RD AUDITORS

OVRLand Renting & OVRVestingV2 Smart Contract, Code Review and Security Analysis Report

Customer: OVER
Prepared on: 30th September 2022
Platform: Polygon
Language: Solidity

rdauditors.com

Table of Contents

Disclaimer

This document may contain confidential information about its systems and intellectual property of the customer as well as information about potential vulnerabilities and methods of their exploitation.

The report containing confidential information can be used internally by the customer or it can be disclosed publicly after all vulnerabilities are fixed - upon the decision of the customer.

Document

Name	Smart Contract Code Review and Security Analysis Report of OVER
Platform	Polygon / Solidity
File 1	OVRLandExperience.sol
MD5 hash	3d4352fdd15d7732b931d776b14c3d74
SHA256 hash	20cae188624df5f69fef9380b151386e3e755e8ebbe6713d545a1826ca70c4a2
File 2	OVRLandRenting.sol
MD5 hash	baf61d3905fc6196cbb707055b94151b
SHA256 hash	b14ae0e02d6a2819cd6acb4f7e0a0c2de4154af5e71dd215e9802c6c11e8ac2c
File 3	OVRVestingV2.sol
MD5 hash	597f34ae1ae3b374a51729677cf2f26c
SHA256 hash	220518a4a73b861595665bdb18adac79afa854f244720d4101a8e694190ca989
Date	30/9/2022

Introduction

RD Auditors (Consultant) were contracted by OVER (Customer) to conduct a Smart Contracts Code Review and Security Analysis. This report represents the findings of the security assessment of the customer`s smart contracts and its code review conducted between 20th - 30th Sept 2022.

This contract consists of three files.

Project Scope

The scope of the project is a smart contract. We have scanned this smart contract for commonly known and more specific vulnerabilities, below are those considered (the full list includes but is not limited to):

- Reentrancy
- Timestamp Dependence
- Gas Limit and Loops
- DoS with (Unexpected) Throw
- DoS with Block Gas Limit
- Transaction-Ordering Dependence
- Byte array vulnerabilities
- Style guide violation
- Transfer forwards all gas
- ERC20 API violation
- Malicious libraries
- Compiler version not fixed
- Unchecked external call - Unchecked math
- Unsafe type inference
- Implicit visibility level

Executive Summary

According to the assessment, the customer's solidity smart contract is **well-Secured.**

You are Here

🟥 **Insecure** 🟧 **Poorly Secured** 🟩 **Secure** 🟦 **Well-Secured**

Automated checks are with smartDec, Mythril, Slither and remix IDE. All issues were performed by our team, which included the analysis of code functionality, the manual audit found during automated analysis were manually reviewed and applicable vulnerabilities are presented in the audit overview section. The general overview is presented in the AS-IS section and all issues found are located in the audit overview section.

We found the following;

Total Issues	0
🟥 Critical	0
🟧 High	0
🟨 Medium	0
🟩 Low	0
🟦 Very Low	0

Code Quality

Please note that within this report IBEP20, Context, Ownable are taken from the popular OpenZeppelin library.

The libraries within this smart contract are part of a 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 to a specific address and its properties/methods can be reused many times by other contracts.

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

Documentation

We were given the source code as a Github link:

https://github.com/OVR-Platform/ovrland-renting

The hash of that file is mentioned in the table. As mentioned above, It's recommended to write comments in the smart contract code, so anyone can quickly understand the programming flow as well as complex code logic.

Comments are very helpful in understanding the overall architecture of the protocol. It also provides a clear overview of the system components, including helpful details, like the lifetime of the background script.

Use of Dependencies

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

AS-IS Overview

OVRLandRenting

File And Function Level Report

File:	OVRLandExperience.sol
Contract:	OVRLandExperience
Observation:	Passed
Test Report:	Passed

Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	initialize	public	Passed	All Passed	No Issue	Passed
2	addAdmin	public	Passed	All Passed	No Issue	Passed
3	removeAdmin	public	Passed	All Passed	No Issue	Passed
4	getExperienceInfo	read	Passed	All Passed	No Issue	Passed
5	isTokenRented	read	Passed	All Passed	No Issue	Passed
6	experienceURI	read	Passed	All Passed	No Issue	Passed
7	rentingExpiration	read	Passed	All Passed	No Issue	Passed
8	updateExperience	public	Passed	All Passed	No Issue	Passed
9	adminUpdateExperience	public	Passed	All Passed	No Issue	Passed
10	startTokenRenting	public	Passed	All Passed	No Issue	Passed
11	setContainerAddress	public	Passed	All Passed	No Issue	Passed

12	setRentingAddress	public	Passed	All Passed	No Issue	Passed

File:	OVRLandRenting.sol
Contract:	OVRLandRenting
Observation:	Passed
Test Report:	Passed

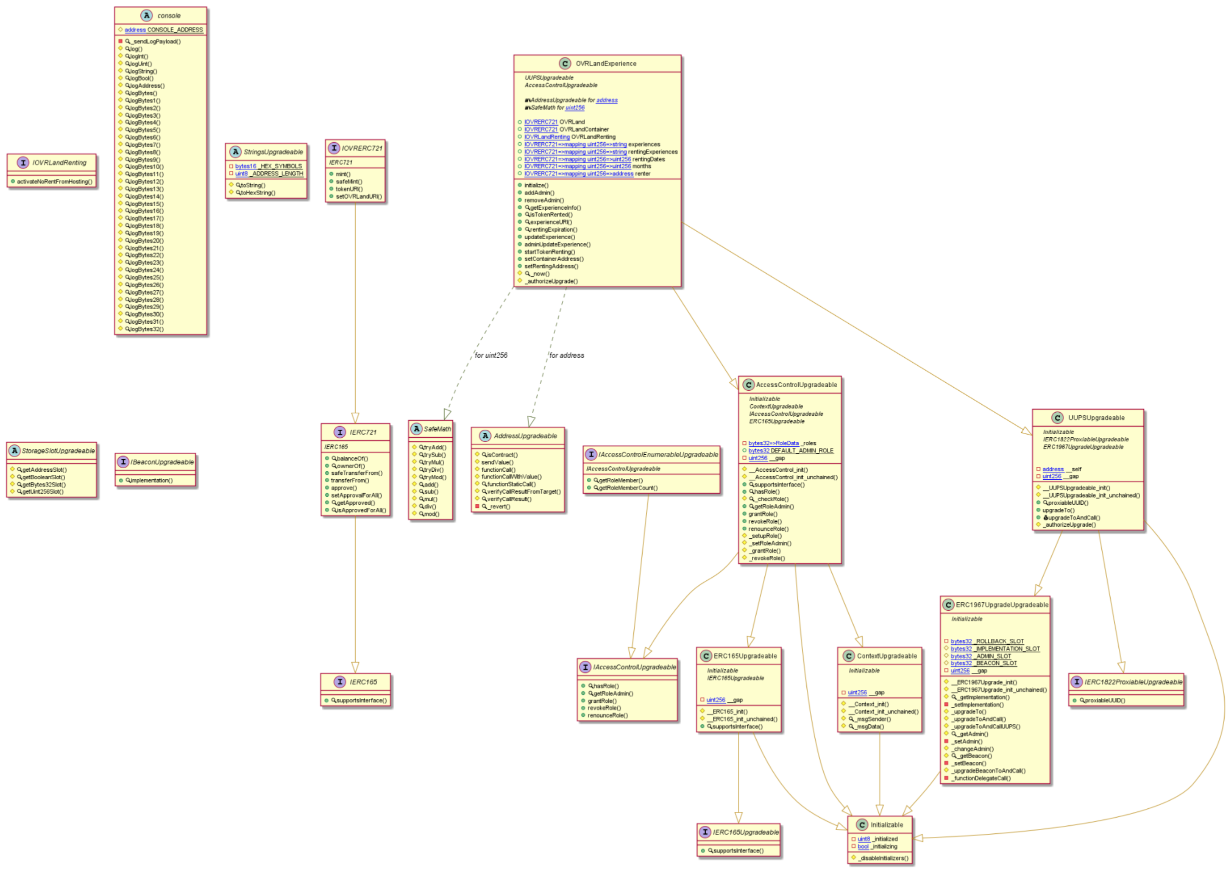
Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	initialize	public	Passed	All Passed	No Issue	Passed
2	addAdmin	public	Passed	All Passed	No Issue	Passed
3	removeAdmin	public	Passed	All Passed	No Issue	Passed
4	setBasePriceChanger	public	Passed	All Passed	No Issue	Passed
5	removeBasePriceChanger	public	Passed	All Passed	No Issue	Passed
6	setOVRLandExperienceAddress	public	Passed	All Passed	No Issue	Passed
7	setFeePerc	public	Passed	All Passed	No Issue	Passed
8	setFeeReceiver	public	Passed	All Passed	No Issue	Passed
9	activateNoRent	public	Passed	All Passed	No Issue	Passed
10	activateNoRentFromHosting	public	Passed	All Passed	No Issue	Passed
11	deactivateNoRent	public	Passed	All Passed	No Issue	Passed

12	setHostingContract	public	Passed	All Passed	No Issue	Passed
13	setPriceLandAddress	public	Passed	All Passed	No Issue	Passed
14	isNoRentActive	read	Passed	All Passed	No Issue	Passed
15	setOVRFloorPrice	public	Passed	All Passed	No Issue	Passed
16	changeNoRentPriceLand	public	Passed	All Passed	No Issue	Passed
17	changeNoRentPriceContainer	public	Passed	All Passed	No Issue	Passed
18	changeNoRentDuration	public	Passed	All Passed	No Issue	Passed
19	getRentingInfo	read	Passed	All Passed	No Issue	Passed
20	placeOffer	public	Passed	All Passed	No Issue	Passed
21	placeOfferInternal	internal	Passed	All Passed	No Issue	Passed
22	repayPreviousOfferer	internal	Passed	All Passed	No Issue	Passed
23	saveOffer	internal	Passed	All Passed	No Issue	Passed
24	acceptOffer	public	Passed	All Passed	No Issue	Passed
25	cancelOffer	public	Passed	All Passed	No Issue	Passed
26	changeBasePriceLands	public	Passed	All Passed	No Issue	Passed
27	changeBasePriceContainers	public	Passed	All Passed	No Issue	Passed
28	basePrice	read	Passed	All Passed	No Issue	Passed
29	pause	public	Passed	All Passed	No Issue	Passed
30	unpause	public	Passed	All Passed	No Issue	Passed

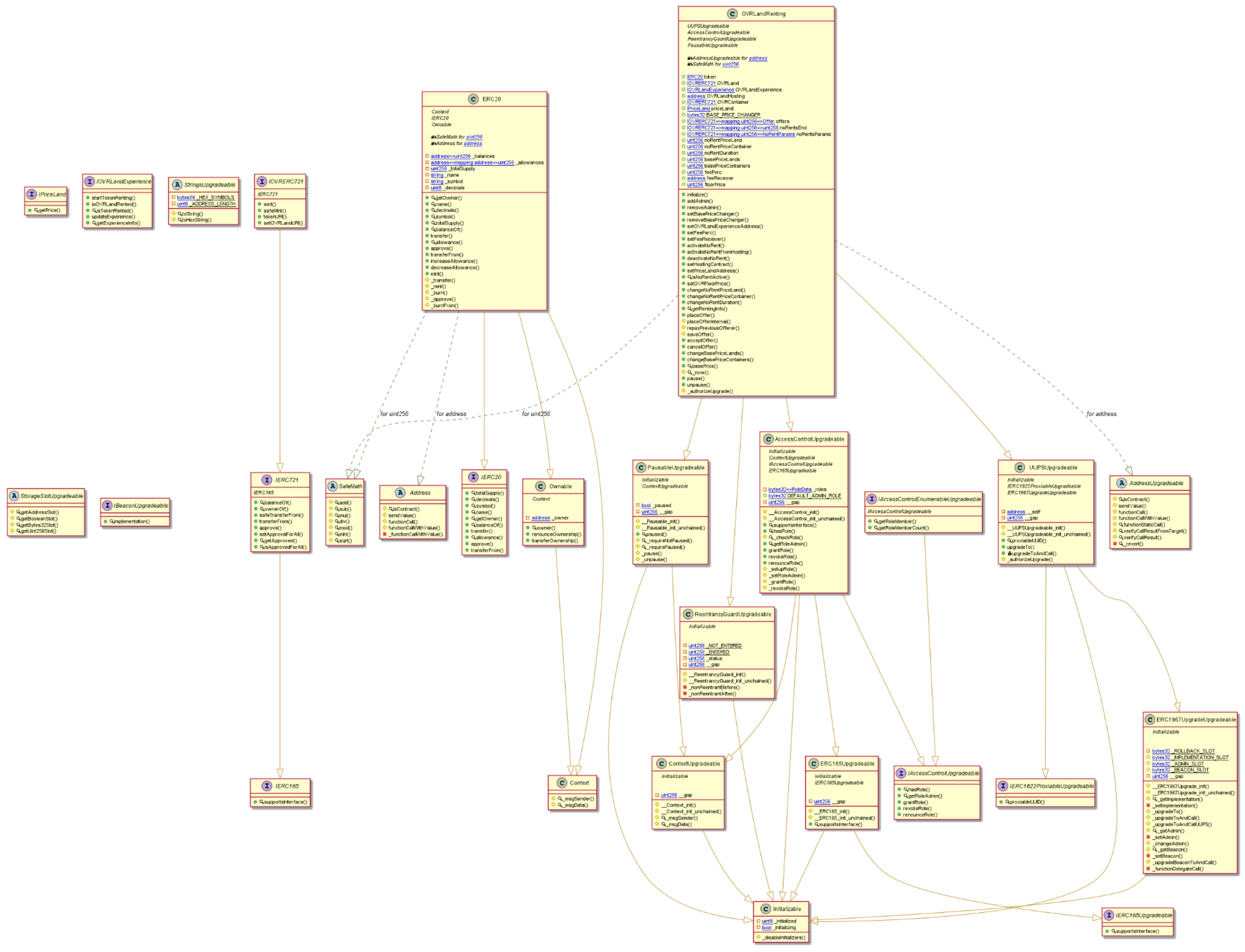
File:	OVRVestingV2.sol
Contract:	OVRVestingV2
Observation:	Passed
Test Report:	Passed

