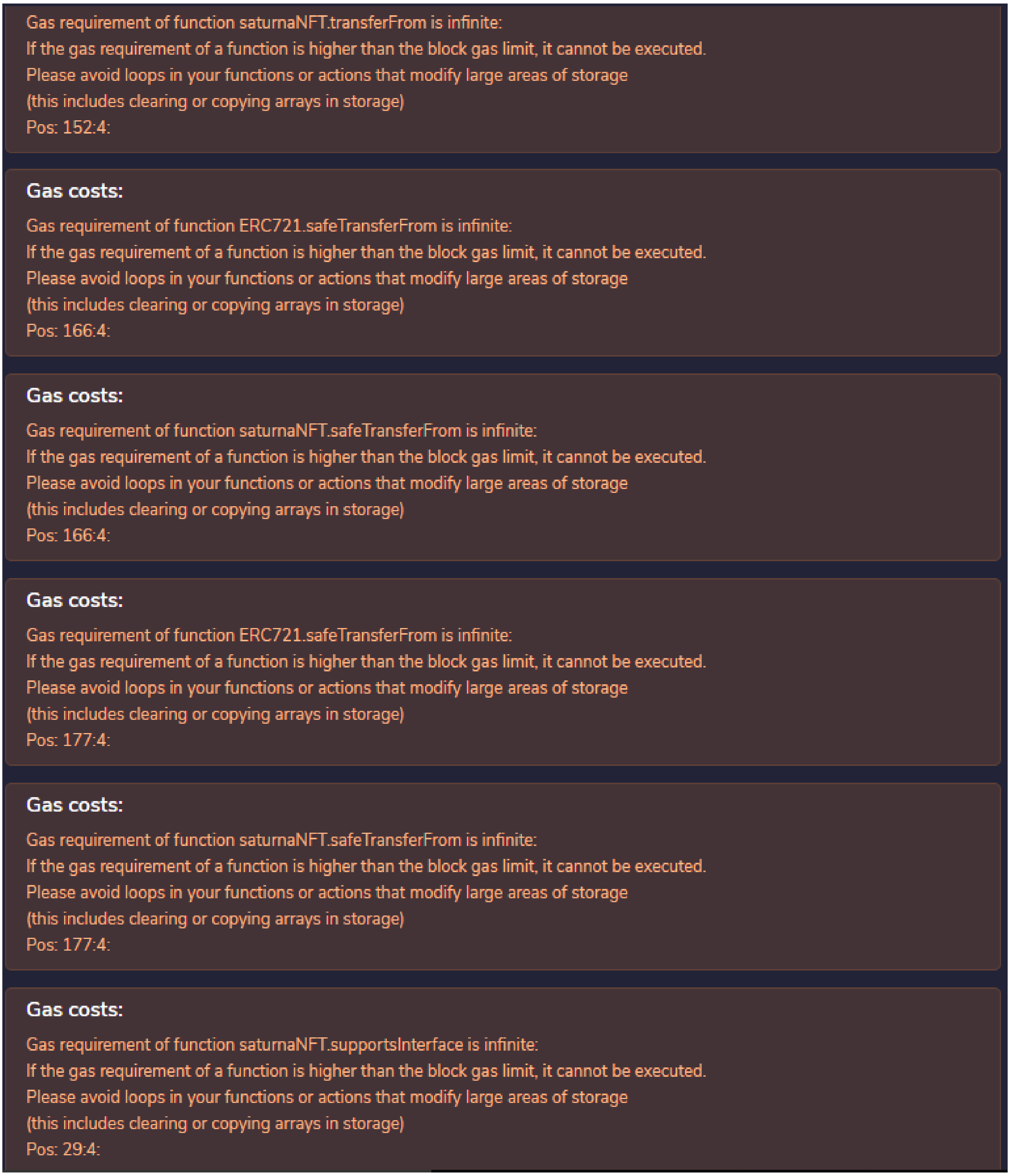
Gas requirement of function ERC721.isApprovedForAll is infinite:
If the gas requirement of a function is higher than the block gas limit, it cannot be executed.
Please avoid loops in your functions or actions that modify large areas of storage
(this includes clearing or copying arrays in storage)
Pos: 145:4:

Gas costs:

Gas requirement of function saturnaNFT.isApprovedForAll is infinite:
If the gas requirement of a function is higher than the block gas limit, it cannot be executed.
Please avoid loops in your functions or actions that modify large areas of storage
(this includes clearing or copying arrays in storage)
Pos: 145:4:

Gas costs:

Gas requirement of function ERC721.transferFrom is infinite:
If the gas requirement of a function is higher than the block gas limit, it cannot be executed.
Please avoid loops in your functions or actions that modify large areas of storage
(this includes clearing or copying arrays in storage)
Pos: 152:4:



Gas requirement of function saturnaNFT.transferFrom is infinite:
If the gas requirement of a function is higher than the block gas limit, it cannot be executed.
Please avoid loops in your functions or actions that modify large areas of storage
(this includes clearing or copying arrays in storage)
Pos: 152:4:

Gas costs:

Gas requirement of function ERC721.safeTransferFrom is infinite:
If the gas requirement of a function is higher than the block gas limit, it cannot be executed.
Please avoid loops in your functions or actions that modify large areas of storage
(this includes clearing or copying arrays in storage)
Pos: 166:4:

Gas costs:

Gas requirement of function saturnaNFT.safeTransferFrom is infinite:
If the gas requirement of a function is higher than the block gas limit, it cannot be executed.
Please avoid loops in your functions or actions that modify large areas of storage
(this includes clearing or copying arrays in storage)
Pos: 166:4:

Gas costs:

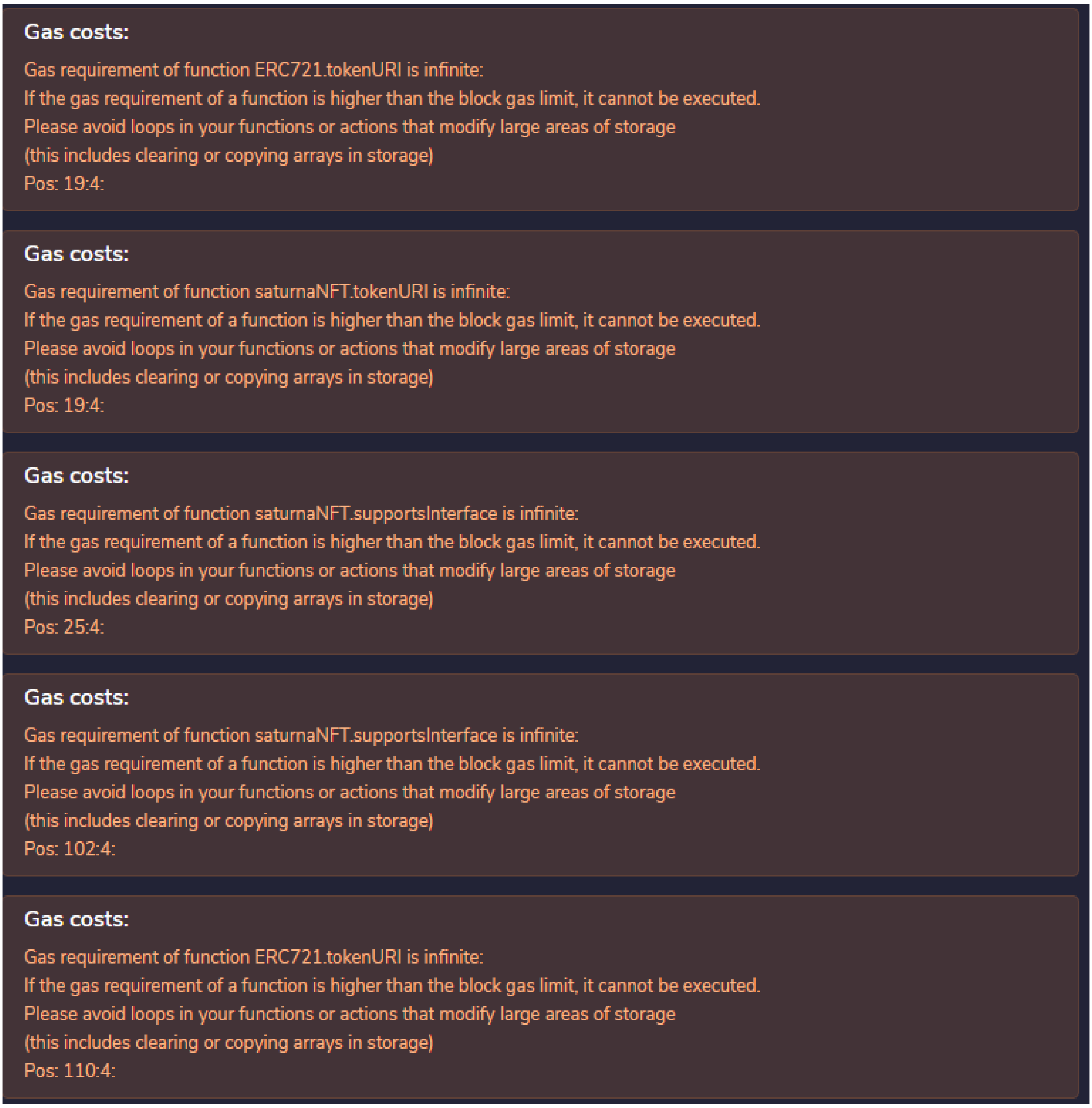
Gas requirement of function ERC721.safeTransferFrom is infinite:
If the gas requirement of a function is higher than the block gas limit, it cannot be executed.
Please avoid loops in your functions or actions that modify large areas of storage
(this includes clearing or copying arrays in storage)
Pos: 177:4:

Gas costs:

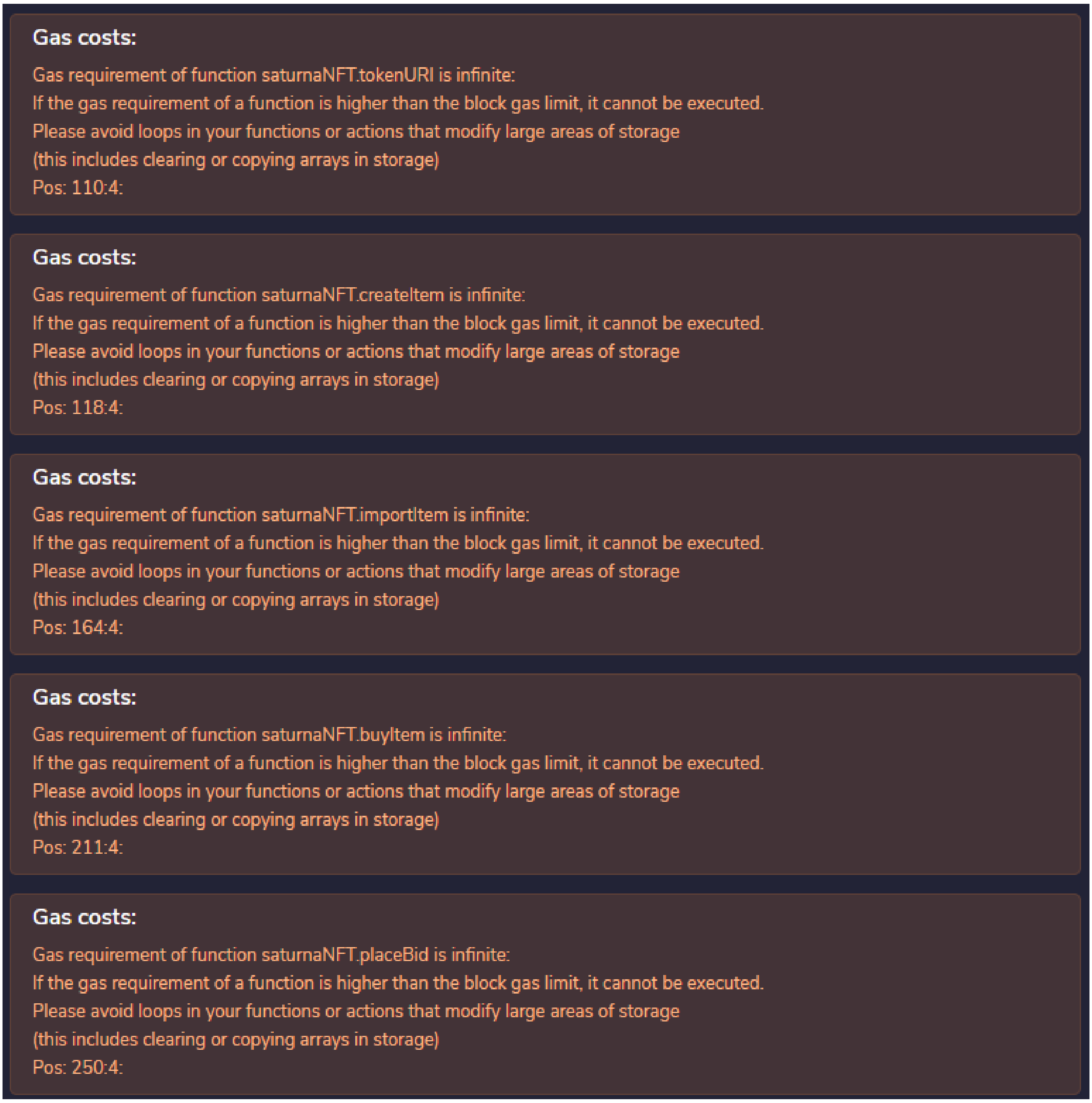
Gas requirement of function saturnaNFT.safeTransferFrom is infinite:
If the gas requirement of a function is higher than the block gas limit, it cannot be executed.
Please avoid loops in your functions or actions that modify large areas of storage
(this includes clearing or copying arrays in storage)
Pos: 177:4:

Gas costs:

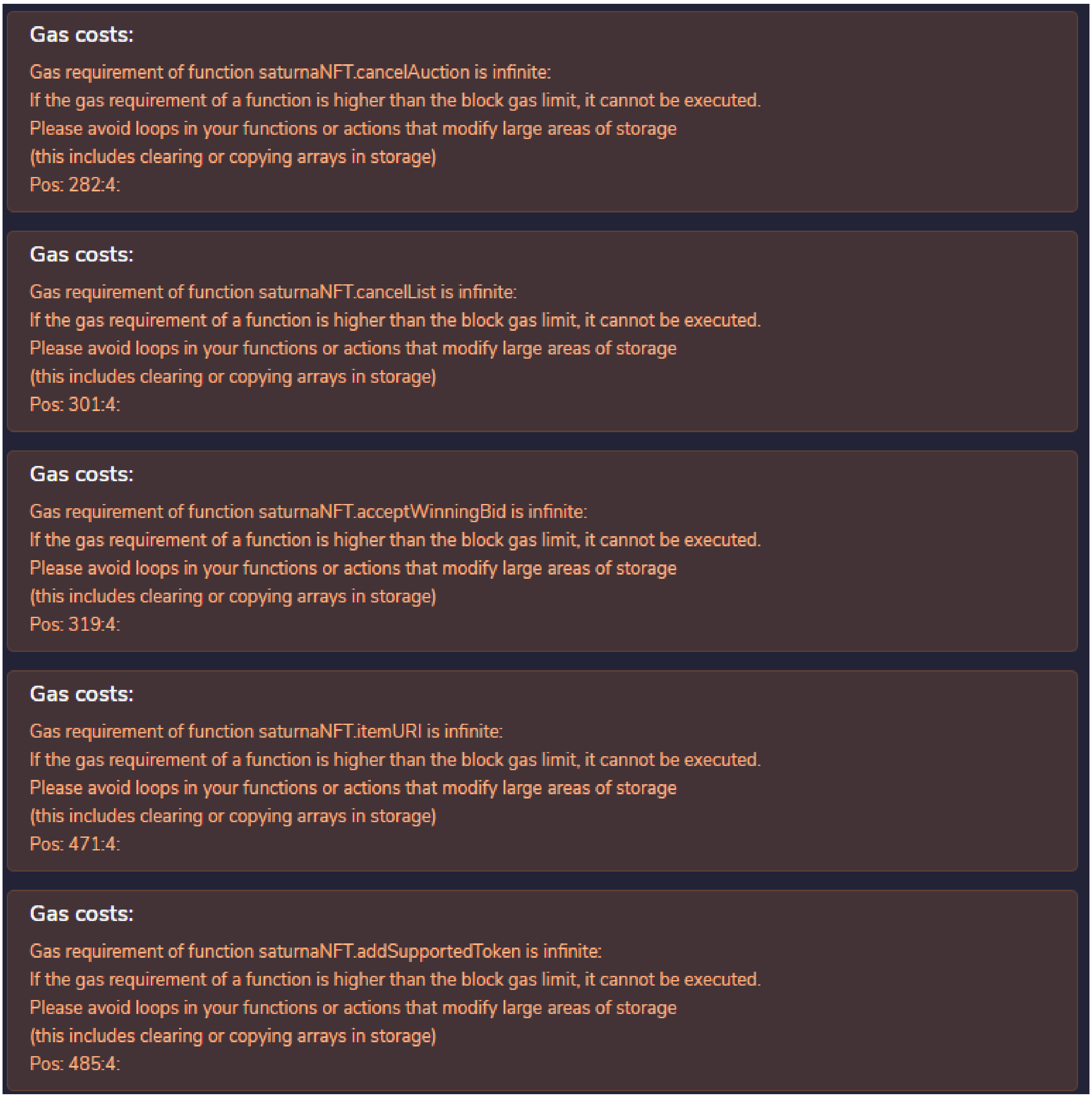
Gas requirement of function saturnaNFT.supportsInterface is infinite:
If the gas requirement of a function is higher than the block gas limit, it cannot be executed.
Please avoid loops in your functions or actions that modify large areas of storage
(this includes clearing or copying arrays in storage)
Pos: 29:4:



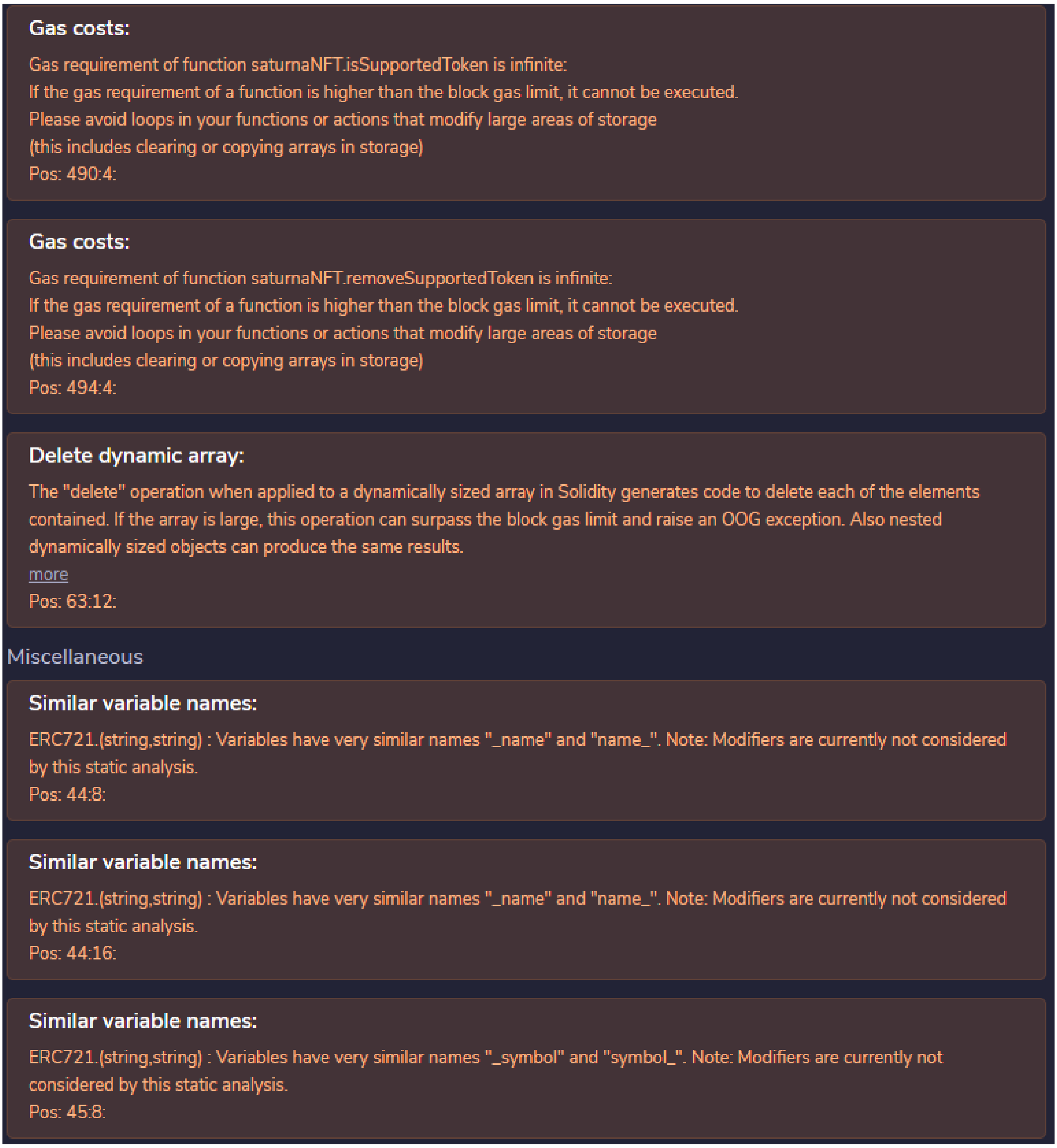
```
Gas costs:
Gas requirement of function ERC721.tokenURI is infinite:
If the gas requirement of a function is higher than the block gas limit, it cannot be executed.
Please avoid loops in your functions or actions that modify large areas of storage
(this includes clearing or copying arrays in storage)
Pos: 19:4:

Gas costs:
Gas requirement of function saturnaNFT.tokenURI is infinite:
If the gas requirement of a function is higher than the block gas limit, it cannot be executed.
Please avoid loops in your functions or actions that modify large areas of storage
(this includes clearing or copying arrays in storage)
Pos: 19:4:

Gas costs:
Gas requirement of function saturnaNFT.supportsInterface is infinite:
If the gas requirement of a function is higher than the block gas limit, it cannot be executed.
Please avoid loops in your functions or actions that modify large areas of storage
(this includes clearing or copying arrays in storage)
Pos: 25:4:

Gas costs:
Gas requirement of function saturnaNFT.supportsInterface is infinite:
If the gas requirement of a function is higher than the block gas limit, it cannot be executed.
Please avoid loops in your functions or actions that modify large areas of storage
(this includes clearing or copying arrays in storage)
Pos: 102:4:

Gas costs:
Gas requirement of function ERC721.tokenURI is infinite:
If the gas requirement of a function is higher than the block gas limit, it cannot be executed.
Please avoid loops in your functions or actions that modify large areas of storage
(this includes clearing or copying arrays in storage)
Pos: 110:4:
```



```
Gas costs:
Gas requirement of function saturnaNFT.tokenURI is infinite:
If the gas requirement of a function is higher than the block gas limit, it cannot be executed.
Please avoid loops in your functions or actions that modify large areas of storage
(this includes clearing or copying arrays in storage)
Pos: 110:4:

Gas costs:
Gas requirement of function saturnaNFT.createItem is infinite:
If the gas requirement of a function is higher than the block gas limit, it cannot be executed.
Please avoid loops in your functions or actions that modify large areas of storage
(this includes clearing or copying arrays in storage)
Pos: 118:4:

Gas costs:
Gas requirement of function saturnaNFT.importItem is infinite:
If the gas requirement of a function is higher than the block gas limit, it cannot be executed.
Please avoid loops in your functions or actions that modify large areas of storage
(this includes clearing or copying arrays in storage)
Pos: 164:4:

Gas costs:
Gas requirement of function saturnaNFT.buyItem is infinite:
If the gas requirement of a function is higher than the block gas limit, it cannot be executed.
Please avoid loops in your functions or actions that modify large areas of storage
(this includes clearing or copying arrays in storage)
Pos: 211:4:

Gas costs:
Gas requirement of function saturnaNFT.placeBid is infinite:
If the gas requirement of a function is higher than the block gas limit, it cannot be executed.
Please avoid loops in your functions or actions that modify large areas of storage
(this includes clearing or copying arrays in storage)
Pos: 250:4:
```



```
Gas costs:
Gas requirement of function saturnaNFT.cancelAuction is infinite:
If the gas requirement of a function is higher than the block gas limit, it cannot be executed.
Please avoid loops in your functions or actions that modify large areas of storage
(this includes clearing or copying arrays in storage)
Pos: 282:4:

Gas costs:
Gas requirement of function saturnaNFT.cancelList is infinite:
If the gas requirement of a function is higher than the block gas limit, it cannot be executed.
Please avoid loops in your functions or actions that modify large areas of storage
(this includes clearing or copying arrays in storage)
Pos: 301:4:

Gas costs:
Gas requirement of function saturnaNFT.acceptWinningBid is infinite:
If the gas requirement of a function is higher than the block gas limit, it cannot be executed.
Please avoid loops in your functions or actions that modify large areas of storage
(this includes clearing or copying arrays in storage)
Pos: 319:4:

Gas costs:
Gas requirement of function saturnaNFT.itemURI is infinite:
If the gas requirement of a function is higher than the block gas limit, it cannot be executed.
Please avoid loops in your functions or actions that modify large areas of storage
(this includes clearing or copying arrays in storage)
Pos: 471:4:

Gas costs:
Gas requirement of function saturnaNFT.addSupportedToken is infinite:
If the gas requirement of a function is higher than the block gas limit, it cannot be executed.
Please avoid loops in your functions or actions that modify large areas of storage
(this includes clearing or copying arrays in storage)
Pos: 485:4:
```



Gas costs:

Gas requirement of function saturnaNFT.isSupportedToken is infinite:
If the gas requirement of a function is higher than the block gas limit, it cannot be executed.
Please avoid loops in your functions or actions that modify large areas of storage
(this includes clearing or copying arrays in storage)
Pos: 490:4:

Gas costs:

Gas requirement of function saturnaNFT.removeSupportedToken is infinite:
If the gas requirement of a function is higher than the block gas limit, it cannot be executed.
Please avoid loops in your functions or actions that modify large areas of storage
(this includes clearing or copying arrays in storage)
Pos: 494:4:

Delete dynamic array:

The "delete" operation when applied to a dynamically sized array in Solidity generates code to delete each of the elements contained. If the array is large, this operation can surpass the block gas limit and raise an OOG exception. Also nested dynamically sized objects can produce the same results.
more
Pos: 63:12:

Miscellaneous

Similar variable names:

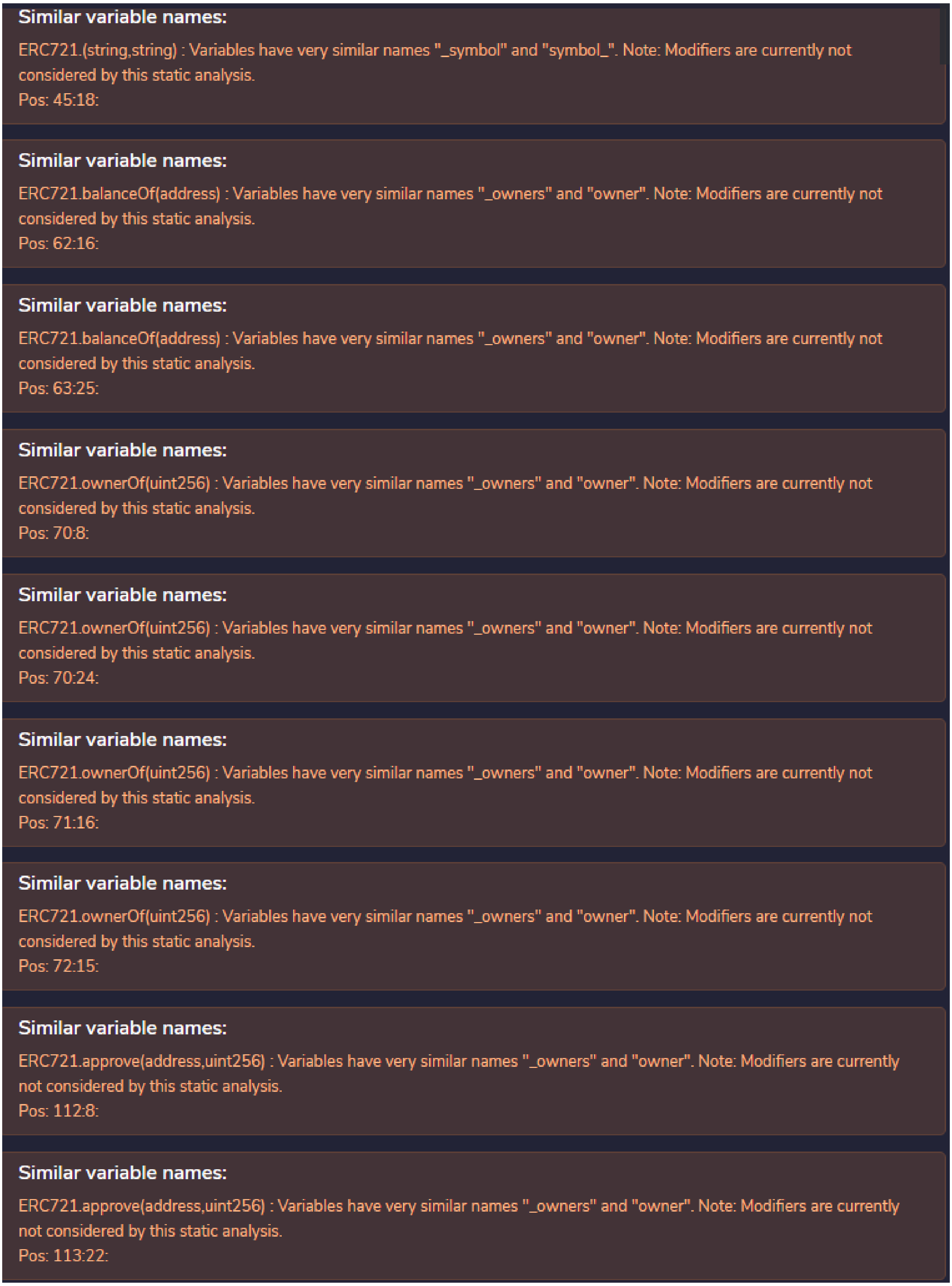
ERC721.(string,string) : Variables have very similar names "_name" and "name_". Note: Modifiers are currently not considered by this static analysis.
Pos: 44:8:

Similar variable names:

ERC721.(string,string) : Variables have very similar names "_name" and "name_". Note: Modifiers are currently not considered by this static analysis.
Pos: 44:16:

Similar variable names:

ERC721.(string,string) : Variables have very similar names "_symbol" and "symbol_". Note: Modifiers are currently not considered by this static analysis.
Pos: 45:8:



Similar variable names:

ERC721.(string,string) : Variables have very similar names "_symbol" and "symbol_". Note: Modifiers are currently not considered by this static analysis.
Pos: 45:18:

Similar variable names:

ERC721.balanceOf(address) : Variables have very similar names "_owners" and "owner". Note: Modifiers are currently not considered by this static analysis.
Pos: 62:16:

Similar variable names:

ERC721.balanceOf(address) : Variables have very similar names "_owners" and "owner". Note: Modifiers are currently not considered by this static analysis.
Pos: 63:25:

Similar variable names:

ERC721.ownerOf(uint256) : Variables have very similar names "_owners" and "owner". Note: Modifiers are currently not considered by this static analysis.
Pos: 70:8:

Similar variable names:

ERC721.ownerOf(uint256) : Variables have very similar names "_owners" and "owner". Note: Modifiers are currently not considered by this static analysis.
Pos: 70:24:

Similar variable names:

ERC721.ownerOf(uint256) : Variables have very similar names "_owners" and "owner". Note: Modifiers are currently not considered by this static analysis.
Pos: 71:16:

Similar variable names:

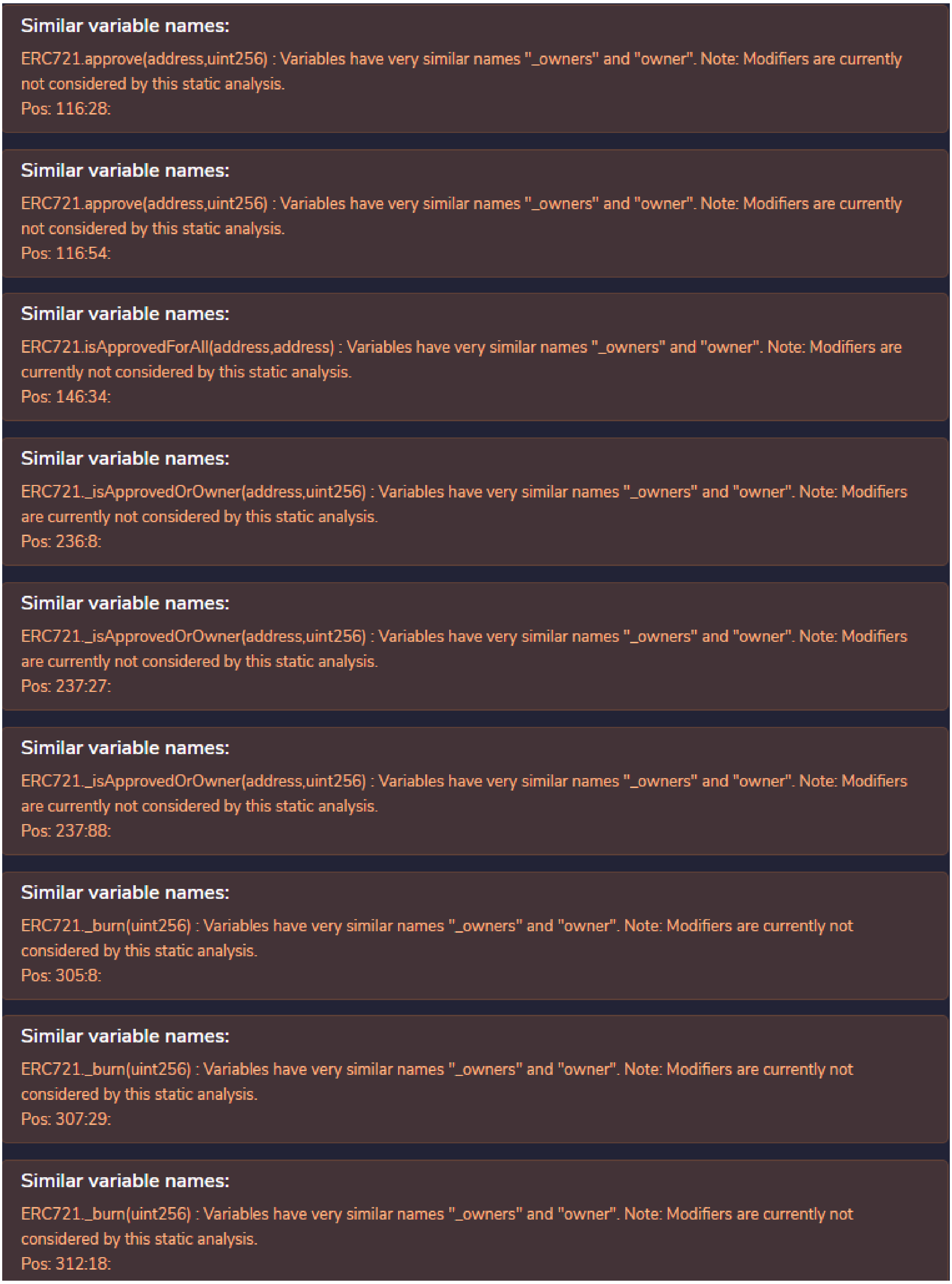
ERC721.ownerOf(uint256) : Variables have very similar names "_owners" and "owner". Note: Modifiers are currently not considered by this static analysis.
Pos: 72:15:

Similar variable names:

ERC721.approve(address,uint256) : Variables have very similar names "_owners" and "owner". Note: Modifiers are currently not considered by this static analysis.
Pos: 112:8:

Similar variable names:

ERC721.approve(address,uint256) : Variables have very similar names "_owners" and "owner". Note: Modifiers are currently not considered by this static analysis.
Pos: 113:22:



Similar variable names:

ERC721.approve(address,uint256) : Variables have very similar names "_owners" and "owner". Note: Modifiers are currently not considered by this static analysis.
Pos: 116:28:

Similar variable names:

ERC721.approve(address,uint256) : Variables have very similar names "_owners" and "owner". Note: Modifiers are currently not considered by this static analysis.
Pos: 116:54:

Similar variable names:

ERC721.isApprovedForAll(address,address) : Variables have very similar names "_owners" and "owner". Note: Modifiers are currently not considered by this static analysis.
Pos: 146:34:

Similar variable names:

ERC721._isApprovedOrOwner(address,uint256) : Variables have very similar names "_owners" and "owner". Note: Modifiers are currently not considered by this static analysis.
Pos: 236:8:

Similar variable names:

ERC721._isApprovedOrOwner(address,uint256) : Variables have very similar names "_owners" and "owner". Note: Modifiers are currently not considered by this static analysis.
Pos: 237:27:

Similar variable names:

ERC721._isApprovedOrOwner(address,uint256) : Variables have very similar names "_owners" and "owner". Note: Modifiers are currently not considered by this static analysis.
Pos: 237:88:

Similar variable names:

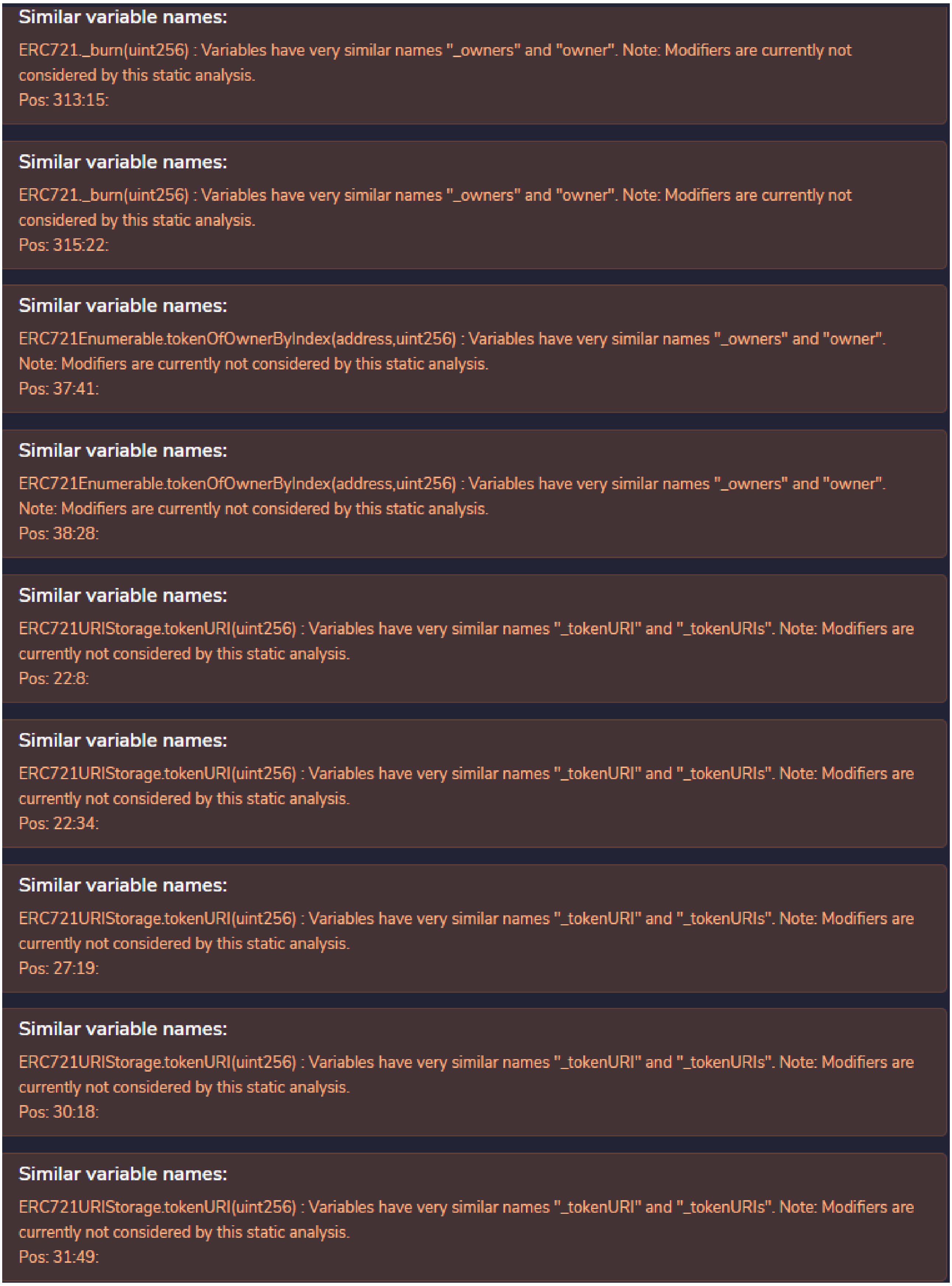
ERC721._burn(uint256) : Variables have very similar names "_owners" and "owner". Note: Modifiers are currently not considered by this static analysis.
Pos: 305:8:

Similar variable names:

ERC721._burn(uint256) : Variables have very similar names "_owners" and "owner". Note: Modifiers are currently not considered by this static analysis.
Pos: 307:29:

Similar variable names:

ERC721._burn(uint256) : Variables have very similar names "_owners" and "owner". Note: Modifiers are currently not considered by this static analysis.
Pos: 312:18:



Similar variable names:

ERC721._burn(uint256) : Variables have very similar names "_owners" and "owner". Note: Modifiers are currently not considered by this static analysis.
Pos: 313:15:

Similar variable names:

ERC721._burn(uint256) : Variables have very similar names "_owners" and "owner". Note: Modifiers are currently not considered by this static analysis.
Pos: 315:22:

Similar variable names:

ERC721Enumerable.tokenOfOwnerByIndex(address,uint256) : Variables have very similar names "_owners" and "owner". Note: Modifiers are currently not considered by this static analysis.
Pos: 37:41:

Similar variable names:

ERC721Enumerable.tokenOfOwnerByIndex(address,uint256) : Variables have very similar names "_owners" and "owner". Note: Modifiers are currently not considered by this static analysis.
Pos: 38:28:

Similar variable names:

ERC721URIStorage.tokenURI(uint256) : Variables have very similar names "_tokenURI" and "_tokenURIs". Note: Modifiers are currently not considered by this static analysis.
Pos: 22:8:

Similar variable names:

ERC721URIStorage.tokenURI(uint256) : Variables have very similar names "_tokenURI" and "_tokenURIs". Note: Modifiers are currently not considered by this static analysis.
Pos: 22:34:

Similar variable names:

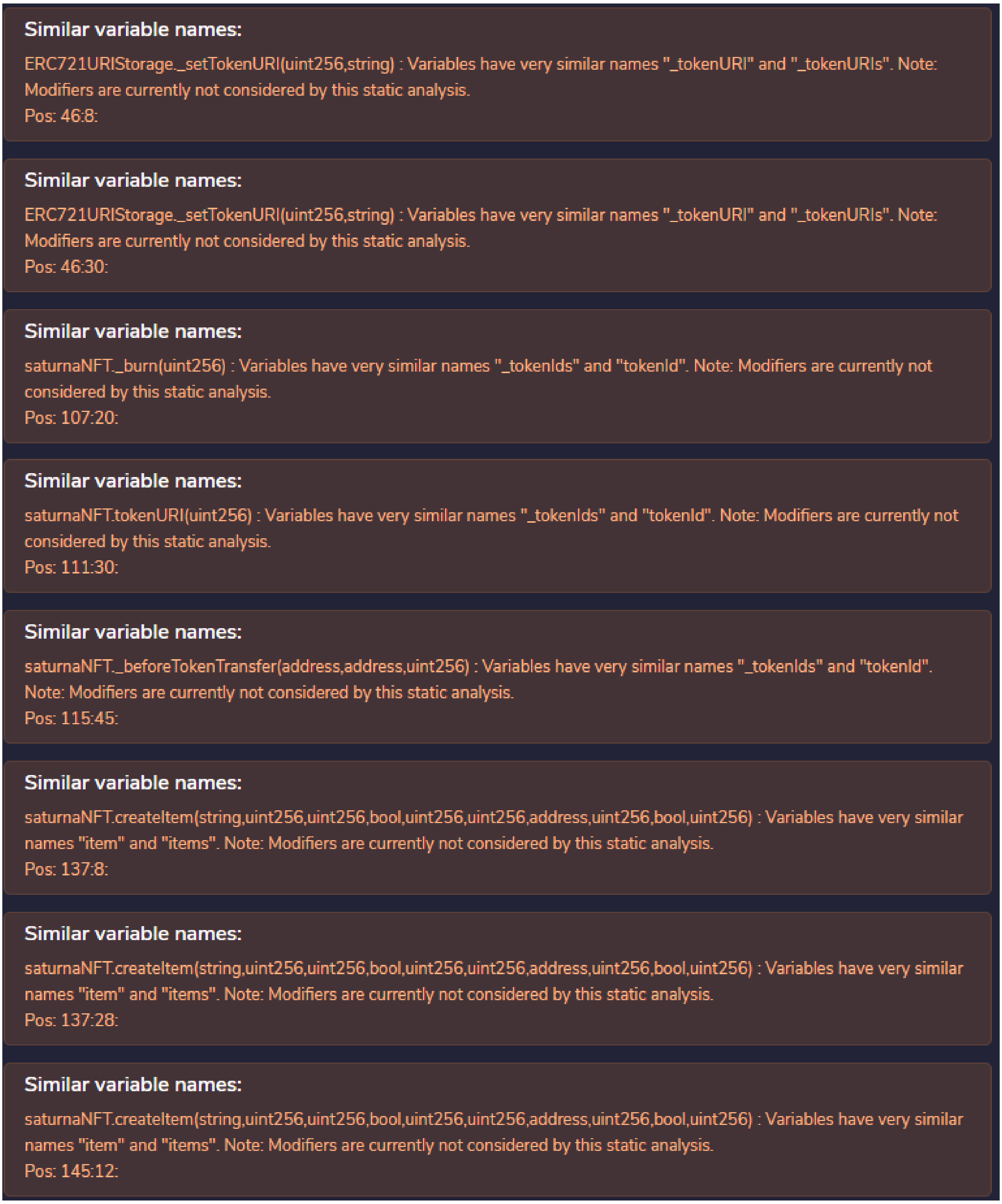
ERC721URIStorage.tokenURI(uint256) : Variables have very similar names "_tokenURI" and "_tokenURIs". Note: Modifiers are currently not considered by this static analysis.
Pos: 27:19:

Similar variable names:

ERC721URIStorage.tokenURI(uint256) : Variables have very similar names "_tokenURI" and "_tokenURIs". Note: Modifiers are currently not considered by this static analysis.
Pos: 30:18:

Similar variable names:

ERC721URIStorage.tokenURI(uint256) : Variables have very similar names "_tokenURI" and "_tokenURIs". Note: Modifiers are currently not considered by this static analysis.
Pos: 31:49:



Similar variable names:

ERC721URIStorage._setTokenURI(uint256,string) : Variables have very similar names "_tokenURI" and "_tokenURIs". Note: Modifiers are currently not considered by this static analysis.
Pos: 46:8:

Similar variable names:

ERC721URIStorage._setTokenURI(uint256,string) : Variables have very similar names "_tokenURI" and "_tokenURIs". Note: Modifiers are currently not considered by this static analysis.
Pos: 46:30:

Similar variable names:

saturnaNFT._burn(uint256) : Variables have very similar names "_tokenIds" and "tokenId". Note: Modifiers are currently not considered by this static analysis.
Pos: 107:20:

Similar variable names:

saturnaNFT.tokenURI(uint256) : Variables have very similar names "_tokenIds" and "tokenId". Note: Modifiers are currently not considered by this static analysis.
Pos: 111:30:

Similar variable names:

saturnaNFT._beforeTokenTransfer(address,address,uint256) : Variables have very similar names "_tokenIds" and "tokenId". Note: Modifiers are currently not considered by this static analysis.
Pos: 115:45:

Similar variable names:

saturnaNFT.createItem(string,uint256,uint256,bool,uint256,uint256,address,uint256,bool,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 137:8:

Similar variable names:

saturnaNFT.createItem(string,uint256,uint256,bool,uint256,uint256,address,uint256,bool,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 137:28:

Similar variable names:

saturnaNFT.createItem(string,uint256,uint256,bool,uint256,uint256,address,uint256,bool,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 145:12:

Similar variable names:

saturnaNFT.createItem(string,uint256,uint256,bool,uint256,uint256,address,uint256,bool,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 146:12:

Similar variable names:

saturnaNFT.createItem(string,uint256,uint256,bool,uint256,uint256,address,uint256,bool,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 147:12:

Similar variable names:

saturnaNFT.createItem(string,uint256,uint256,bool,uint256,uint256,address,uint256,bool,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 148:12:

Similar variable names:

saturnaNFT.createItem(string,uint256,uint256,bool,uint256,uint256,address,uint256,bool,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 149:12:

Similar variable names:

saturnaNFT.createItem(string,uint256,uint256,bool,uint256,uint256,address,uint256,bool,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 150:12:

Similar variable names:

saturnaNFT.createItem(string,uint256,uint256,bool,uint256,uint256,address,uint256,bool,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 151:12:

Similar variable names:

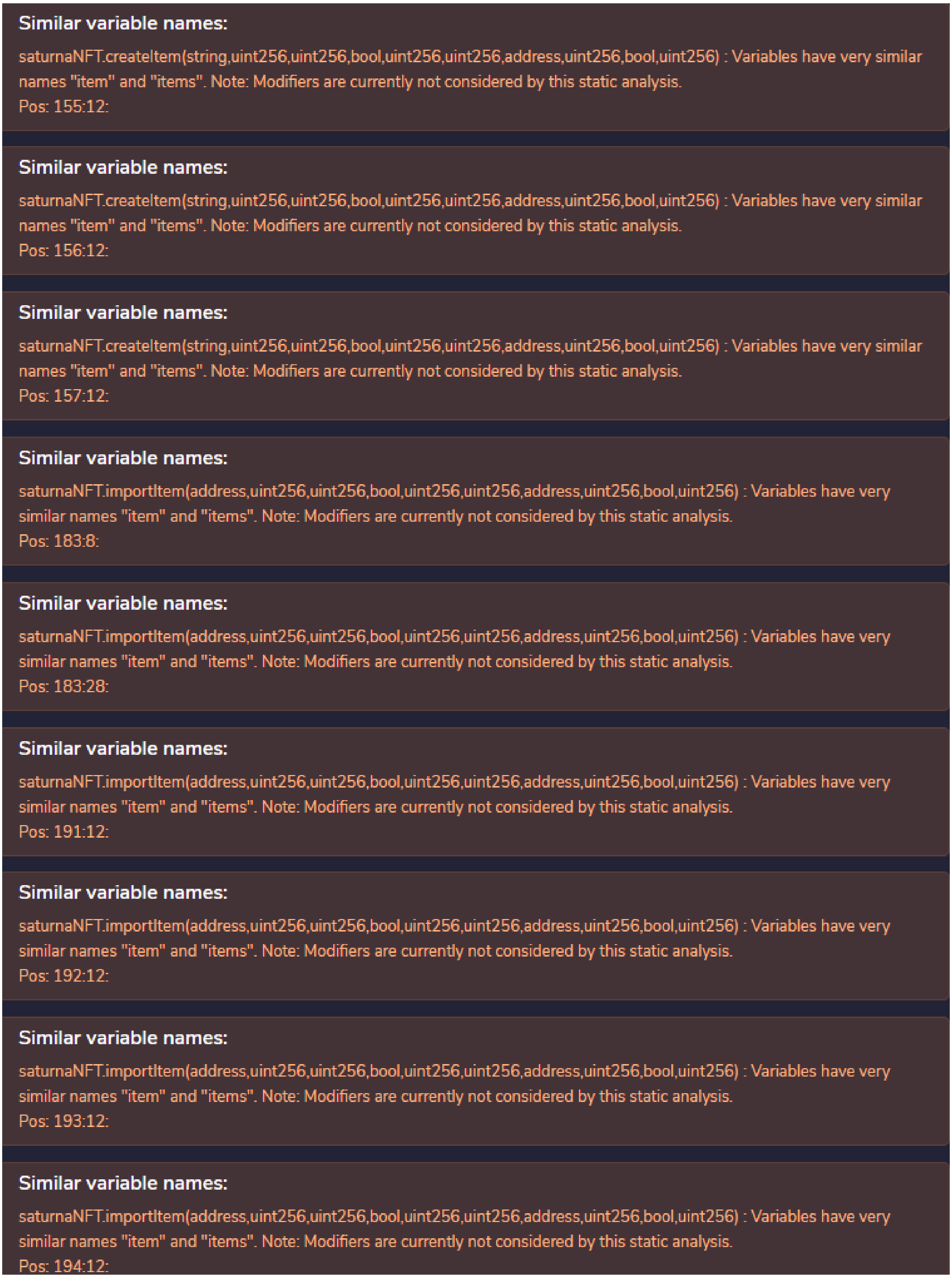
saturnaNFT.createItem(string,uint256,uint256,bool,uint256,uint256,address,uint256,bool,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 152:12:

Similar variable names:

saturnaNFT.createItem(string,uint256,uint256,bool,uint256,uint256,address,uint256,bool,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 153:12:

Similar variable names:

saturnaNFT.createItem(string,uint256,uint256,bool,uint256,uint256,address,uint256,bool,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 154:12:



Similar variable names:

saturnaNFT.createItem(string,uint256,uint256,bool,uint256,uint256,address,uint256,bool,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 155:12:

Similar variable names:

saturnaNFT.createItem(string,uint256,uint256,bool,uint256,uint256,address,uint256,bool,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 156:12:

Similar variable names:

saturnaNFT.createItem(string,uint256,uint256,bool,uint256,uint256,address,uint256,bool,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 157:12:

Similar variable names:

saturnaNFT.importItem(address,uint256,uint256,bool,uint256,uint256,address,uint256,bool,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 183:8:

Similar variable names:

saturnaNFT.importItem(address,uint256,uint256,bool,uint256,uint256,address,uint256,bool,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 183:28:

Similar variable names:

saturnaNFT.importItem(address,uint256,uint256,bool,uint256,uint256,address,uint256,bool,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 191:12:

Similar variable names:

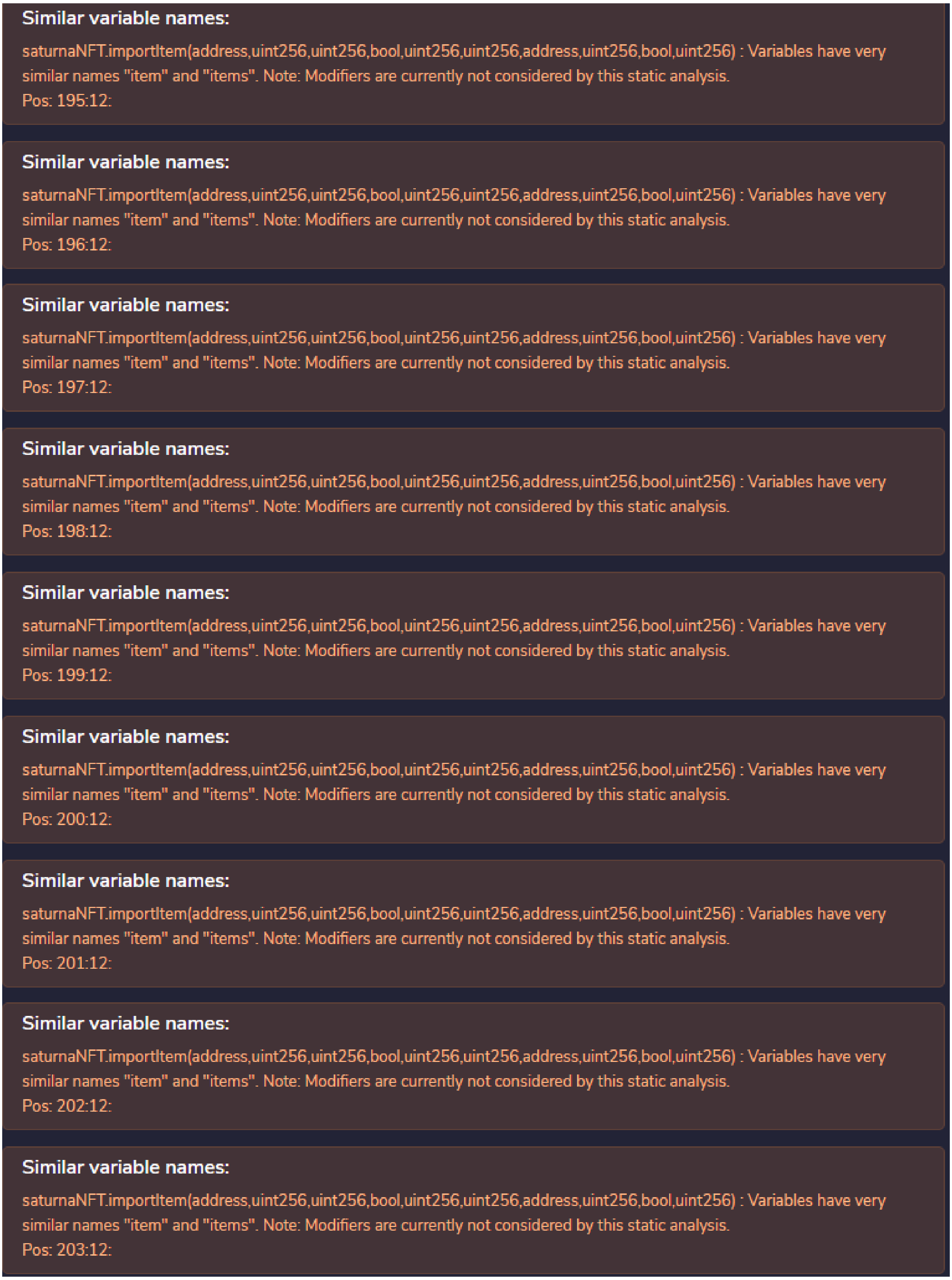
saturnaNFT.importItem(address,uint256,uint256,bool,uint256,uint256,address,uint256,bool,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 192:12:

Similar variable names:

saturnaNFT.importItem(address,uint256,uint256,bool,uint256,uint256,address,uint256,bool,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 193:12:

Similar variable names:

saturnaNFT.importItem(address,uint256,uint256,bool,uint256,uint256,address,uint256,bool,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 194:12:



Similar variable names:

saturnaNFT.importItem(address,uint256,uint256,bool,uint256,uint256,address,uint256,bool,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 195:12:

Similar variable names:

saturnaNFT.importItem(address,uint256,uint256,bool,uint256,uint256,address,uint256,bool,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 196:12:

Similar variable names:

saturnaNFT.importItem(address,uint256,uint256,bool,uint256,uint256,address,uint256,bool,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 197:12:

Similar variable names:

saturnaNFT.importItem(address,uint256,uint256,bool,uint256,uint256,address,uint256,bool,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 198:12:

Similar variable names:

saturnaNFT.importItem(address,uint256,uint256,bool,uint256,uint256,address,uint256,bool,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 199:12:

Similar variable names:

saturnaNFT.importItem(address,uint256,uint256,bool,uint256,uint256,address,uint256,bool,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 200:12:

Similar variable names:

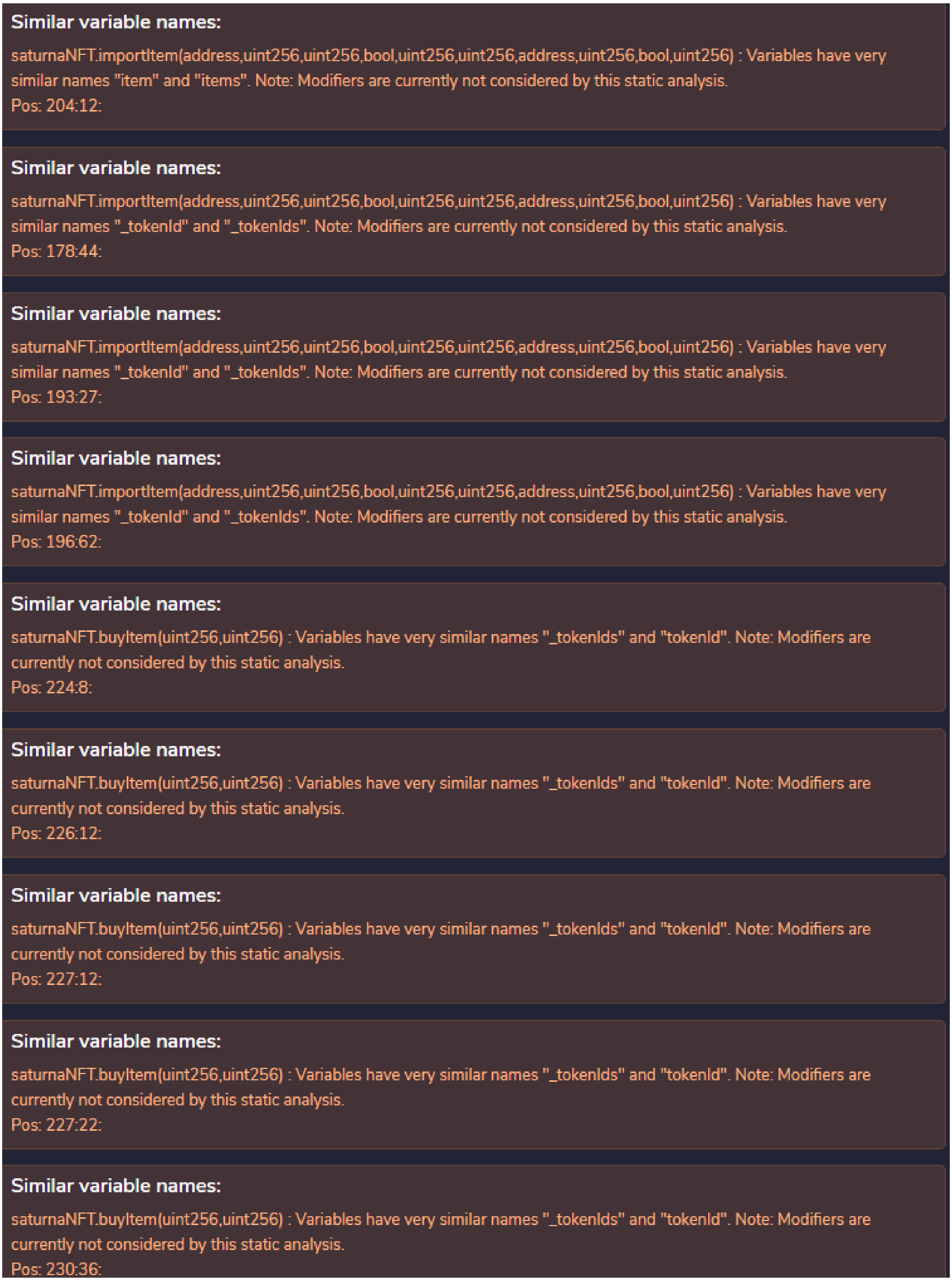
saturnaNFT.importItem(address,uint256,uint256,bool,uint256,uint256,address,uint256,bool,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 201:12:

Similar variable names:

saturnaNFT.importItem(address,uint256,uint256,bool,uint256,uint256,address,uint256,bool,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 202:12:

Similar variable names:

saturnaNFT.importItem(address,uint256,uint256,bool,uint256,uint256,address,uint256,bool,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 203:12:



Similar variable names:

saturnaNFT.importItem(address,uint256,uint256,bool,uint256,uint256,address,uint256,bool,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 204:12:

Similar variable names:

saturnaNFT.importItem(address,uint256,uint256,bool,uint256,uint256,address,uint256,bool,uint256) : Variables have very similar names "_tokenId" and "_tokenIds". Note: Modifiers are currently not considered by this static analysis.
Pos: 178:44:

Similar variable names:

saturnaNFT.importItem(address,uint256,uint256,bool,uint256,uint256,address,uint256,bool,uint256) : Variables have very similar names "_tokenId" and "_tokenIds". Note: Modifiers are currently not considered by this static analysis.
Pos: 193:27:

Similar variable names:

saturnaNFT.importItem(address,uint256,uint256,bool,uint256,uint256,address,uint256,bool,uint256) : Variables have very similar names "_tokenId" and "_tokenIds". Note: Modifiers are currently not considered by this static analysis.
Pos: 196:62:

Similar variable names:

saturnaNFT.buyItem(uint256,uint256) : Variables have very similar names "_tokenIds" and "tokenId". Note: Modifiers are currently not considered by this static analysis.
Pos: 224:8:

Similar variable names:

saturnaNFT.buyItem(uint256,uint256) : Variables have very similar names "_tokenIds" and "tokenId". Note: Modifiers are currently not considered by this static analysis.
Pos: 226:12:

Similar variable names:

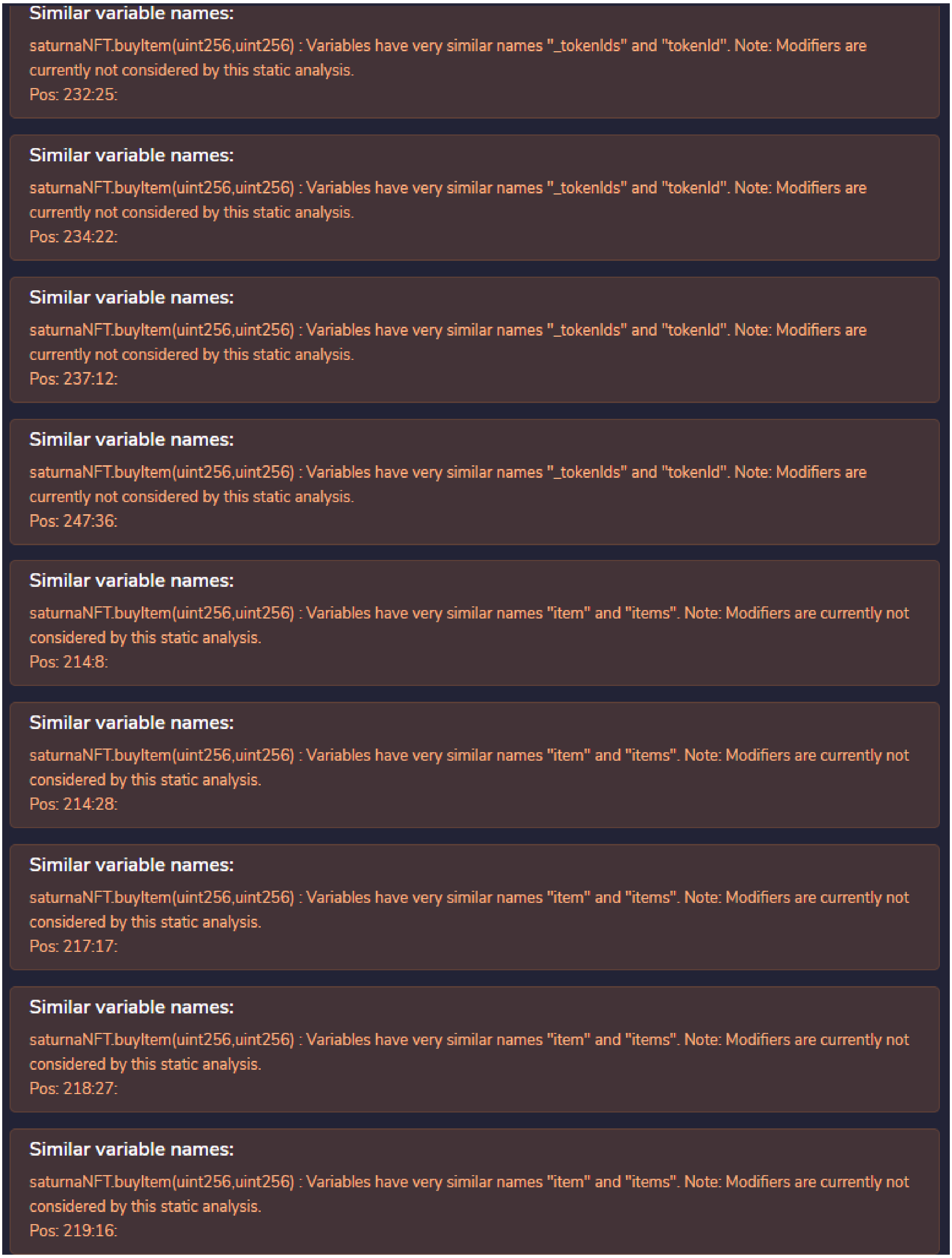
saturnaNFT.buyItem(uint256,uint256) : Variables have very similar names "_tokenIds" and "tokenId". Note: Modifiers are currently not considered by this static analysis.
Pos: 227:12:

Similar variable names:

saturnaNFT.buyItem(uint256,uint256) : Variables have very similar names "_tokenIds" and "tokenId". Note: Modifiers are currently not considered by this static analysis.
Pos: 227:22:

Similar variable names:

saturnaNFT.buyItem(uint256,uint256) : Variables have very similar names "_tokenIds" and "tokenId". Note: Modifiers are currently not considered by this static analysis.
Pos: 230:36:



Similar variable names:

saturnaNFT.buyItem(uint256,uint256) : Variables have very similar names "_tokenIds" and "tokenId". Note: Modifiers are currently not considered by this static analysis.
Pos: 232:25:

Similar variable names:

saturnaNFT.buyItem(uint256,uint256) : Variables have very similar names "_tokenIds" and "tokenId". Note: Modifiers are currently not considered by this static analysis.
Pos: 234:22:

Similar variable names:

saturnaNFT.buyItem(uint256,uint256) : Variables have very similar names "_tokenIds" and "tokenId". Note: Modifiers are currently not considered by this static analysis.
Pos: 237:12:

Similar variable names:

saturnaNFT.buyItem(uint256,uint256) : Variables have very similar names "_tokenIds" and "tokenId". Note: Modifiers are currently not considered by this static analysis.
Pos: 247:36:

Similar variable names:

saturnaNFT.buyItem(uint256,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 214:8:

Similar variable names:

saturnaNFT.buyItem(uint256,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 214:28:

Similar variable names:

saturnaNFT.buyItem(uint256,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 217:17:

Similar variable names:

saturnaNFT.buyItem(uint256,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 218:27:

Similar variable names:

saturnaNFT.buyItem(uint256,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 219:16:

Similar variable names:

saturnaNFT.buyItem(uint256,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 222:43:

Similar variable names:

saturnaNFT.buyItem(uint256,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 225:11:

Similar variable names:

saturnaNFT.buyItem(uint256,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 232:34:

Similar variable names:

saturnaNFT.buyItem(uint256,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 237:22:

Similar variable names:

saturnaNFT.buyItem(uint256,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 238:20:

Similar variable names:

saturnaNFT.buyItem(uint256,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 238:83:

Similar variable names:

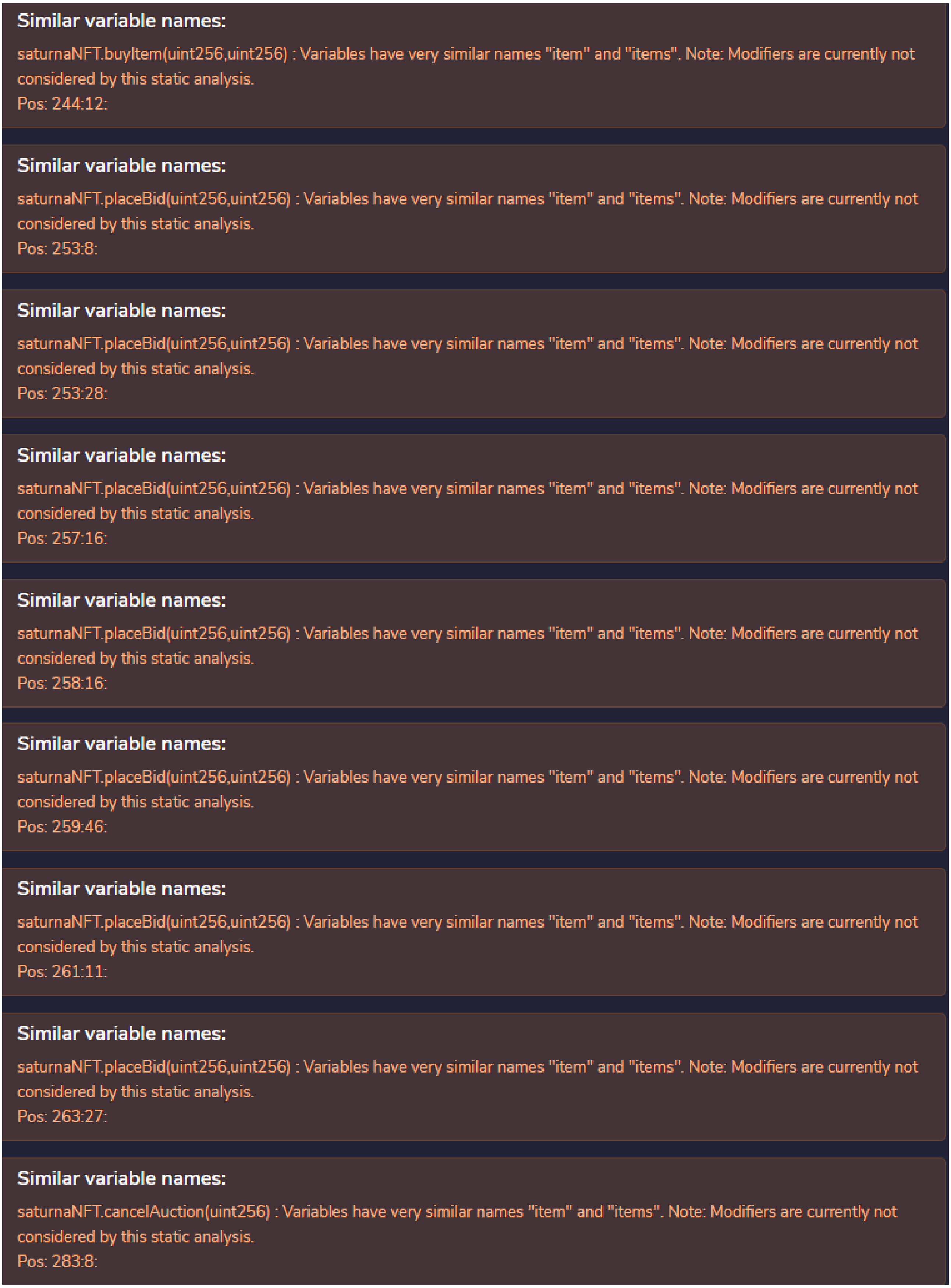
saturnaNFT.buyItem(uint256,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 242:8:

Similar variable names:

saturnaNFT.buyItem(uint256,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 242:21:

Similar variable names:

saturnaNFT.buyItem(uint256,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 243:11:



Similar variable names:

saturnaNFT.buyItem(uint256,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 244:12:

Similar variable names:

saturnaNFT.placeBid(uint256,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 253:8:

Similar variable names:

saturnaNFT.placeBid(uint256,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 253:28:

Similar variable names:

saturnaNFT.placeBid(uint256,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 257:16:

Similar variable names:

saturnaNFT.placeBid(uint256,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 258:16:

Similar variable names:

saturnaNFT.placeBid(uint256,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 259:46:

Similar variable names:

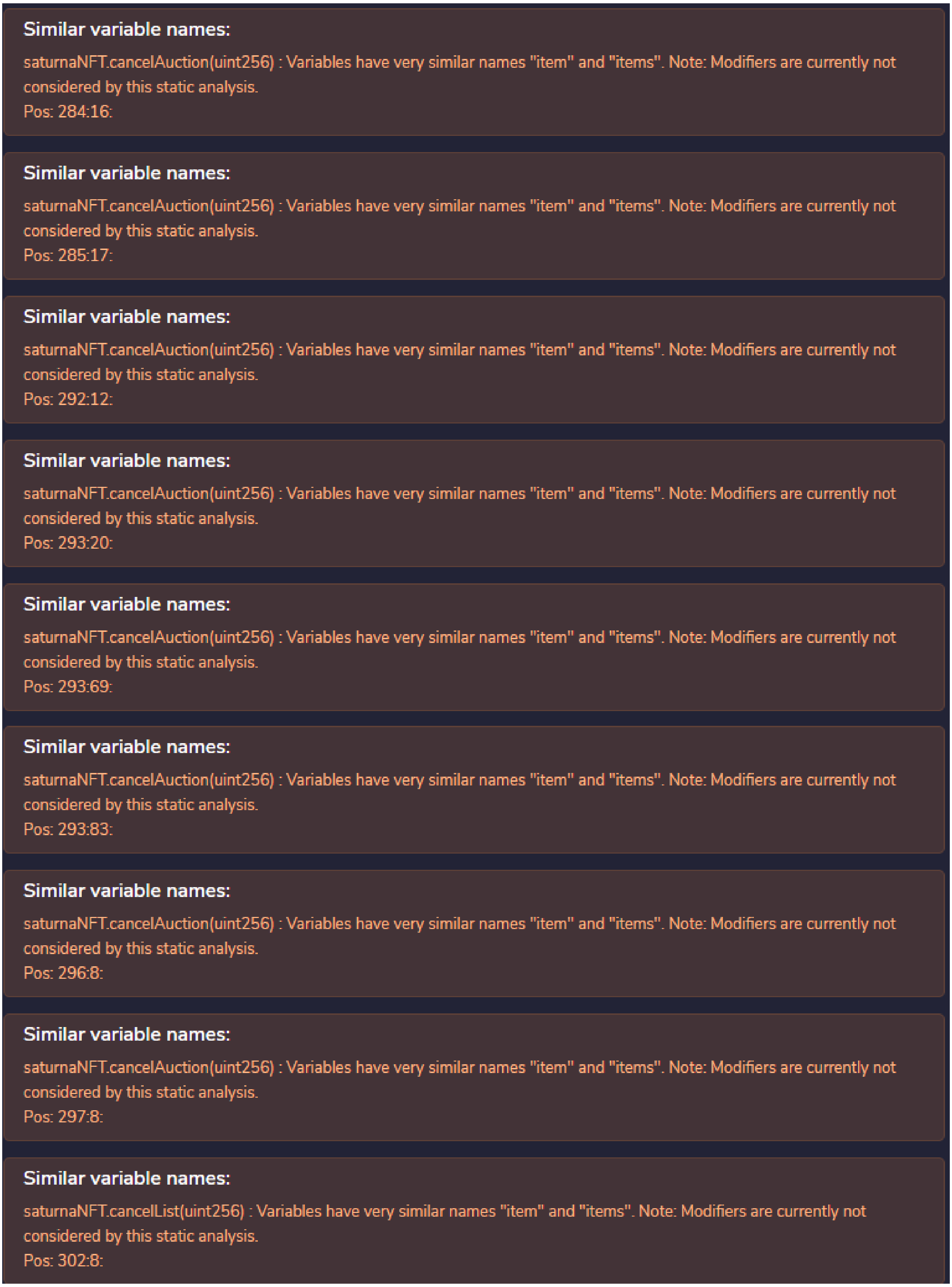
saturnaNFT.placeBid(uint256,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 261:11:

Similar variable names:

saturnaNFT.placeBid(uint256,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 263:27:

Similar variable names:

saturnaNFT.cancelAuction(uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 283:8:



Similar variable names:

saturnaNFT.cancelAuction(uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 284:16:

Similar variable names:

saturnaNFT.cancelAuction(uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 285:17:

Similar variable names:

saturnaNFT.cancelAuction(uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 292:12:

Similar variable names:

saturnaNFT.cancelAuction(uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 293:20:

Similar variable names:

saturnaNFT.cancelAuction(uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 293:69:

Similar variable names:

saturnaNFT.cancelAuction(uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 293:83:

Similar variable names:

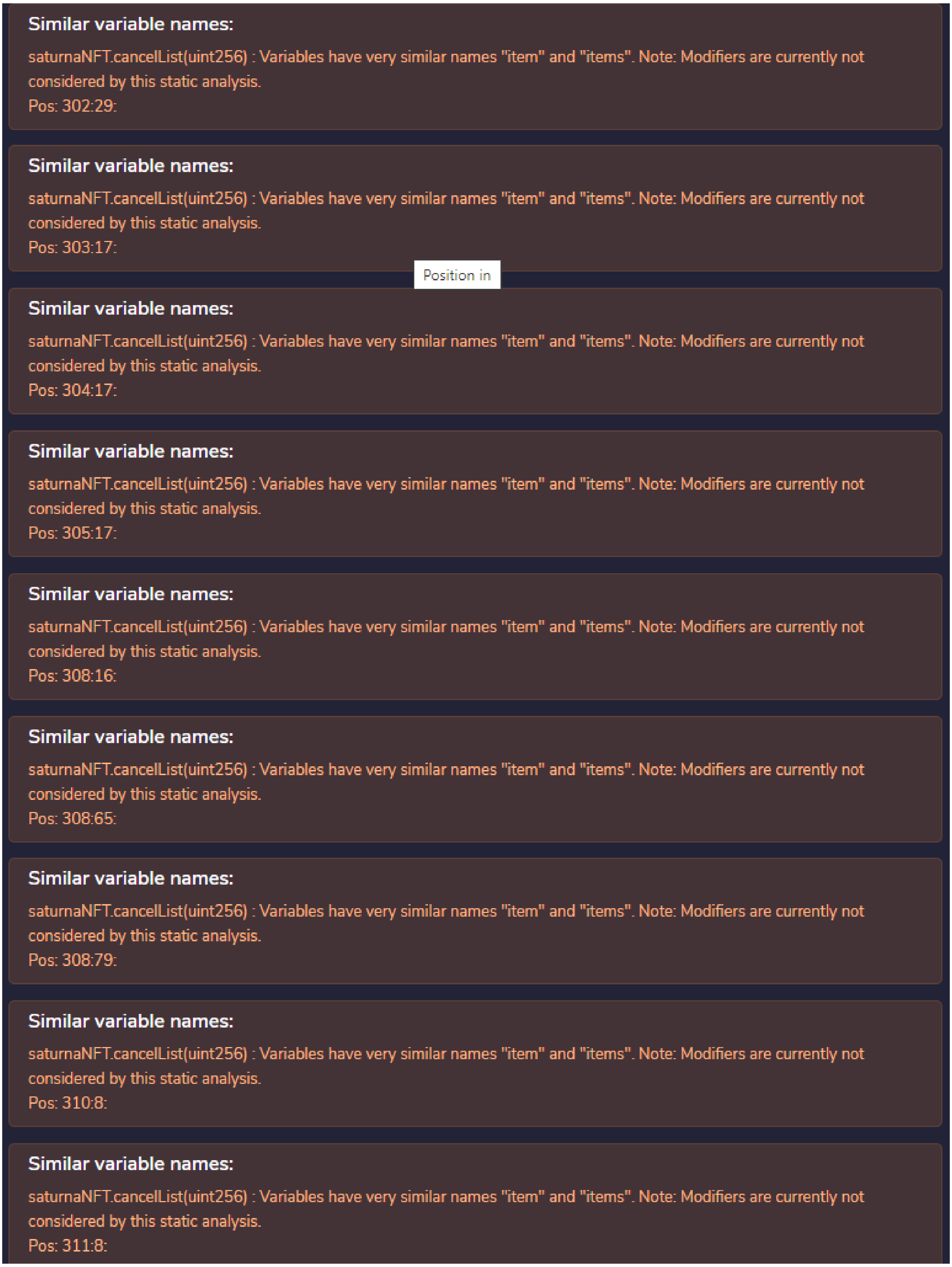
saturnaNFT.cancelAuction(uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 296:8:

Similar variable names:

saturnaNFT.cancelAuction(uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 297:8:

Similar variable names:

saturnaNFT.cancelList(uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 302:8:



Similar variable names:

saturnaNFT.cancelList(uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 302:29:

Similar variable names:

saturnaNFT.cancelList(uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 303:17:

Similar variable names:

saturnaNFT.cancelList(uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 304:17:

Similar variable names:

saturnaNFT.cancelList(uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 305:17:

Similar variable names:

saturnaNFT.cancelList(uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 308:16:

Similar variable names:

saturnaNFT.cancelList(uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 308:65:

Similar variable names:

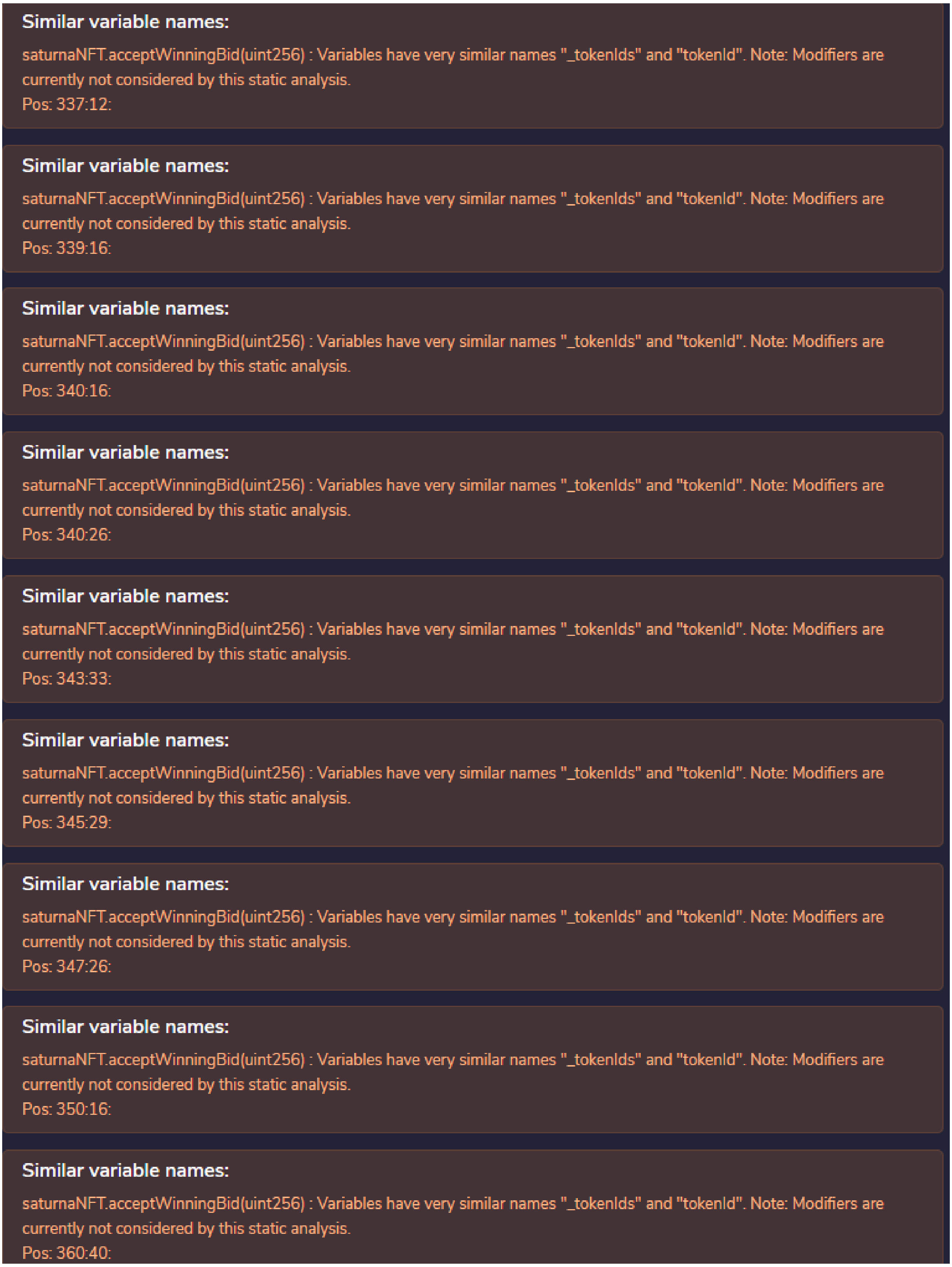
saturnaNFT.cancelList(uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 308:79:

Similar variable names:

saturnaNFT.cancelList(uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 310:8:

Similar variable names:

saturnaNFT.cancelList(uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 311:8:



Similar variable names:

saturnaNFT.acceptWinningBid(uint256) : Variables have very similar names "_tokenIds" and "tokenId". Note: Modifiers are currently not considered by this static analysis.
Pos: 337:12:

Similar variable names:

saturnaNFT.acceptWinningBid(uint256) : Variables have very similar names "_tokenIds" and "tokenId". Note: Modifiers are currently not considered by this static analysis.
Pos: 339:16:

Similar variable names:

saturnaNFT.acceptWinningBid(uint256) : Variables have very similar names "_tokenIds" and "tokenId". Note: Modifiers are currently not considered by this static analysis.
Pos: 340:16:

Similar variable names:

saturnaNFT.acceptWinningBid(uint256) : Variables have very similar names "_tokenIds" and "tokenId". Note: Modifiers are currently not considered by this static analysis.
Pos: 340:26:

Similar variable names:

saturnaNFT.acceptWinningBid(uint256) : Variables have very similar names "_tokenIds" and "tokenId". Note: Modifiers are currently not considered by this static analysis.
Pos: 343:33:

Similar variable names:

saturnaNFT.acceptWinningBid(uint256) : Variables have very similar names "_tokenIds" and "tokenId". Note: Modifiers are currently not considered by this static analysis.
Pos: 345:29:

Similar variable names:

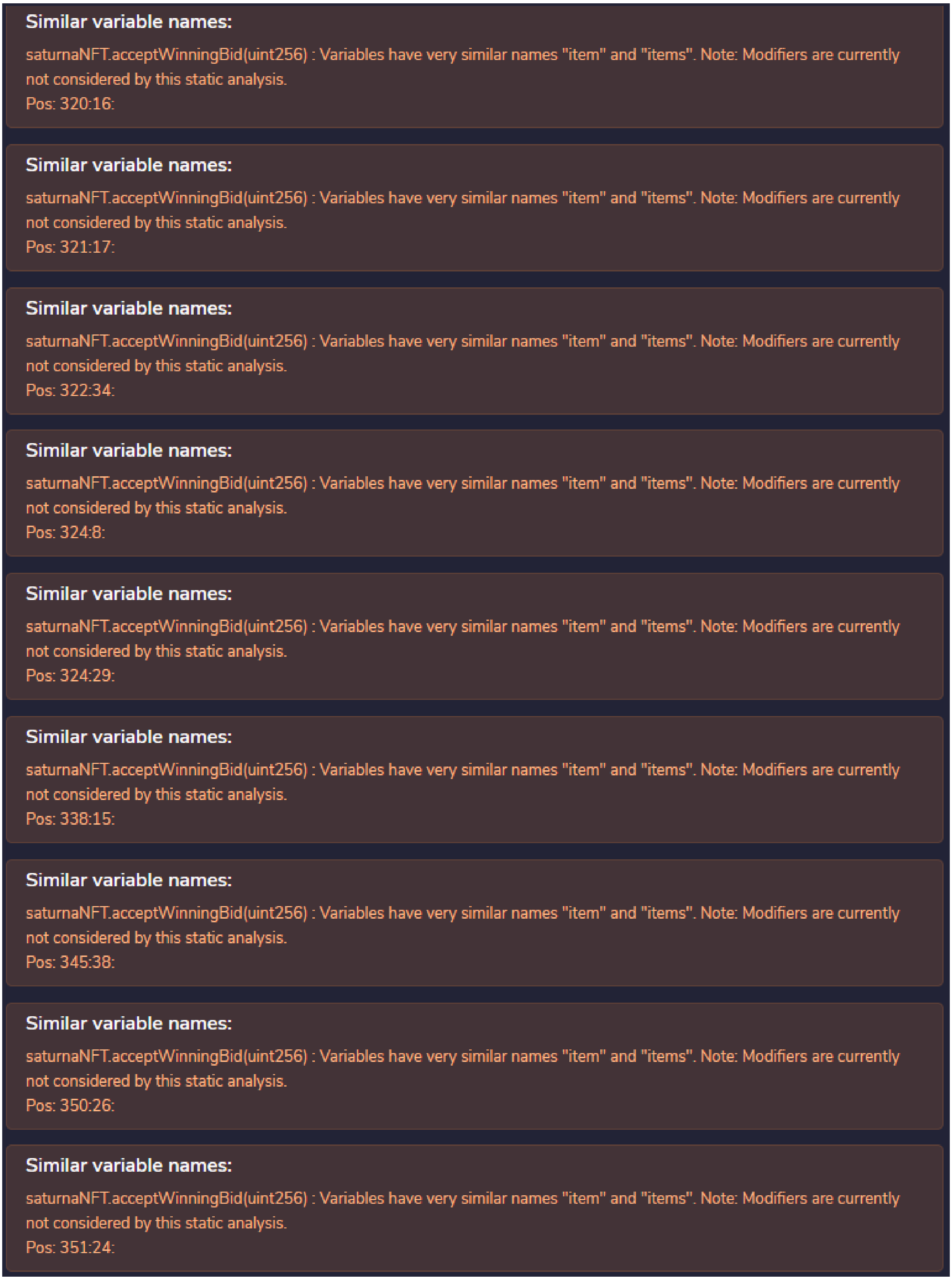
saturnaNFT.acceptWinningBid(uint256) : Variables have very similar names "_tokenIds" and "tokenId". Note: Modifiers are currently not considered by this static analysis.
Pos: 347:26:

Similar variable names:

saturnaNFT.acceptWinningBid(uint256) : Variables have very similar names "_tokenIds" and "tokenId". Note: Modifiers are currently not considered by this static analysis.
Pos: 350:16:

Similar variable names:

saturnaNFT.acceptWinningBid(uint256) : Variables have very similar names "_tokenIds" and "tokenId". Note: Modifiers are currently not considered by this static analysis.
Pos: 360:40:



Similar variable names:

saturnaNFT.acceptWinningBid(uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 320:16:

Similar variable names:

saturnaNFT.acceptWinningBid(uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 321:17:

Similar variable names:

saturnaNFT.acceptWinningBid(uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 322:34:

Similar variable names:

saturnaNFT.acceptWinningBid(uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 324:8:

Similar variable names:

saturnaNFT.acceptWinningBid(uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 324:29:

Similar variable names:

saturnaNFT.acceptWinningBid(uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 338:15:

Similar variable names:

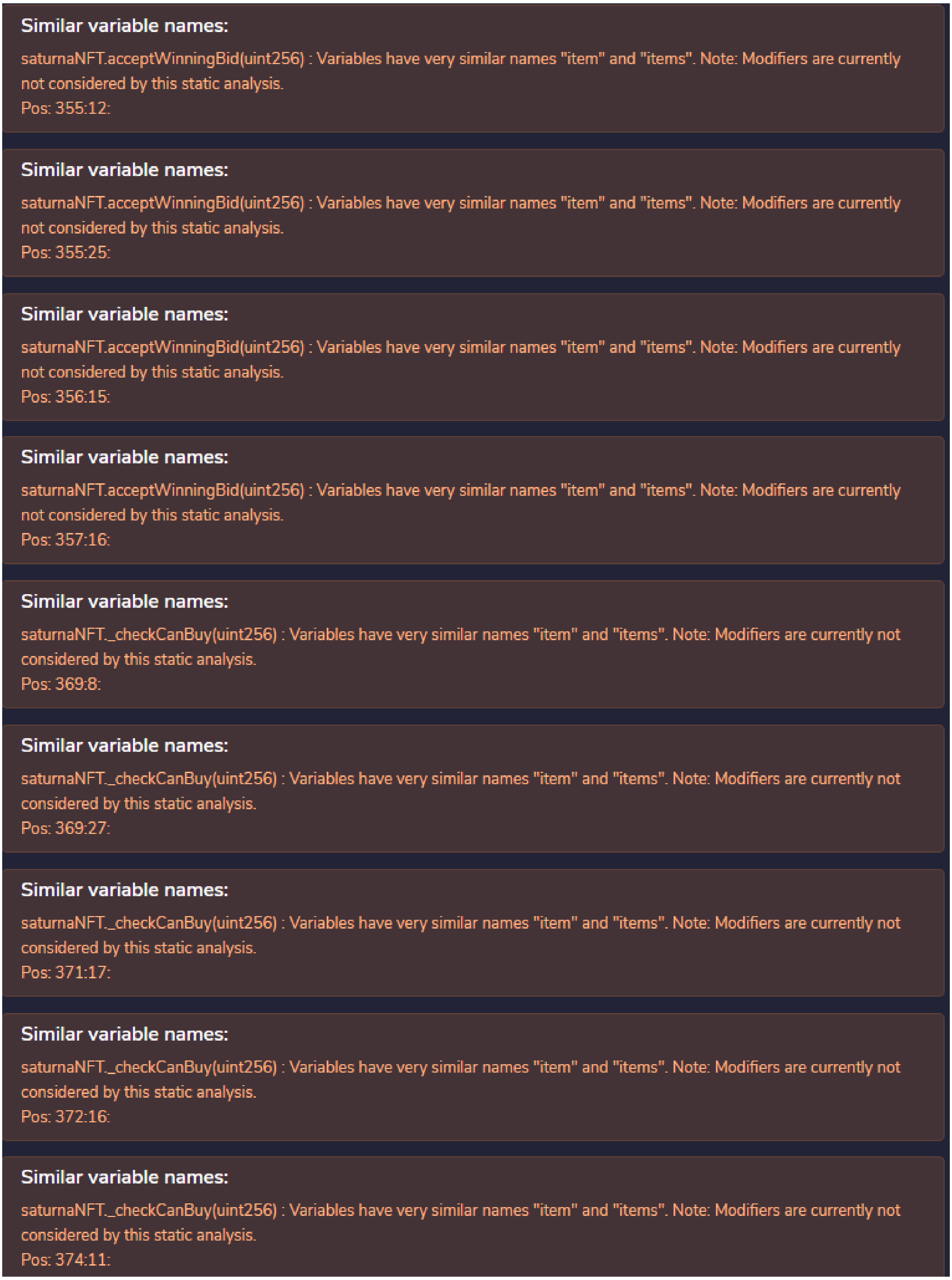
saturnaNFT.acceptWinningBid(uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 345:38:

Similar variable names:

saturnaNFT.acceptWinningBid(uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 350:26:

Similar variable names:

saturnaNFT.acceptWinningBid(uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 351:24:



Similar variable names:

saturnaNFT.acceptWinningBid(uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 355:12:

Similar variable names:

saturnaNFT.acceptWinningBid(uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 355:25:

Similar variable names:

saturnaNFT.acceptWinningBid(uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 356:15:

Similar variable names:

saturnaNFT.acceptWinningBid(uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 357:16:

Similar variable names:

saturnaNFT._checkCanBuy(uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 369:8:

Similar variable names:

saturnaNFT._checkCanBuy(uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 369:27:

Similar variable names:

saturnaNFT._checkCanBuy(uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 371:17:

Similar variable names:

saturnaNFT._checkCanBuy(uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 372:16:

Similar variable names:

saturnaNFT._checkCanBuy(uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 374:11:

Similar variable names:

saturnaNFT._checkCanBuy(uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 375:39:

Similar variable names:

saturnaNFT._checkCanBuy(uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 376:39:

Similar variable names:

saturnaNFT._checkCanBuy(uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 379:12:

Similar variable names:

saturnaNFT._checkCanBuy(uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 380:28:

Similar variable names:

saturnaNFT._checkCanBuy(uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 380:53:

Similar variable names:

saturnaNFT._checkCanBuy(uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 383:16:

Similar variable names:

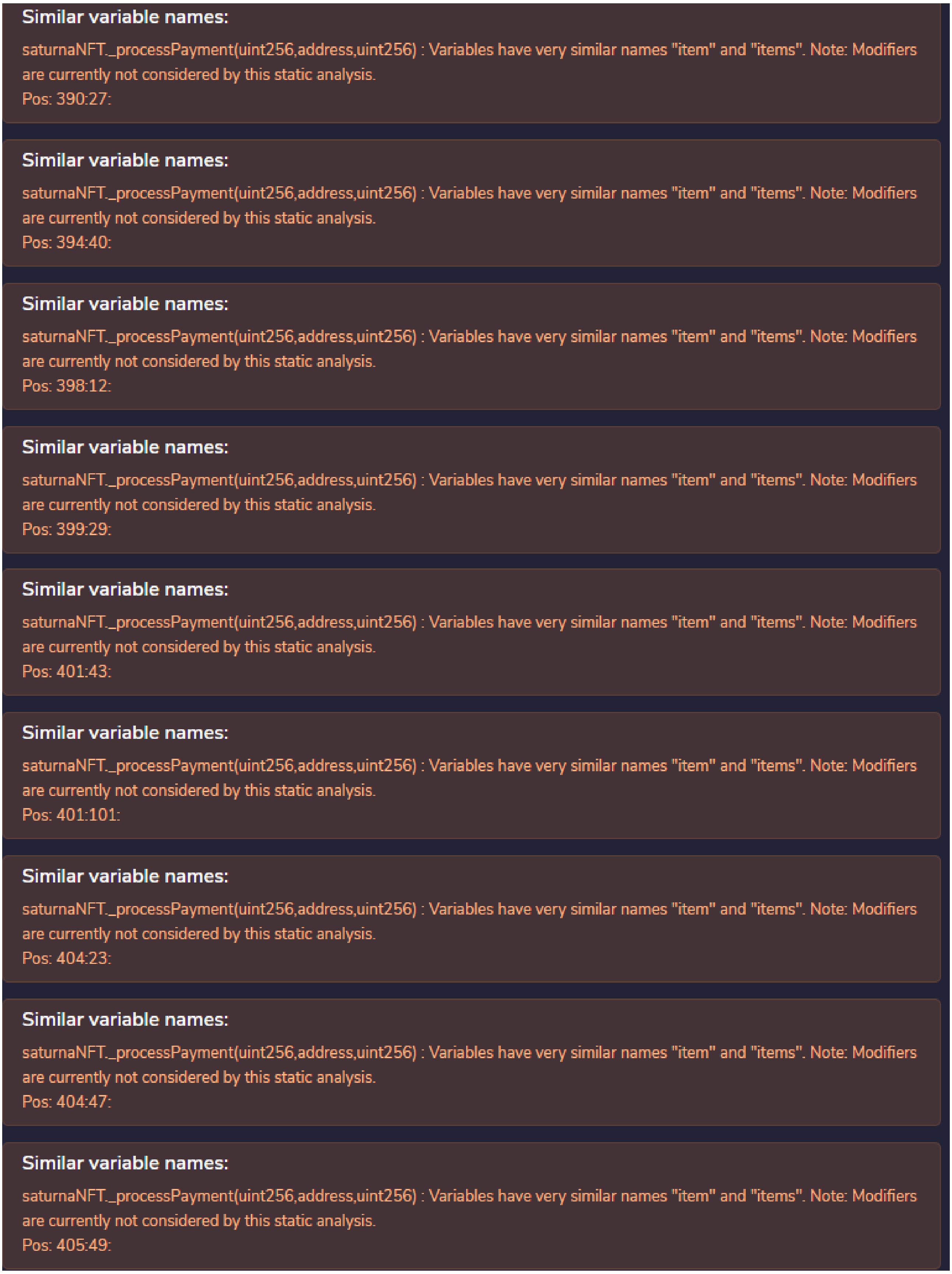
saturnaNFT._checkCanBuy(uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 384:15:

Similar variable names:

saturnaNFT._checkCanBuy(uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 385:17:

Similar variable names:

saturnaNFT._processPayment(uint256,address,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 390:8:



Similar variable names:

saturnaNFT._processPayment(uint256,address,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 390:27:

Similar variable names:

saturnaNFT._processPayment(uint256,address,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 394:40:

Similar variable names:

saturnaNFT._processPayment(uint256,address,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 398:12:

Similar variable names:

saturnaNFT._processPayment(uint256,address,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 399:29:

Similar variable names:

saturnaNFT._processPayment(uint256,address,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 401:43:

Similar variable names:

saturnaNFT._processPayment(uint256,address,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 401:101:

Similar variable names:

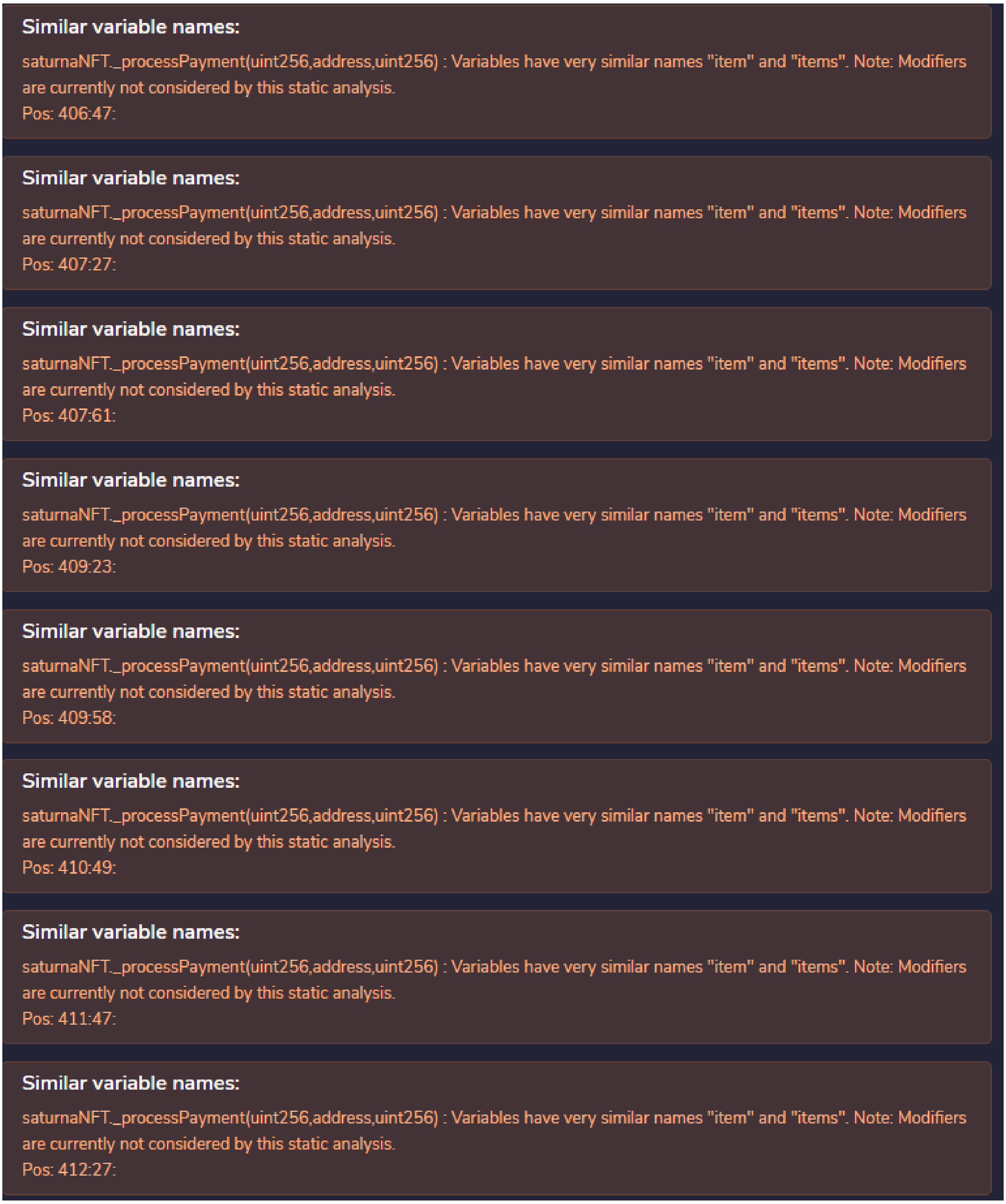
saturnaNFT._processPayment(uint256,address,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 404:23:

Similar variable names:

saturnaNFT._processPayment(uint256,address,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 404:47:

Similar variable names:

saturnaNFT._processPayment(uint256,address,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 405:49:



Similar variable names:

saturnaNFT._processPayment(uint256,address,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 406:47:

Similar variable names:

saturnaNFT._processPayment(uint256,address,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 407:27:

Similar variable names:

saturnaNFT._processPayment(uint256,address,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 407:61:

Similar variable names:

saturnaNFT._processPayment(uint256,address,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 409:23:

Similar variable names:

saturnaNFT._processPayment(uint256,address,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 409:58:

Similar variable names:

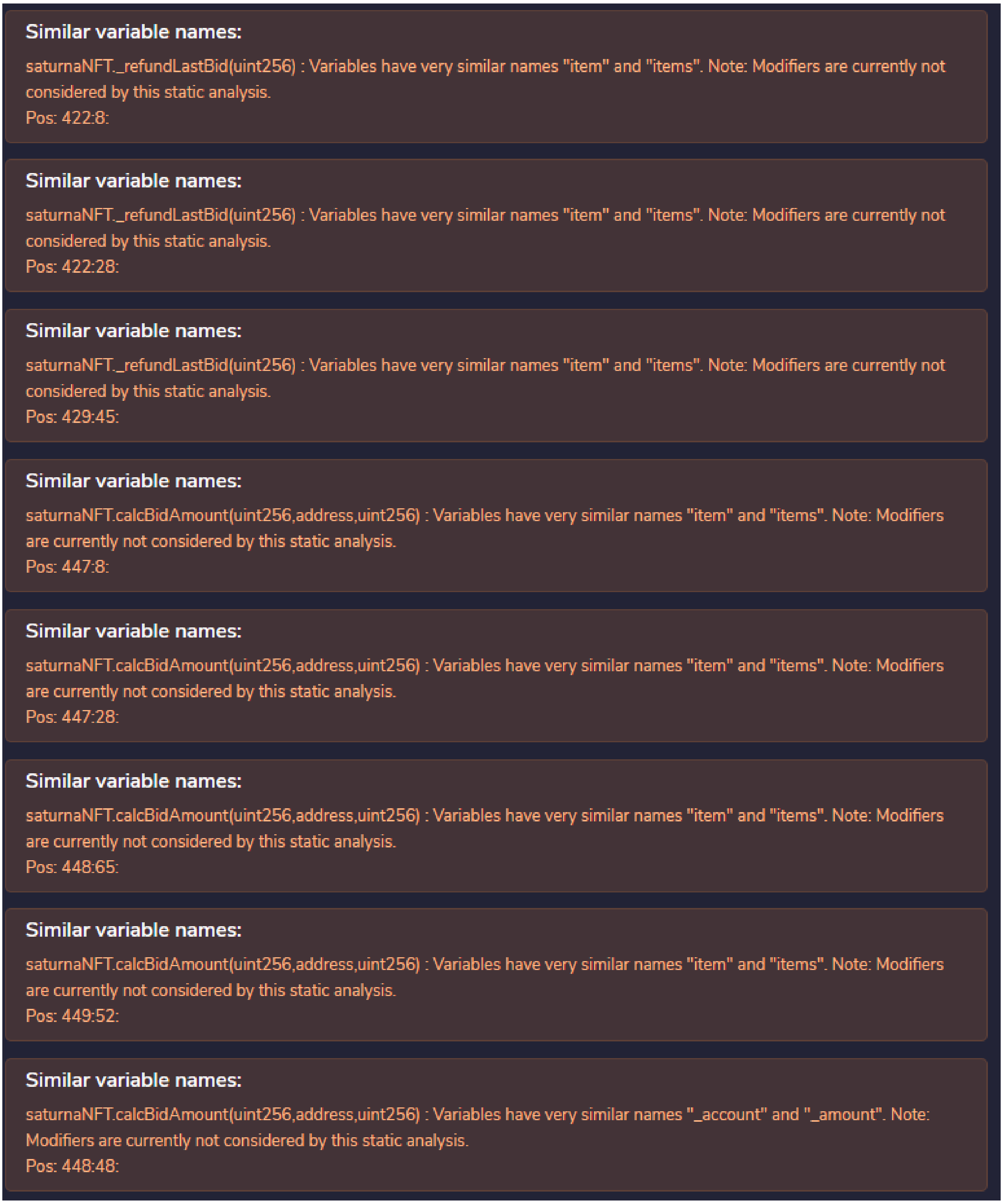
saturnaNFT._processPayment(uint256,address,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 410:49:

Similar variable names:

saturnaNFT._processPayment(uint256,address,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 411:47:

Similar variable names:

saturnaNFT._processPayment(uint256,address,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 412:27:



Similar variable names:

saturnaNFT._refundLastBid(uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 422:8:

Similar variable names:

saturnaNFT._refundLastBid(uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 422:28:

Similar variable names:

saturnaNFT._refundLastBid(uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 429:45:

Similar variable names:

saturnaNFT.calcBidAmount(uint256,address,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 447:8:

Similar variable names:

saturnaNFT.calcBidAmount(uint256,address,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 447:28:

Similar variable names:

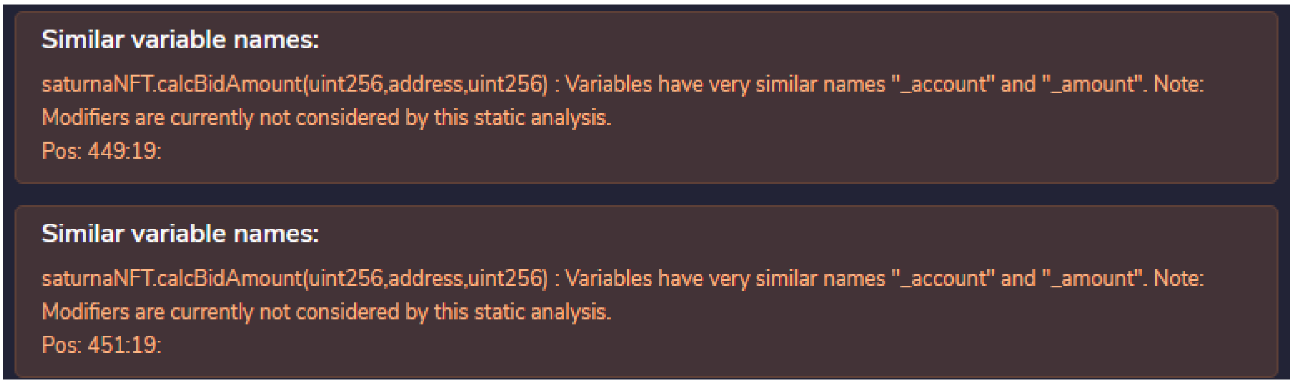
saturnaNFT.calcBidAmount(uint256,address,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 448:65:

Similar variable names:

saturnaNFT.calcBidAmount(uint256,address,uint256) : Variables have very similar names "item" and "items". Note: Modifiers are currently not considered by this static analysis.
Pos: 449:52:

Similar variable names:

saturnaNFT.calcBidAmount(uint256,address,uint256) : Variables have very similar names "_account" and "_amount". Note: Modifiers are currently not considered by this static analysis.
Pos: 448:48:



Similar variable names:

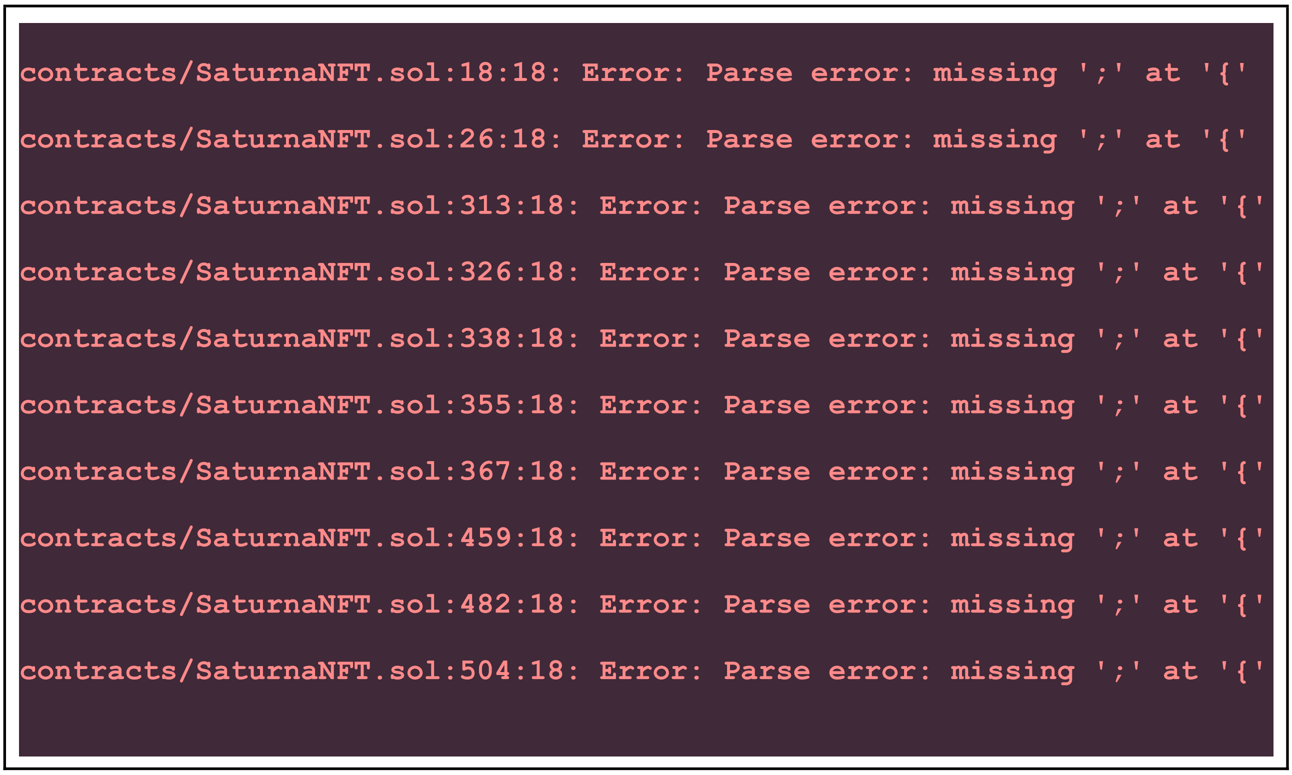
saturnaNFT.calcBidAmount(uint256,address,uint256) : Variables have very similar names "_account" and "_amount". Note: Modifiers are currently not considered by this static analysis.
Pos: 449:19:

Similar variable names:

saturnaNFT.calcBidAmount(uint256,address,uint256) : Variables have very similar names "_account" and "_amount". Note: Modifiers are currently not considered by this static analysis.
Pos: 451:19:

Solhint Linter

SaturnaNFT.sol



```
contracts/SaturnaNFT.sol:18:18: Error: Parse error: missing ';' at '{'

contracts/SaturnaNFT.sol:26:18: Error: Parse error: missing ';' at '{'

contracts/SaturnaNFT.sol:313:18: Error: Parse error: missing ';' at '{'

contracts/SaturnaNFT.sol:326:18: Error: Parse error: missing ';' at '{'

contracts/SaturnaNFT.sol:338:18: Error: Parse error: missing ';' at '{'

contracts/SaturnaNFT.sol:355:18: Error: Parse error: missing ';' at '{'

contracts/SaturnaNFT.sol:367:18: Error: Parse error: missing ';' at '{'

contracts/SaturnaNFT.sol:459:18: Error: Parse error: missing ';' at '{'

contracts/SaturnaNFT.sol:482:18: Error: Parse error: missing ';' at '{'

contracts/SaturnaNFT.sol:504:18: Error: Parse error: missing ';' at '{'
```

Email: audit@EtherAuthority.io