Sl.	Function	Type	Observation	Test Report	Conclusion	Score
1	granting	public	Passed	All Passed	No Issue	Passed
2	unlockVestedTokens	external	Passed	All Passed	No Issue	Passed
3	calcAmountToWithdraw	read	Passed	All Passed	No Issue	Passed
4	revoke	public	Passed	All Passed	No Issue	Passed
5	addAdmin	public	Passed	All Passed	No Issue	Passed
6	removeAdmin	public	Passed	All Passed	No Issue	Passed

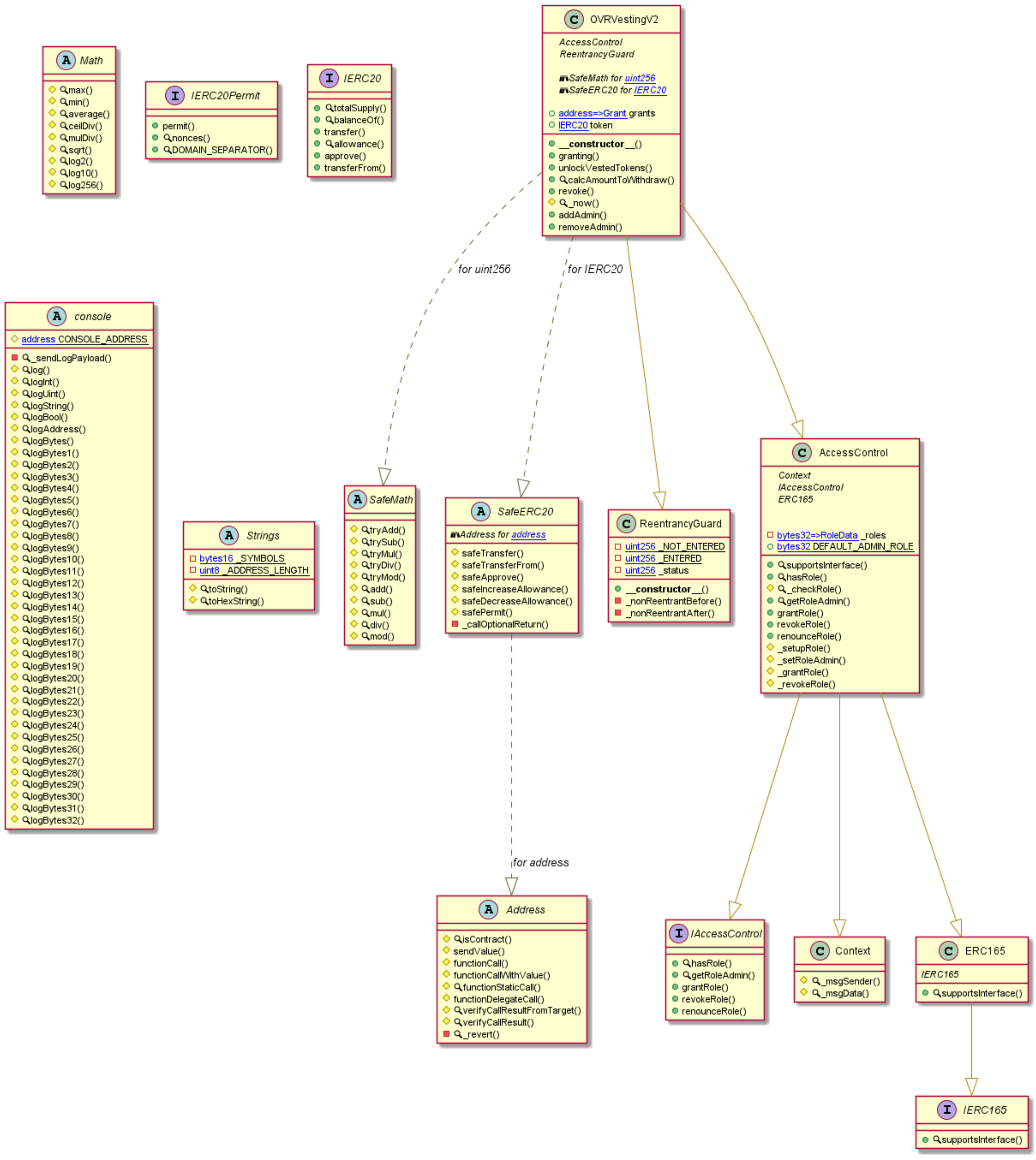
Code Flow Diagram - OVRLandExperience



Code Flow Diagram - OVRLandRenting

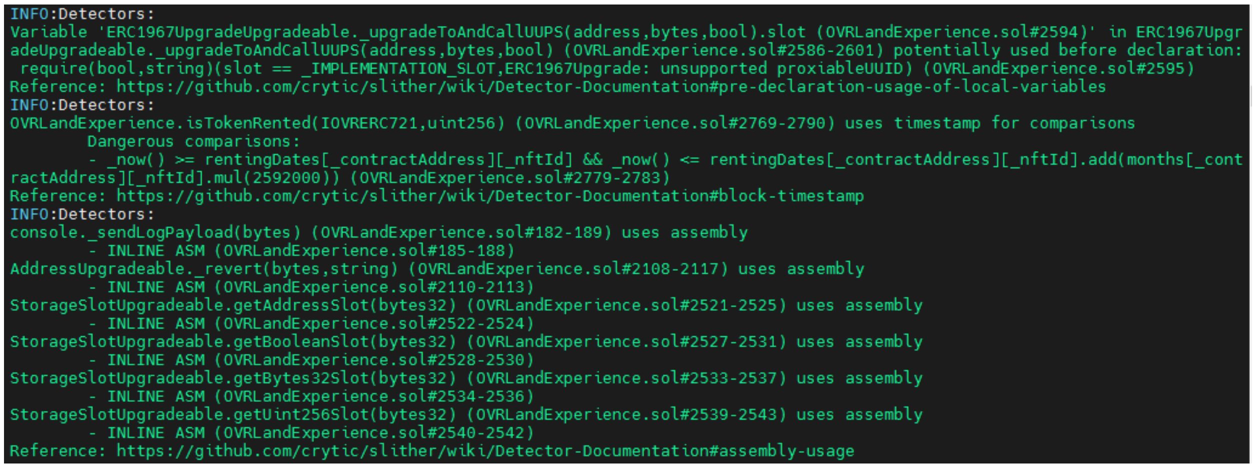


Code Flow Diagram - OVRVestingV2

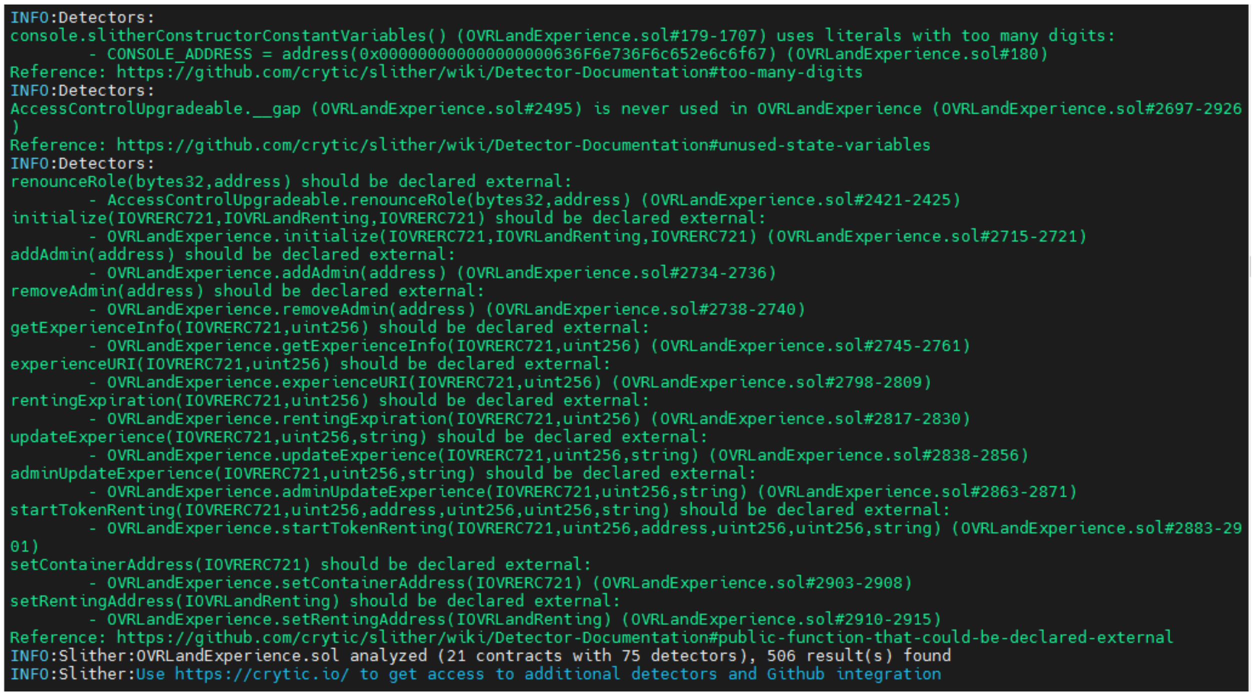


Code Flow Diagram - Slither Results Log

OVRLandExperience



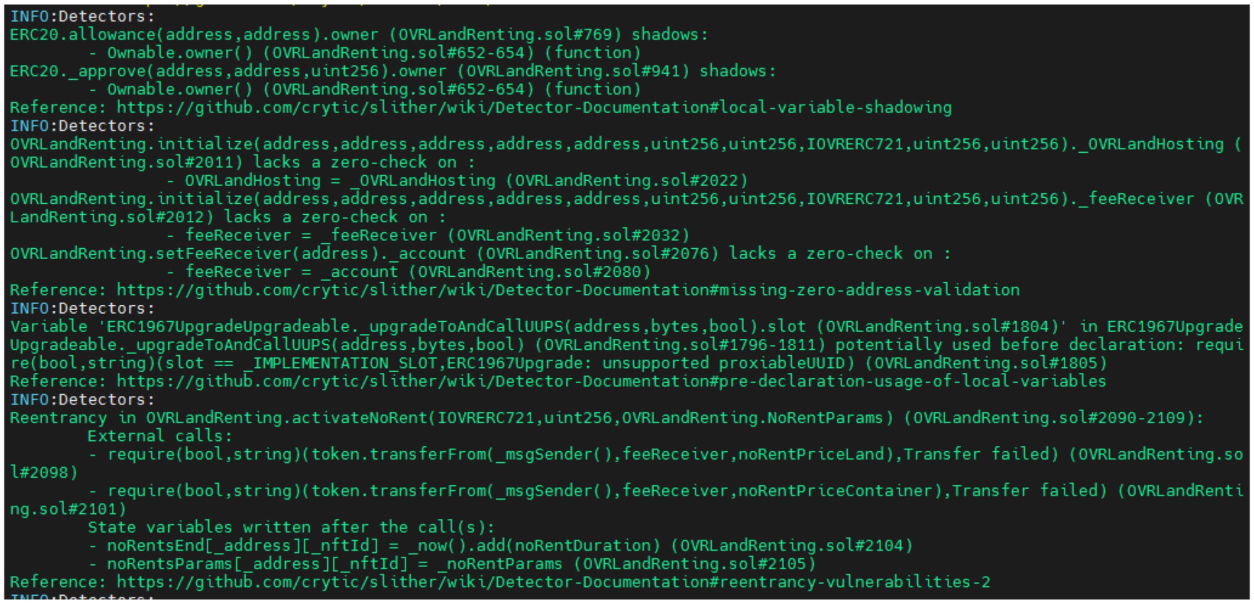
```
INFO:Detectors:
Variable 'ERC1967UpgradeUpgradeable._upgradeToAndCallUUPS(address,bytes,bool).slot (OVRLandExperience.sol#2594)' in ERC1967Upgr
adeUpgradeable._upgradeToAndCallUUPS(address,bytes,bool) (OVRLandExperience.sol#2586-2601) potentially used before declaration:
 require(bool,string)(slot == _IMPLEMENTATION_SLOT,ERC1967Upgrade: unsupported proxiableUUID) (OVRLandExperience.sol#2595)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables
INFO:Detectors:
OVRLandExperience.isTokenRented(IOVRERC721,uint256) (OVRLandExperience.sol#2769-2790) uses timestamp for comparisons
        Dangerous comparisons:
        - _now() >= rentingDates[_contractAddress][_nftId] && _now() <= rentingDates[_contractAddress][_nftId].add(months[_cont
ractAddress][_nftId].mul(2592000)) (OVRLandExperience.sol#2779-2783)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp
INFO:Detectors:
console._sendLogPayload(bytes) (OVRLandExperience.sol#182-189) uses assembly
        - INLINE ASM (OVRLandExperience.sol#185-188)
AddressUpgradeable._revert(bytes,string) (OVRLandExperience.sol#2108-2117) uses assembly
        - INLINE ASM (OVRLandExperience.sol#2110-2113)
StorageSlotUpgradeable.getAddressSlot(bytes32) (OVRLandExperience.sol#2521-2525) uses assembly
        - INLINE ASM (OVRLandExperience.sol#2522-2524)
StorageSlotUpgradeable.getBooleanSlot(bytes32) (OVRLandExperience.sol#2527-2531) uses assembly
        - INLINE ASM (OVRLandExperience.sol#2528-2530)
StorageSlotUpgradeable.getBytes32Slot(bytes32) (OVRLandExperience.sol#2533-2537) uses assembly
        - INLINE ASM (OVRLandExperience.sol#2534-2536)
StorageSlotUpgradeable.getUint256Slot(bytes32) (OVRLandExperience.sol#2539-2543) uses assembly
        - INLINE ASM (OVRLandExperience.sol#2540-2542)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
```



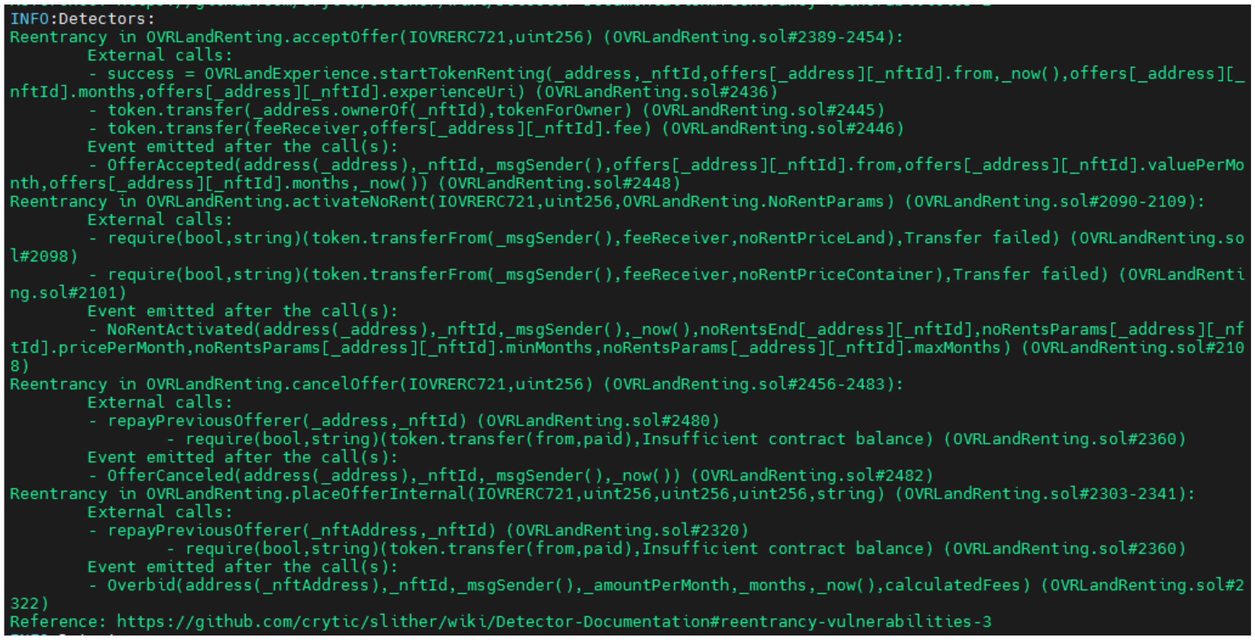
```
INFO:Detectors:
console.slitherConstructorConstantVariables() (OVRLandExperience.sol#179-1707) uses literals with too many digits:
        - CONSOLE_ADDRESS = address(0x000000000000000000636F6e736F6c652e6c6f67) (OVRLandExperience.sol#180)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits
INFO:Detectors:
AccessControlUpgradeable.__gap (OVRLandExperience.sol#2495) is never used in OVRLandExperience (OVRLandExperience.sol#2697-2926
)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variables
INFO:Detectors:
renounceRole(bytes32,address) should be declared external:
        - AccessControlUpgradeable.renounceRole(bytes32,address) (OVRLandExperience.sol#2421-2425)
initialize(IOVRERC721,IOVRLandRenting,IOVRERC721) should be declared external:
        - OVRLandExperience.initialize(IOVRERC721,IOVRLandRenting,IOVRERC721) (OVRLandExperience.sol#2715-2721)
addAdmin(address) should be declared external:
        - OVRLandExperience.addAdmin(address) (OVRLandExperience.sol#2734-2736)
removeAdmin(address) should be declared external:
        - OVRLandExperience.removeAdmin(address) (OVRLandExperience.sol#2738-2740)
getExperienceInfo(IOVRERC721,uint256) should be declared external:
        - OVRLandExperience.getExperienceInfo(IOVRERC721,uint256) (OVRLandExperience.sol#2745-2761)
experienceURI(IOVRERC721,uint256) should be declared external:
        - OVRLandExperience.experienceURI(IOVRERC721,uint256) (OVRLandExperience.sol#2798-2809)
rentingExpiration(IOVRERC721,uint256) should be declared external:
        - OVRLandExperience.rentingExpiration(IOVRERC721,uint256) (OVRLandExperience.sol#2817-2830)
updateExperience(IOVRERC721,uint256,string) should be declared external:
        - OVRLandExperience.updateExperience(IOVRERC721,uint256,string) (OVRLandExperience.sol#2838-2856)
adminUpdateExperience(IOVRERC721,uint256,string) should be declared external:
        - OVRLandExperience.adminUpdateExperience(IOVRERC721,uint256,string) (OVRLandExperience.sol#2863-2871)
startTokenRenting(IOVRERC721,uint256,address,uint256,uint256,string) should be declared external:
        - OVRLandExperience.startTokenRenting(IOVRERC721,uint256,address,uint256,uint256,string) (OVRLandExperience.sol#2883-29
01)
setContainerAddress(IOVRERC721) should be declared external:
        - OVRLandExperience.setContainerAddress(IOVRERC721) (OVRLandExperience.sol#2903-2908)
setRentingAddress(IOVRLandRenting) should be declared external:
        - OVRLandExperience.setRentingAddress(IOVRLandRenting) (OVRLandExperience.sol#2910-2915)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:OVRLandExperience.sol analyzed (21 contracts with 75 detectors), 506 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration
```

Code Flow Diagram - Slither Results Log

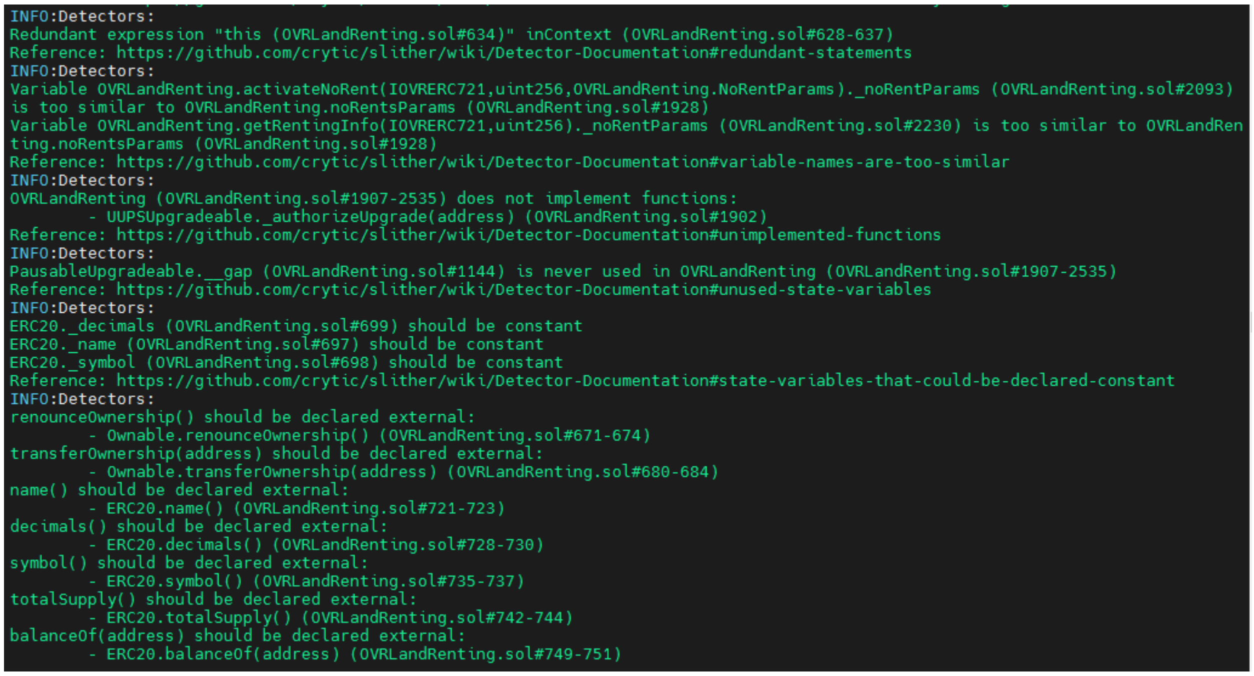
OVRLandRenting



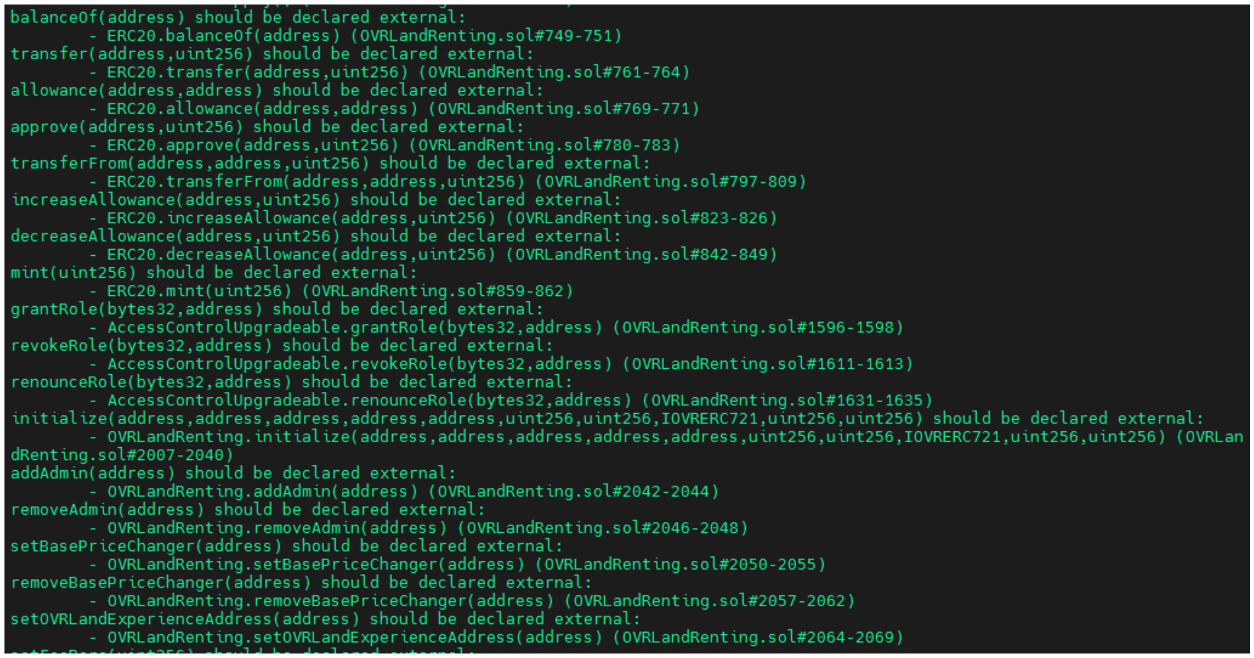
```
INFO:Detectors:
ERC20.allowance(address,address).owner (OVRLandRenting.sol#769) shadows:
        - Ownable.owner() (OVRLandRenting.sol#652-654) (function)
ERC20._approve(address,address,uint256).owner (OVRLandRenting.sol#941) shadows:
        - Ownable.owner() (OVRLandRenting.sol#652-654) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing
INFO:Detectors:
OVRLandRenting.initialize(address,address,address,address,address,uint256,uint256,IOVRERC721,uint256,uint256)._OVRLandHosting (
OVRLandRenting.sol#2011) lacks a zero-check on :
                - OVRLandHosting = _OVRLandHosting (OVRLandRenting.sol#2022)
OVRLandRenting.initialize(address,address,address,address,address,uint256,uint256,IOVRERC721,uint256,uint256)._feeReceiver (OVR
LandRenting.sol#2012) lacks a zero-check on :
                - feeReceiver = _feeReceiver (OVRLandRenting.sol#2032)
OVRLandRenting.setFeeReceiver(address)._account (OVRLandRenting.sol#2076) lacks a zero-check on :
                - feeReceiver = _account (OVRLandRenting.sol#2080)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation
INFO:Detectors:
Variable 'ERC1967UpgradeUpgradeable._upgradeToAndCallUUPS(address,bytes,bool).slot (OVRLandRenting.sol#1804)' in ERC1967Upgrade
Upgradeable._upgradeToAndCallUUPS(address,bytes,bool) (OVRLandRenting.sol#1796-1811) potentially used before declaration: requi
re(bool,string)(slot == _IMPLEMENTATION_SLOT,ERC1967Upgrade: unsupported proxiableUUID) (OVRLandRenting.sol#1805)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables
INFO:Detectors:
Reentrancy in OVRLandRenting.activateNoRent(IOVRERC721,uint256,OVRLandRenting.NoRentParams) (OVRLandRenting.sol#2090-2109):
        External calls:
        - require(bool,string)(token.transferFrom(_msgSender(),feeReceiver,noRentPriceLand),Transfer failed) (OVRLandRenting.so
l#2098)
        - require(bool,string)(token.transferFrom(_msgSender(),feeReceiver,noRentPriceContainer),Transfer failed) (OVRLandRenti
ng.sol#2101)
        State variables written after the call(s):
        - noRentsEnd[_address][_nftId] = _now().add(noRentDuration) (OVRLandRenting.sol#2104)
        - noRentsParams[_address][_nftId] = _noRentParams (OVRLandRenting.sol#2105)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2
```



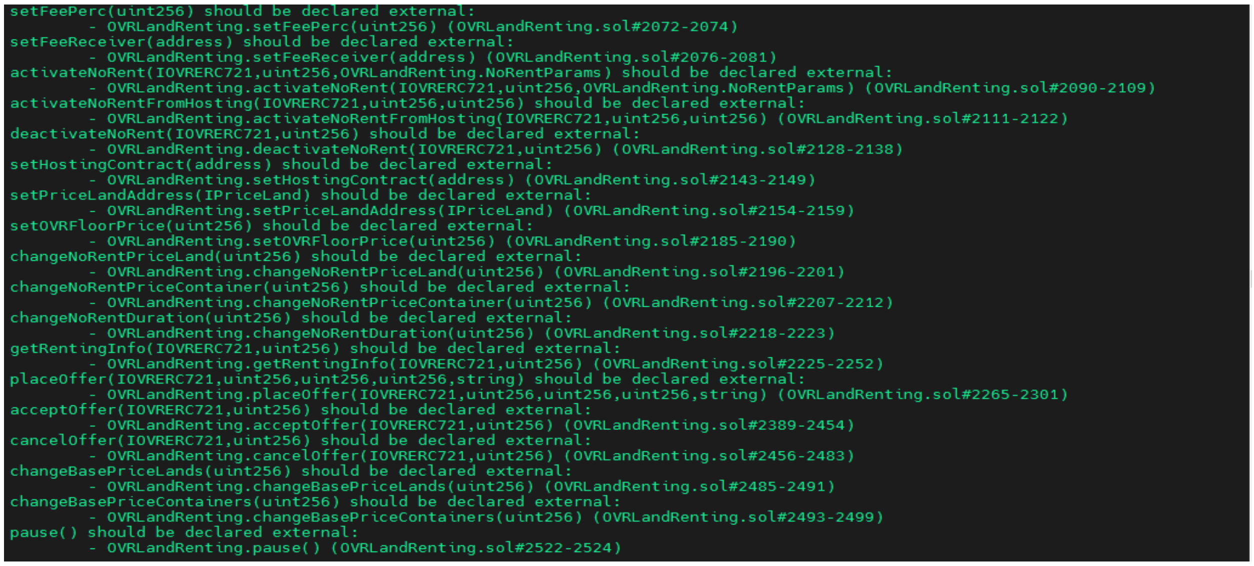
```
INFO:Detectors:
Reentrancy in OVRLandRenting.acceptOffer(IOVRERC721,uint256) (OVRLandRenting.sol#2389-2454):
        External calls:
        - success = OVRLandExperience.startTokenRenting(_address,_nftId,offers[_address][_nftId].from,_now(),offers[_address][_
nftId].months,offers[_address][_nftId].experienceUri) (OVRLandRenting.sol#2436)
        - token.transfer(_address.ownerOf(_nftId),tokenForOwner) (OVRLandRenting.sol#2445)
        - token.transfer(feeReceiver,offers[_address][_nftId].fee) (OVRLandRenting.sol#2446)
        Event emitted after the call(s):
        - OfferAccepted(address(_address),_nftId,_msgSender(),offers[_address][_nftId].from,offers[_address][_nftId].valuePerMo
nth,offers[_address][_nftId].months,_now()) (OVRLandRenting.sol#2448)
Reentrancy in OVRLandRenting.activateNoRent(IOVRERC721,uint256,OVRLandRenting.NoRentParams) (OVRLandRenting.sol#2090-2109):
        External calls:
        - require(bool,string)(token.transferFrom(_msgSender(),feeReceiver,noRentPriceLand),Transfer failed) (OVRLandRenting.so
l#2098)
        - require(bool,string)(token.transferFrom(_msgSender(),feeReceiver,noRentPriceContainer),Transfer failed) (OVRLandRenti
ng.sol#2101)
        Event emitted after the call(s):
        - NoRentActivated(address(_address),_nftId,_msgSender(),_now(),noRentsEnd[_address][_nftId],noRentsParams[_address][_nf
tId].pricePerMonth,noRentsParams[_address][_nftId].minMonths,noRentsParams[_address][_nftId].maxMonths) (OVRLandRenting.sol#210
8)
Reentrancy in OVRLandRenting.cancelOffer(IOVRERC721,uint256) (OVRLandRenting.sol#2456-2483):
        External calls:
        - repayPreviousOfferer(_address,_nftId) (OVRLandRenting.sol#2480)
                - require(bool,string)(token.transfer(from,paid),Insufficient contract balance) (OVRLandRenting.sol#2360)
        Event emitted after the call(s):
        - OfferCanceled(address(_address),_nftId,_msgSender(),_now()) (OVRLandRenting.sol#2482)
Reentrancy in OVRLandRenting.placeOfferInternal(IOVRERC721,uint256,uint256,uint256,string) (OVRLandRenting.sol#2303-2341):
        External calls:
        - repayPreviousOfferer(_nftAddress,_nftId) (OVRLandRenting.sol#2320)
                - require(bool,string)(token.transfer(from,paid),Insufficient contract balance) (OVRLandRenting.sol#2360)
        Event emitted after the call(s):
        - Overbid(address(_nftAddress),_nftId,_msgSender(),_amountPerMonth,_months,_now(),calculatedFees) (OVRLandRenting.sol#2
322)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3
```



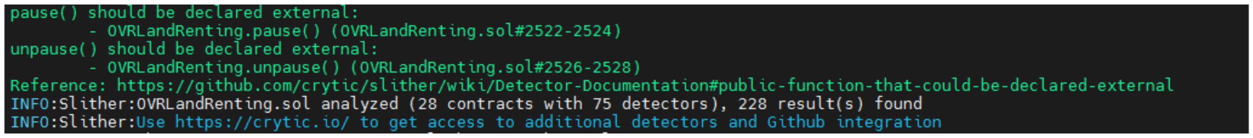
```
INFO:Detectors:
Redundant expression "this (OVRLandRenting.sol#634)" inContext (OVRLandRenting.sol#628-637)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements
INFO:Detectors:
Variable OVRLandRenting.activateNoRent(IOVRERC721,uint256,OVRLandRenting.NoRentParams)._noRentParams (OVRLandRenting.sol#2093)
 is too similar to OVRLandRenting.noRentsParams (OVRLandRenting.sol#1928)
Variable OVRLandRenting.getRentingInfo(IOVRERC721,uint256)._noRentParams (OVRLandRenting.sol#2230) is too similar to OVRLandRen
ting.noRentsParams (OVRLandRenting.sol#1928)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-are-too-similar
INFO:Detectors:
OVRLandRenting (OVRLandRenting.sol#1907-2535) does not implement functions:
        - UUPSUpgradeable._authorizeUpgrade(address) (OVRLandRenting.sol#1902)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unimplemented-functions
INFO:Detectors:
PausableUpgradeable.__gap (OVRLandRenting.sol#1144) is never used in OVRLandRenting (OVRLandRenting.sol#1907-2535)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variables
INFO:Detectors:
ERC20._decimals (OVRLandRenting.sol#699) should be constant
ERC20._name (OVRLandRenting.sol#697) should be constant
ERC20._symbol (OVRLandRenting.sol#698) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant
INFO:Detectors:
renounceOwnership() should be declared external:
        - Ownable.renounceOwnership() (OVRLandRenting.sol#671-674)
transferOwnership(address) should be declared external:
        - Ownable.transferOwnership(address) (OVRLandRenting.sol#680-684)
name() should be declared external:
        - ERC20.name() (OVRLandRenting.sol#721-723)
decimals() should be declared external:
        - ERC20.decimals() (OVRLandRenting.sol#728-730)
symbol() should be declared external:
        - ERC20.symbol() (OVRLandRenting.sol#735-737)
totalSupply() should be declared external:
        - ERC20.totalSupply() (OVRLandRenting.sol#742-744)
balanceOf(address) should be declared external:
        - ERC20.balanceOf(address) (OVRLandRenting.sol#749-751)
```



```
balanceOf(address) should be declared external:
        - ERC20.balanceOf(address) (OVRLandRenting.sol#749-751)
transfer(address,uint256) should be declared external:
        - ERC20.transfer(address,uint256) (OVRLandRenting.sol#761-764)
allowance(address,address) should be declared external:
        - ERC20.allowance(address,address) (OVRLandRenting.sol#769-771)
approve(address,uint256) should be declared external:
        - ERC20.approve(address,uint256) (OVRLandRenting.sol#780-783)
transferFrom(address,address,uint256) should be declared external:
        - ERC20.transferFrom(address,address,uint256) (OVRLandRenting.sol#797-809)
increaseAllowance(address,uint256) should be declared external:
        - ERC20.increaseAllowance(address,uint256) (OVRLandRenting.sol#823-826)
decreaseAllowance(address,uint256) should be declared external:
        - ERC20.decreaseAllowance(address,uint256) (OVRLandRenting.sol#842-849)
mint(uint256) should be declared external:
        - ERC20.mint(uint256) (OVRLandRenting.sol#859-862)
grantRole(bytes32,address) should be declared external:
        - AccessControlUpgradeable.grantRole(bytes32,address) (OVRLandRenting.sol#1596-1598)
revokeRole(bytes32,address) should be declared external:
        - AccessControlUpgradeable.revokeRole(bytes32,address) (OVRLandRenting.sol#1611-1613)
renounceRole(bytes32,address) should be declared external:
        - AccessControlUpgradeable.renounceRole(bytes32,address) (OVRLandRenting.sol#1631-1635)
initialize(address,address,address,address,address,uint256,uint256,IOVRERC721,uint256,uint256) should be declared external:
        - OVRLandRenting.initialize(address,address,address,address,address,uint256,uint256,IOVRERC721,uint256,uint256) (OVRLan
dRenting.sol#2007-2040)
addAdmin(address) should be declared external:
        - OVRLandRenting.addAdmin(address) (OVRLandRenting.sol#2042-2044)
removeAdmin(address) should be declared external:
        - OVRLandRenting.removeAdmin(address) (OVRLandRenting.sol#2046-2048)
setBasePriceChanger(address) should be declared external:
        - OVRLandRenting.setBasePriceChanger(address) (OVRLandRenting.sol#2050-2055)
removeBasePriceChanger(address) should be declared external:
        - OVRLandRenting.removeBasePriceChanger(address) (OVRLandRenting.sol#2057-2062)
setOVRLandExperienceAddress(address) should be declared external:
        - OVRLandRenting.setOVRLandExperienceAddress(address) (OVRLandRenting.sol#2064-2069)
```



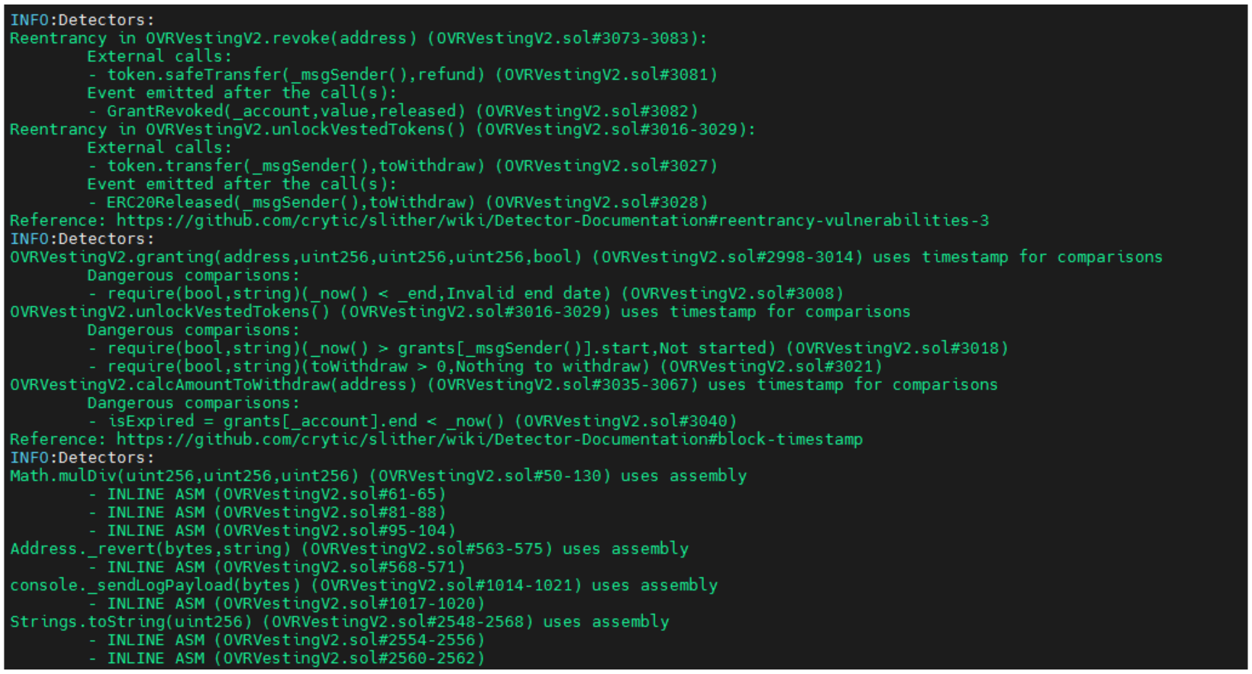
```
setFeePerc(uint256) should be declared external:
        - OVRLandRenting.setFeePerc(uint256) (OVRLandRenting.sol#2072-2074)
setFeeReceiver(address) should be declared external:
        - OVRLandRenting.setFeeReceiver(address) (OVRLandRenting.sol#2076-2081)
activateNoRent(IOVRERC721,uint256,OVRLandRenting.NoRentParams) should be declared external:
        - OVRLandRenting.activateNoRent(IOVRERC721,uint256,OVRLandRenting.NoRentParams) (OVRLandRenting.sol#2090-2109)
activateNoRentFromHosting(IOVRERC721,uint256,uint256) should be declared external:
        - OVRLandRenting.activateNoRentFromHosting(IOVRERC721,uint256,uint256) (OVRLandRenting.sol#2111-2122)
deactivateNoRent(IOVRERC721,uint256) should be declared external:
        - OVRLandRenting.deactivateNoRent(IOVRERC721,uint256) (OVRLandRenting.sol#2128-2138)
setHostingContract(address) should be declared external:
        - OVRLandRenting.setHostingContract(address) (OVRLandRenting.sol#2143-2149)
setPriceLandAddress(IPriceLand) should be declared external:
        - OVRLandRenting.setPriceLandAddress(IPriceLand) (OVRLandRenting.sol#2154-2159)
setOVRFloorPrice(uint256) should be declared external:
        - OVRLandRenting.setOVRFloorPrice(uint256) (OVRLandRenting.sol#2185-2190)
changeNoRentPriceLand(uint256) should be declared external:
        - OVRLandRenting.changeNoRentPriceLand(uint256) (OVRLandRenting.sol#2196-2201)
changeNoRentPriceContainer(uint256) should be declared external:
        - OVRLandRenting.changeNoRentPriceContainer(uint256) (OVRLandRenting.sol#2207-2212)
changeNoRentDuration(uint256) should be declared external:
        - OVRLandRenting.changeNoRentDuration(uint256) (OVRLandRenting.sol#2218-2223)
getRentingInfo(IOVRERC721,uint256) should be declared external:
        - OVRLandRenting.getRentingInfo(IOVRERC721,uint256) (OVRLandRenting.sol#2225-2252)
placeOffer(IOVRERC721,uint256,uint256,uint256,string) should be declared external:
        - OVRLandRenting.placeOffer(IOVRERC721,uint256,uint256,uint256,string) (OVRLandRenting.sol#2265-2301)
acceptOffer(IOVRERC721,uint256) should be declared external:
        - OVRLandRenting.acceptOffer(IOVRERC721,uint256) (OVRLandRenting.sol#2389-2454)
cancelOffer(IOVRERC721,uint256) should be declared external:
        - OVRLandRenting.cancelOffer(IOVRERC721,uint256) (OVRLandRenting.sol#2456-2483)
changeBasePriceLands(uint256) should be declared external:
        - OVRLandRenting.changeBasePriceLands(uint256) (OVRLandRenting.sol#2485-2491)
changeBasePriceContainers(uint256) should be declared external:
        - OVRLandRenting.changeBasePriceContainers(uint256) (OVRLandRenting.sol#2493-2499)
pause() should be declared external:
        - OVRLandRenting.pause() (OVRLandRenting.sol#2522-2524)
```



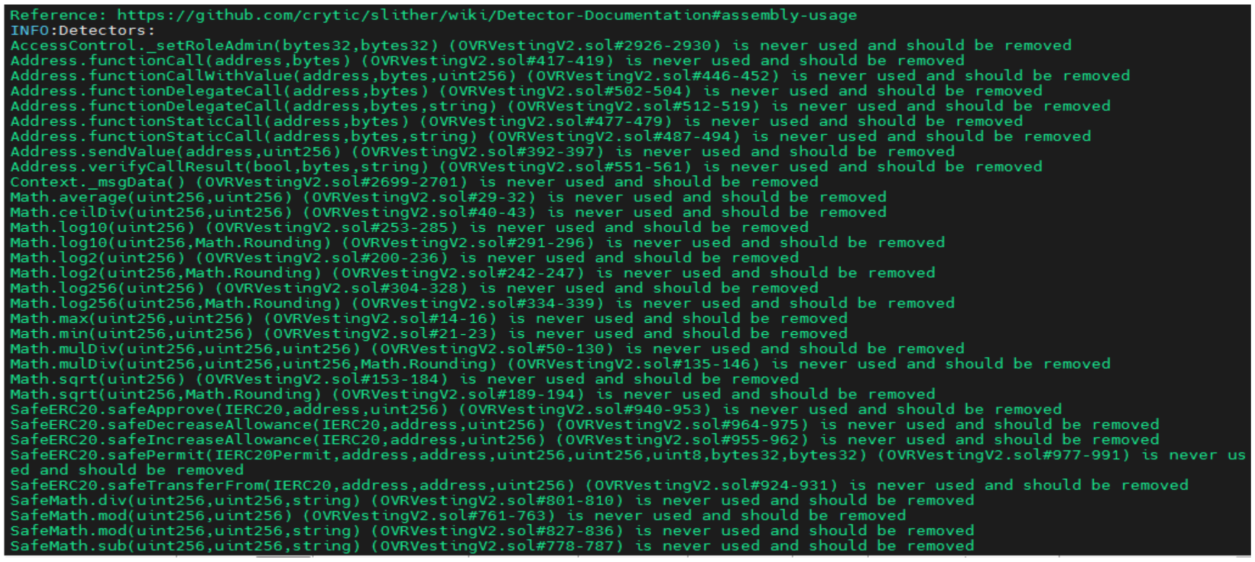
```
pause() should be declared external:
        - OVRLandRenting.pause() (OVRLandRenting.sol#2522-2524)
unpause() should be declared external:
        - OVRLandRenting.unpause() (OVRLandRenting.sol#2526-2528)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:OVRLandRenting.sol analyzed (28 contracts with 75 detectors), 228 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration
```

Code Flow Diagram - Slither Results Log

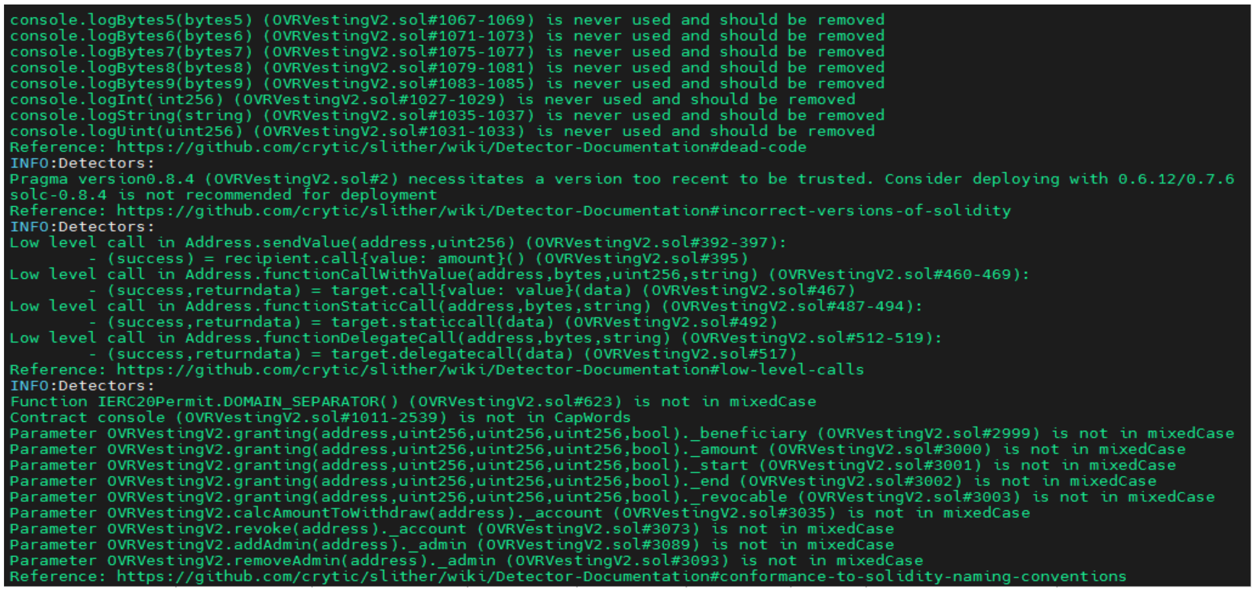
OVRVestingV2



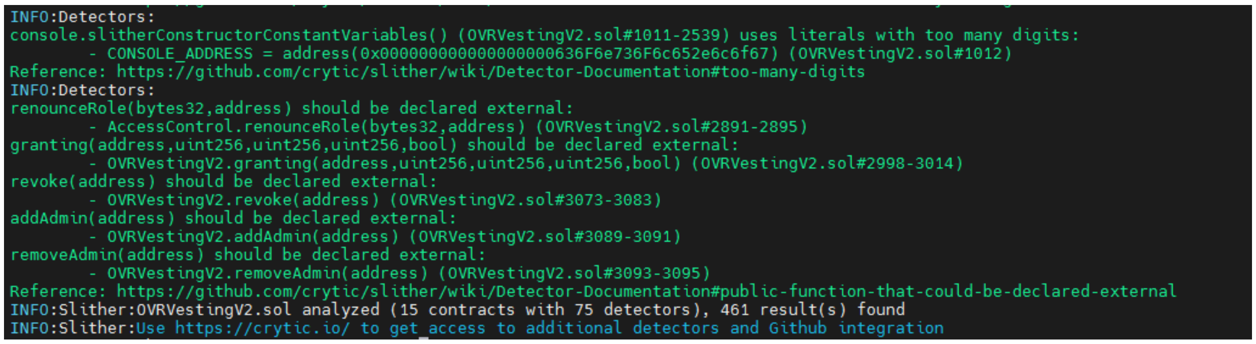
```
INFO:Detectors:
Reentrancy in OVRVestingV2.revoke(address) (OVRVestingV2.sol#3073-3083):
        External calls:
        - token.safeTransfer(_msgSender(),refund) (OVRVestingV2.sol#3081)
        Event emitted after the call(s):
        - GrantRevoked(_account,value,released) (OVRVestingV2.sol#3082)
Reentrancy in OVRVestingV2.unlockVestedTokens() (OVRVestingV2.sol#3016-3029):
        External calls:
        - token.transfer(_msgSender(),toWithdraw) (OVRVestingV2.sol#3027)
        Event emitted after the call(s):
        - ERC20Released(_msgSender(),toWithdraw) (OVRVestingV2.sol#3028)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3
INFO:Detectors:
OVRVestingV2.granting(address,uint256,uint256,uint256,bool) (OVRVestingV2.sol#2998-3014) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(_now() < _end,Invalid end date) (OVRVestingV2.sol#3008)
OVRVestingV2.unlockVestedTokens() (OVRVestingV2.sol#3016-3029) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(_now() > grants[_msgSender()].start,Not started) (OVRVestingV2.sol#3018)
        - require(bool,string)(toWithdraw > 0,Nothing to withdraw) (OVRVestingV2.sol#3021)
OVRVestingV2.calcAmountToWithdraw(address) (OVRVestingV2.sol#3035-3067) uses timestamp for comparisons
        Dangerous comparisons:
        - isExpired = grants[_account].end < _now() (OVRVestingV2.sol#3040)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp
INFO:Detectors:
Math.mulDiv(uint256,uint256,uint256) (OVRVestingV2.sol#50-130) uses assembly
        - INLINE ASM (OVRVestingV2.sol#61-65)
        - INLINE ASM (OVRVestingV2.sol#81-88)
        - INLINE ASM (OVRVestingV2.sol#95-104)
Address._revert(bytes,string) (OVRVestingV2.sol#563-575) uses assembly
        - INLINE ASM (OVRVestingV2.sol#568-571)
console._sendLogPayload(bytes) (OVRVestingV2.sol#1014-1021) uses assembly
        - INLINE ASM (OVRVestingV2.sol#1017-1020)
Strings.toString(uint256) (OVRVestingV2.sol#2548-2568) uses assembly
        - INLINE ASM (OVRVestingV2.sol#2554-2556)
        - INLINE ASM (OVRVestingV2.sol#2560-2562)
```



```
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
INFO:Detectors:
AccessControl._setRoleAdmin(bytes32,bytes32) (OVRVestingV2.sol#2926-2930) is never used and should be removed
Address.functionCall(address,bytes) (OVRVestingV2.sol#417-419) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (OVRVestingV2.sol#446-452) is never used and should be removed
Address.functionDelegateCall(address,bytes) (OVRVestingV2.sol#502-504) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (OVRVestingV2.sol#512-519) is never used and should be removed
Address.functionStaticCall(address,bytes) (OVRVestingV2.sol#477-479) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (OVRVestingV2.sol#487-494) is never used and should be removed
Address.sendValue(address,uint256) (OVRVestingV2.sol#392-397) is never used and should be removed
Address.verifyCallResult(bool,bytes,string) (OVRVestingV2.sol#551-561) is never used and should be removed
Context._msgData() (OVRVestingV2.sol#2699-2701) is never used and should be removed
Math.average(uint256,uint256) (OVRVestingV2.sol#29-32) is never used and should be removed
Math.ceilDiv(uint256,uint256) (OVRVestingV2.sol#40-43) is never used and should be removed
Math.log10(uint256) (OVRVestingV2.sol#253-285) is never used and should be removed
Math.log10(uint256,Math.Rounding) (OVRVestingV2.sol#291-296) is never used and should be removed
Math.log2(uint256) (OVRVestingV2.sol#200-236) is never used and should be removed
Math.log2(uint256,Math.Rounding) (OVRVestingV2.sol#242-247) is never used and should be removed
Math.log256(uint256) (OVRVestingV2.sol#304-328) is never used and should be removed
Math.log256(uint256,Math.Rounding) (OVRVestingV2.sol#334-339) is never used and should be removed
Math.max(uint256,uint256) (OVRVestingV2.sol#14-16) is never used and should be removed
Math.min(uint256,uint256) (OVRVestingV2.sol#21-23) is never used and should be removed
Math.mulDiv(uint256,uint256,uint256) (OVRVestingV2.sol#50-130) is never used and should be removed
Math.mulDiv(uint256,uint256,uint256,Math.Rounding) (OVRVestingV2.sol#135-146) is never used and should be removed
Math.sqrt(uint256) (OVRVestingV2.sol#153-184) is never used and should be removed
Math.sqrt(uint256,Math.Rounding) (OVRVestingV2.sol#189-194) is never used and should be removed
SafeERC20.safeApprove(IERC20,address,uint256) (OVRVestingV2.sol#940-953) is never used and should be removed
SafeERC20.safeDecreaseAllowance(IERC20,address,uint256) (OVRVestingV2.sol#964-975) is never used and should be removed
SafeERC20.safeIncreaseAllowance(IERC20,address,uint256) (OVRVestingV2.sol#955-962) is never used and should be removed
SafeERC20.safePermit(IERC20Permit,address,address,uint256,uint256,uint8,bytes32,bytes32) (OVRVestingV2.sol#977-991) is never us
ed and should be removed
SafeERC20.safeTransferFrom(IERC20,address,address,uint256) (OVRVestingV2.sol#924-931) is never used and should be removed
SafeMath.div(uint256,uint256,string) (OVRVestingV2.sol#801-810) is never used and should be removed
SafeMath.mod(uint256,uint256) (OVRVestingV2.sol#761-763) is never used and should be removed
SafeMath.mod(uint256,uint256,string) (OVRVestingV2.sol#827-836) is never used and should be removed
SafeMath.sub(uint256,uint256,string) (OVRVestingV2.sol#778-787) is never used and should be removed
```



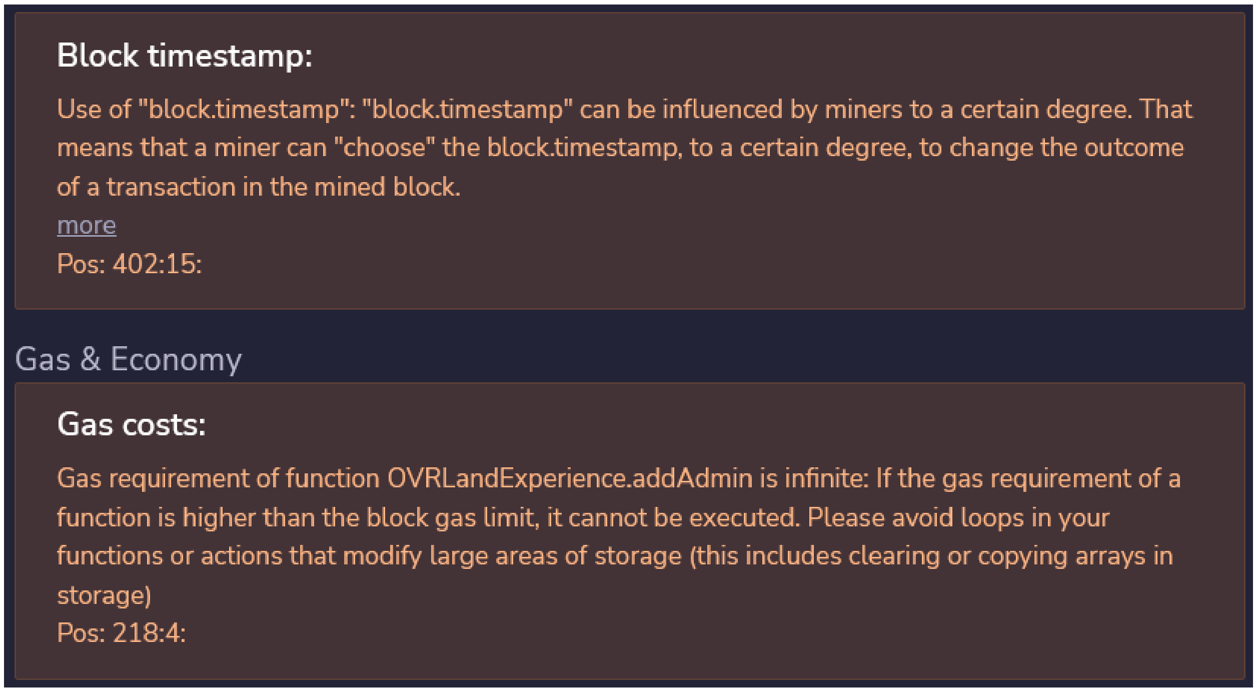
```
console.logBytes5(bytes5) (OVRVestingV2.sol#1067-1069) is never used and should be removed
console.logBytes6(bytes6) (OVRVestingV2.sol#1071-1073) is never used and should be removed
console.logBytes7(bytes7) (OVRVestingV2.sol#1075-1077) is never used and should be removed
console.logBytes8(bytes8) (OVRVestingV2.sol#1079-1081) is never used and should be removed
console.logBytes9(bytes9) (OVRVestingV2.sol#1083-1085) is never used and should be removed
console.logInt(int256) (OVRVestingV2.sol#1027-1029) is never used and should be removed
console.logString(string) (OVRVestingV2.sol#1035-1037) is never used and should be removed
console.logUint(uint256) (OVRVestingV2.sol#1031-1033) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
Pragma version0.8.4 (OVRVestingV2.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
solc-0.8.4 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Low level call in Address.sendValue(address,uint256) (OVRVestingV2.sol#392-397):
        - (success) = recipient.call{value: amount}() (OVRVestingV2.sol#395)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (OVRVestingV2.sol#460-469):
        - (success,returndata) = target.call{value: value}(data) (OVRVestingV2.sol#467)
Low level call in Address.functionStaticCall(address,bytes,string) (OVRVestingV2.sol#487-494):
        - (success,returndata) = target.staticcall(data) (OVRVestingV2.sol#492)
Low level call in Address.functionDelegateCall(address,bytes,string) (OVRVestingV2.sol#512-519):
        - (success,returndata) = target.delegatecall(data) (OVRVestingV2.sol#517)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
INFO:Detectors:
Function IERC20Permit.DOMAIN_SEPARATOR() (OVRVestingV2.sol#623) is not in mixedCase
Contract console (OVRVestingV2.sol#1011-2539) is not in CapWords
Parameter OVRVestingV2.granting(address,uint256,uint256,uint256,bool)._beneficiary (OVRVestingV2.sol#2999) is not in mixedCase
Parameter OVRVestingV2.granting(address,uint256,uint256,uint256,bool)._amount (OVRVestingV2.sol#3000) is not in mixedCase
Parameter OVRVestingV2.granting(address,uint256,uint256,uint256,bool)._start (OVRVestingV2.sol#3001) is not in mixedCase
Parameter OVRVestingV2.granting(address,uint256,uint256,uint256,bool)._end (OVRVestingV2.sol#3002) is not in mixedCase
Parameter OVRVestingV2.granting(address,uint256,uint256,uint256,bool)._revocable (OVRVestingV2.sol#3003) is not in mixedCase
Parameter OVRVestingV2.calcAmountToWithdraw(address)._account (OVRVestingV2.sol#3035) is not in mixedCase
Parameter OVRVestingV2.revoke(address)._account (OVRVestingV2.sol#3073) is not in mixedCase
Parameter OVRVestingV2.addAdmin(address)._admin (OVRVestingV2.sol#3089) is not in mixedCase
Parameter OVRVestingV2.removeAdmin(address)._admin (OVRVestingV2.sol#3093) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
```



```
INFO:Detectors:
console.slitherConstructorConstantVariables() (OVRVestingV2.sol#1011-2539) uses literals with too many digits:
        - CONSOLE_ADDRESS = address(0x000000000000000000636F6e736F6c652e6c6f67) (OVRVestingV2.sol#1012)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits
INFO:Detectors:
renounceRole(bytes32,address) should be declared external:
        - AccessControl.renounceRole(bytes32,address) (OVRVestingV2.sol#2891-2895)
granting(address,uint256,uint256,uint256,bool) should be declared external:
        - OVRVestingV2.granting(address,uint256,uint256,uint256,bool) (OVRVestingV2.sol#2998-3014)
revoke(address) should be declared external:
        - OVRVestingV2.revoke(address) (OVRVestingV2.sol#3073-3083)
addAdmin(address) should be declared external:
        - OVRVestingV2.addAdmin(address) (OVRVestingV2.sol#3089-3091)
removeAdmin(address) should be declared external:
        - OVRVestingV2.removeAdmin(address) (OVRVestingV2.sol#3093-3095)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
INFO:Slither:OVRVestingV2.sol analyzed (15 contracts with 75 detectors), 461 result(s) found
INFO:Slither:Use https://crytic.io/ to get access to additional detectors and Github integration
```

Solidity Static Analysis

OVRLandExperience



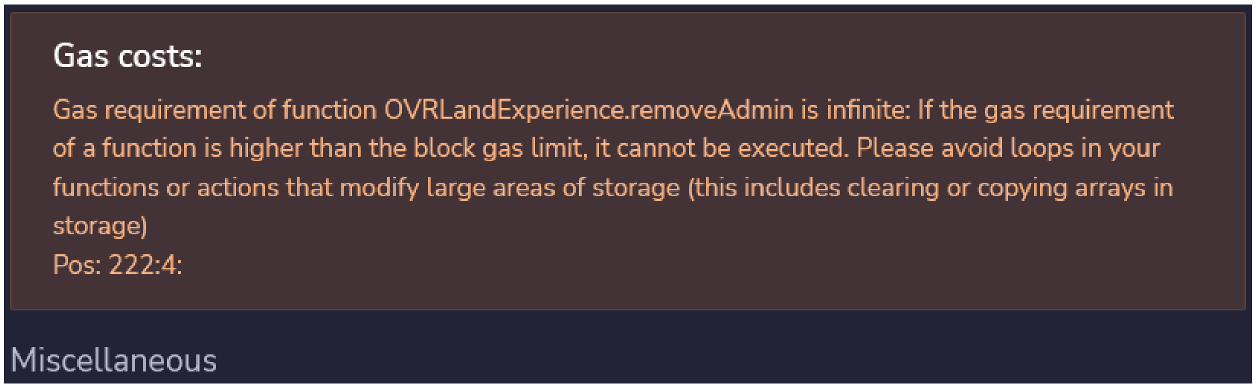
Block timestamp:

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: 402:15:

Gas & Economy

Gas costs:

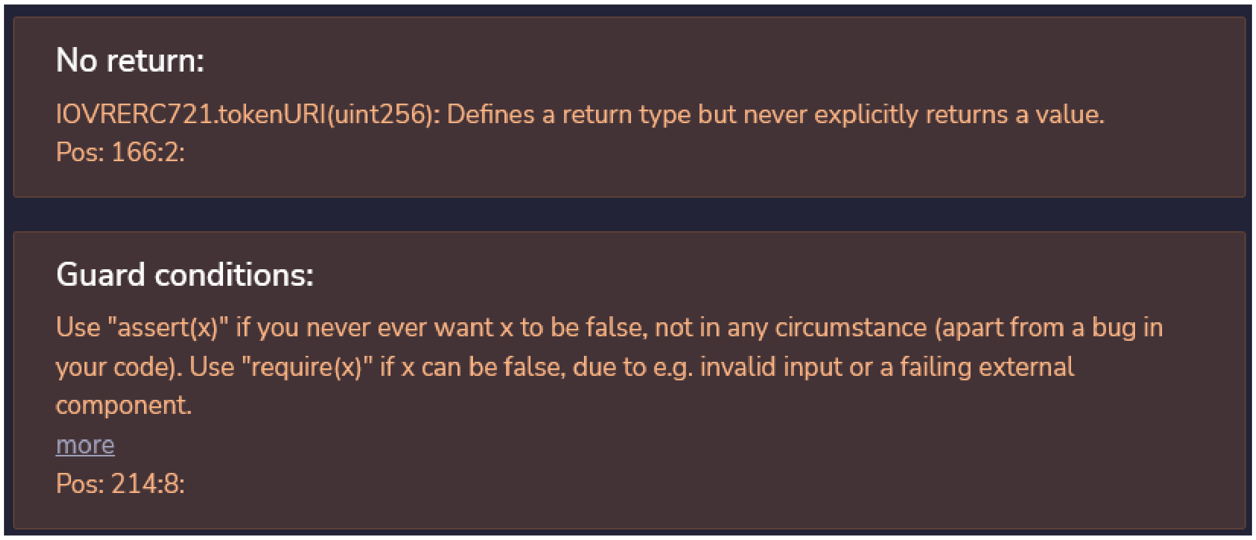
Gas requirement of function OVRLandExperience.addAdmin is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)
Pos: 218:4:



Gas costs:

Gas requirement of function OVRLandExperience.removeAdmin 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: 222:4:

Miscellaneous

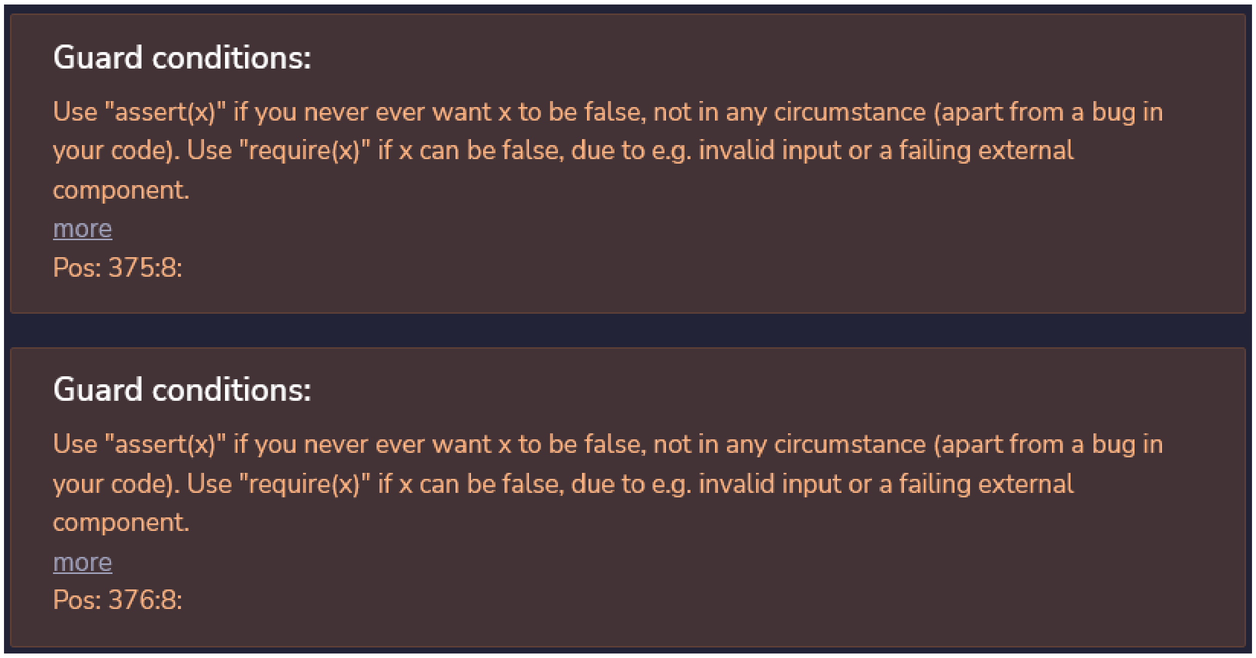


No return:

IOVRERC721.tokenURI(uint256): Defines a return type but never explicitly returns a value.
Pos: 166:2:

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.
more
Pos: 214:8:



Guard conditions:

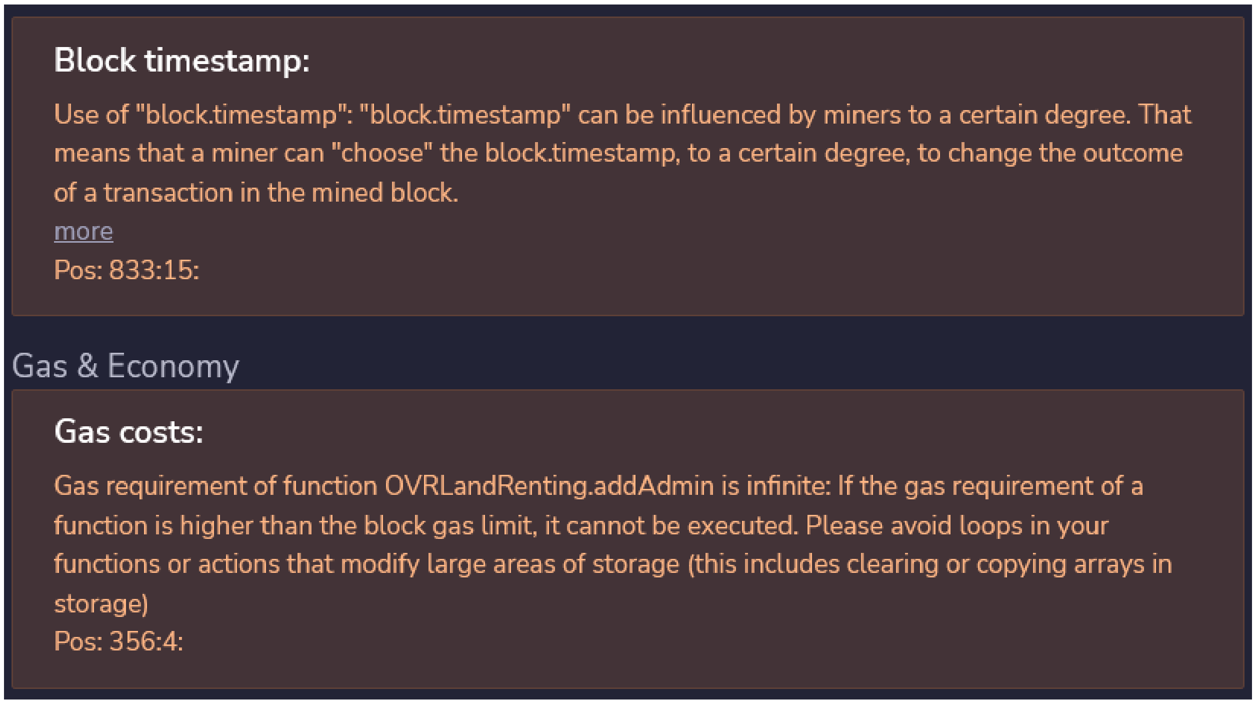
Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.
more
Pos: 375:8:

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.
more
Pos: 376:8:

Solidity Static Analysis

OVRLandRenting



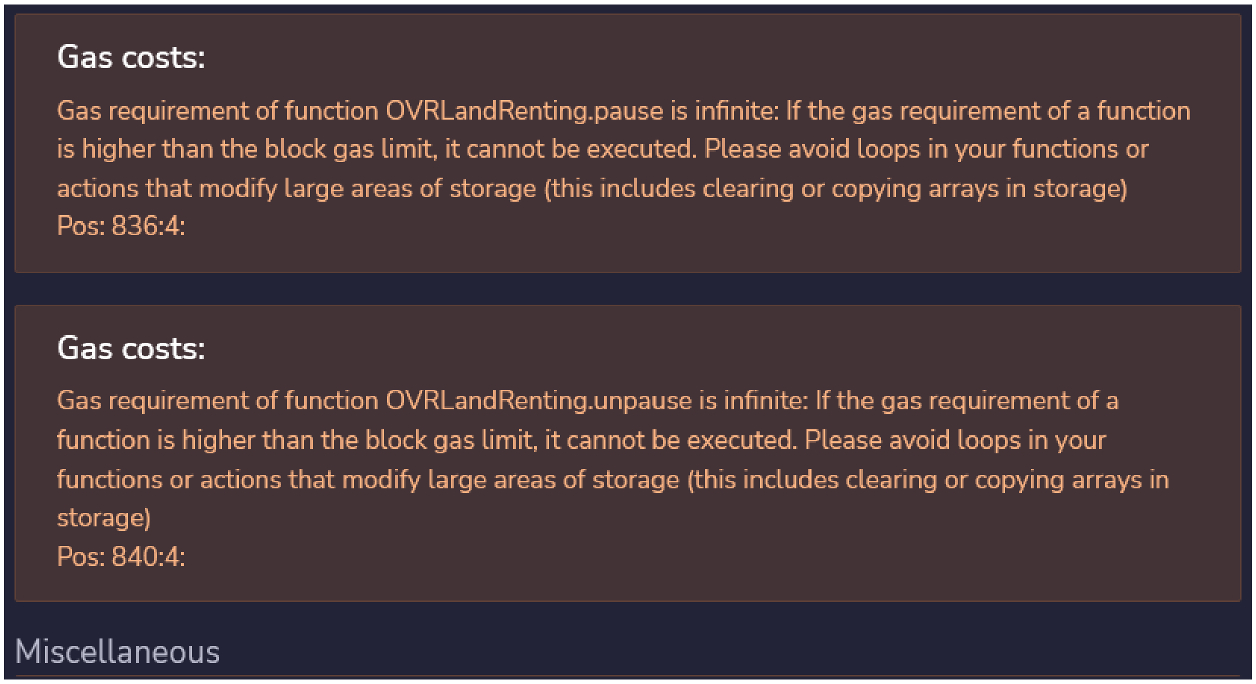
Block timestamp:

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: 833:15:

Gas & Economy

Gas costs:

Gas requirement of function OVRLandRenting.addAdmin 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: 356:4:



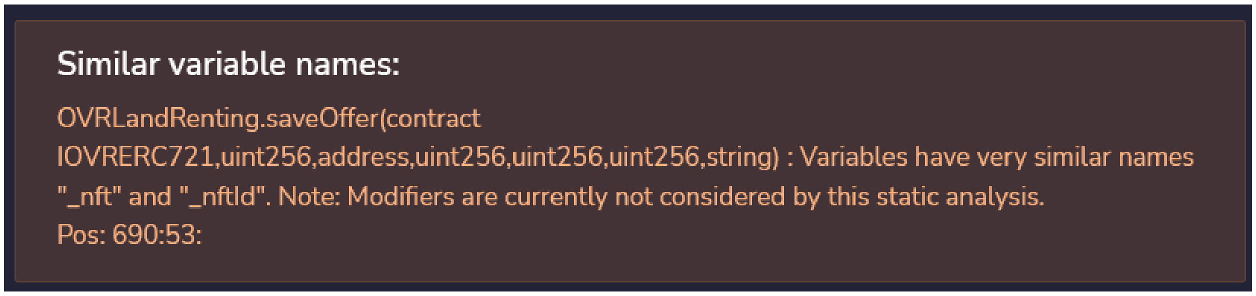
Gas costs:

Gas requirement of function OVRLandRenting.pause 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: 836:4:

Gas costs:

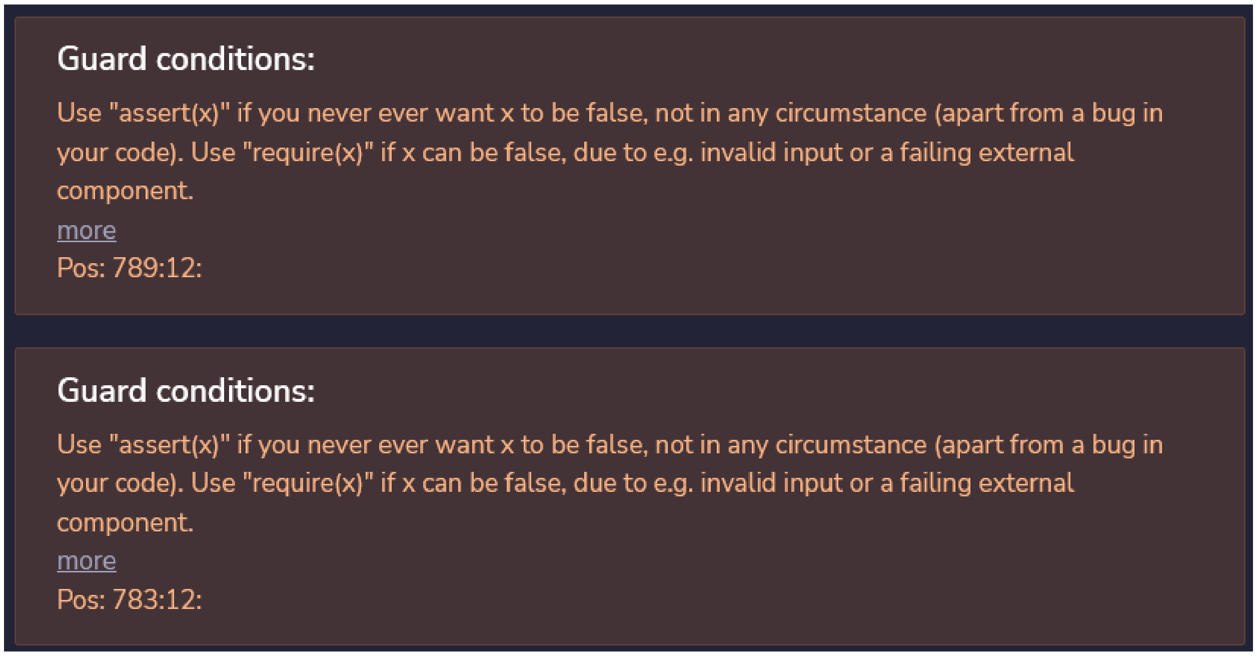
Gas requirement of function OVRLandRenting.unpause is infinite: If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)
Pos: 840:4:

Miscellaneous



Similar variable names:

OVRLandRenting.saveOffer(contract IOVRERC721,uint256,address,uint256,uint256,uint256,string) : Variables have very similar names "_nft" and "_nftId". Note: Modifiers are currently not considered by this static analysis.
Pos: 690:53:

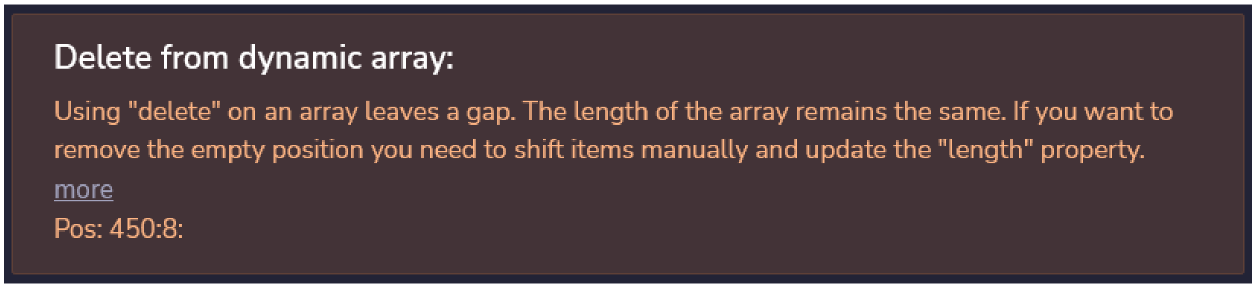


Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.
more
Pos: 789:12:

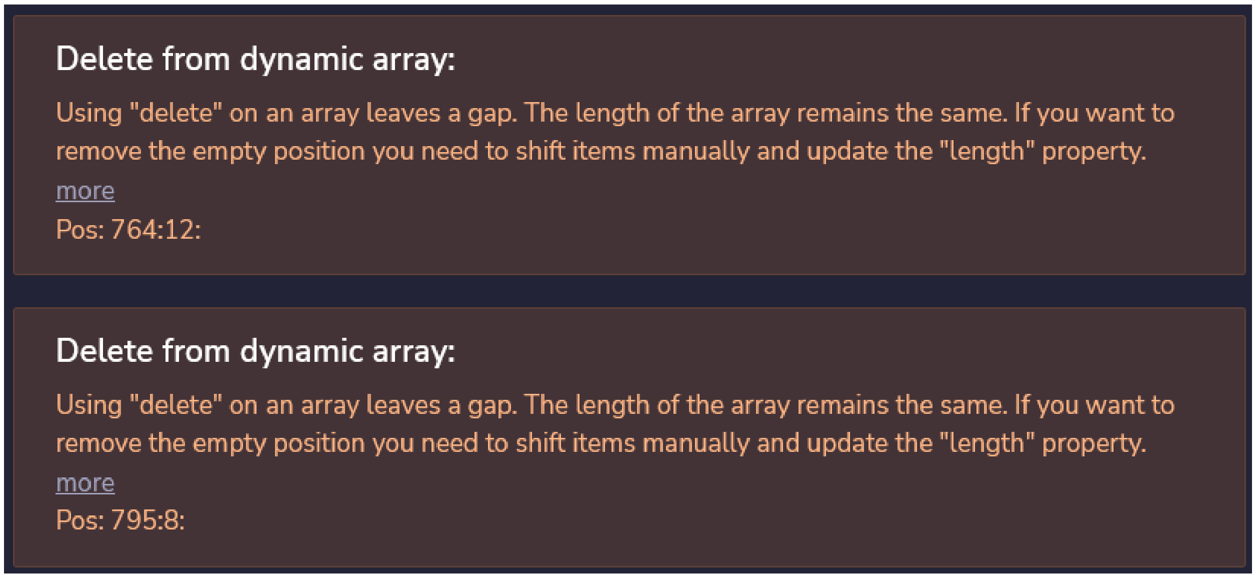
Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.
more
Pos: 783:12:



Delete from dynamic array:

Using "delete" on an array leaves a gap. The length of the array remains the same. If you want to remove the empty position you need to shift items manually and update the "length" property.
more
Pos: 450:8:



Delete from dynamic array:

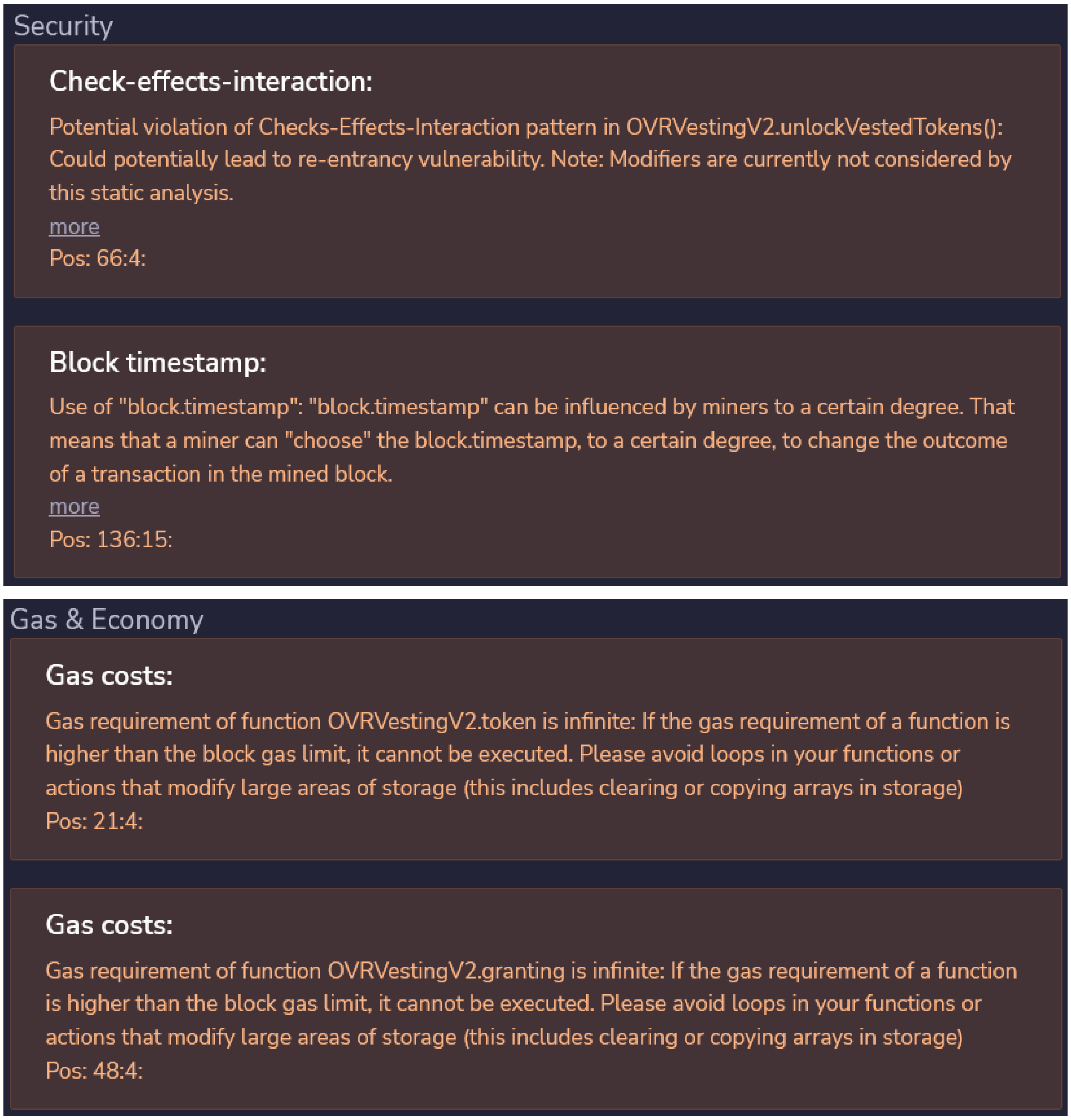
Using "delete" on an array leaves a gap. The length of the array remains the same. If you want to remove the empty position you need to shift items manually and update the "length" property.
more
Pos: 764:12:

Delete from dynamic array:

Using "delete" on an array leaves a gap. The length of the array remains the same. If you want to remove the empty position you need to shift items manually and update the "length" property.
more
Pos: 795:8:

Solidity Static Analysis

OVRVestingV2



Security

Check-effects-interaction:

Potential violation of Checks-Effects-Interaction pattern in OVRVestingV2.unlockVestedTokens(): Could potentially lead to re-entrancy vulnerability. Note: Modifiers are currently not considered by this static analysis.
more
Pos: 66:4:

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: 136:15:

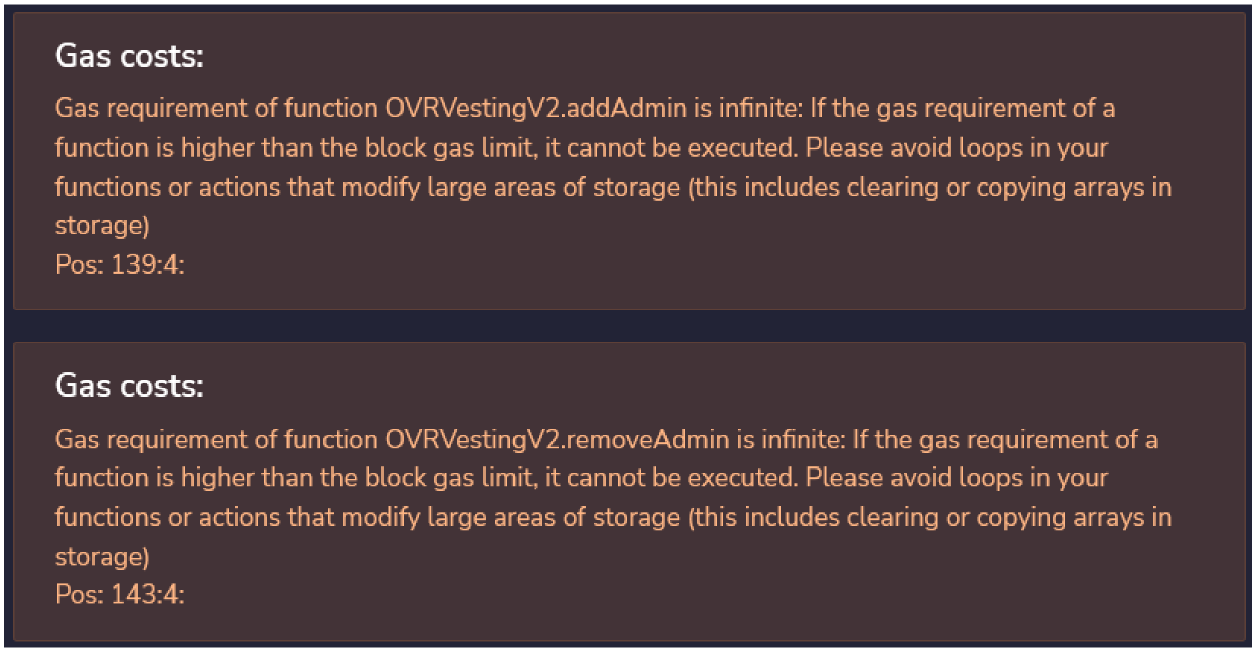
Gas & Economy

Gas costs:

Gas requirement of function OVRVestingV2.token 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: 21:4:

Gas costs:

Gas requirement of function OVRVestingV2.granting 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: 48:4:

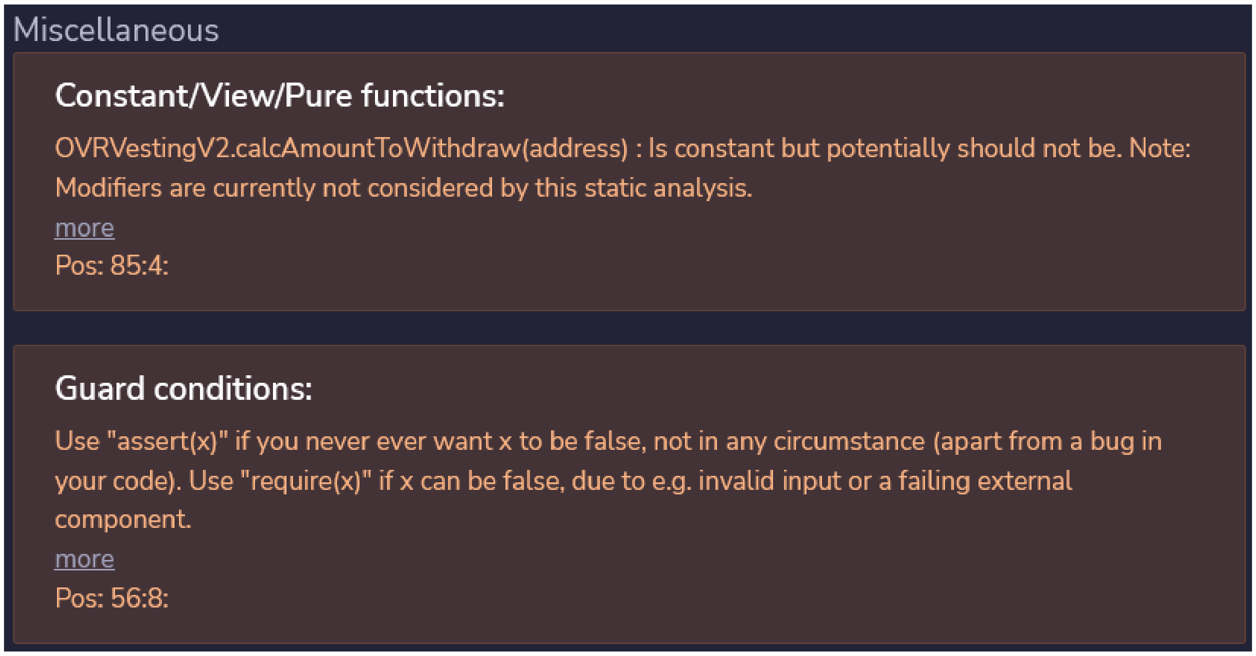


Gas costs:

Gas requirement of function OVRVestingV2.addAdmin 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: 139:4:

Gas costs:

Gas requirement of function OVRVestingV2.removeAdmin 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: 143:4:



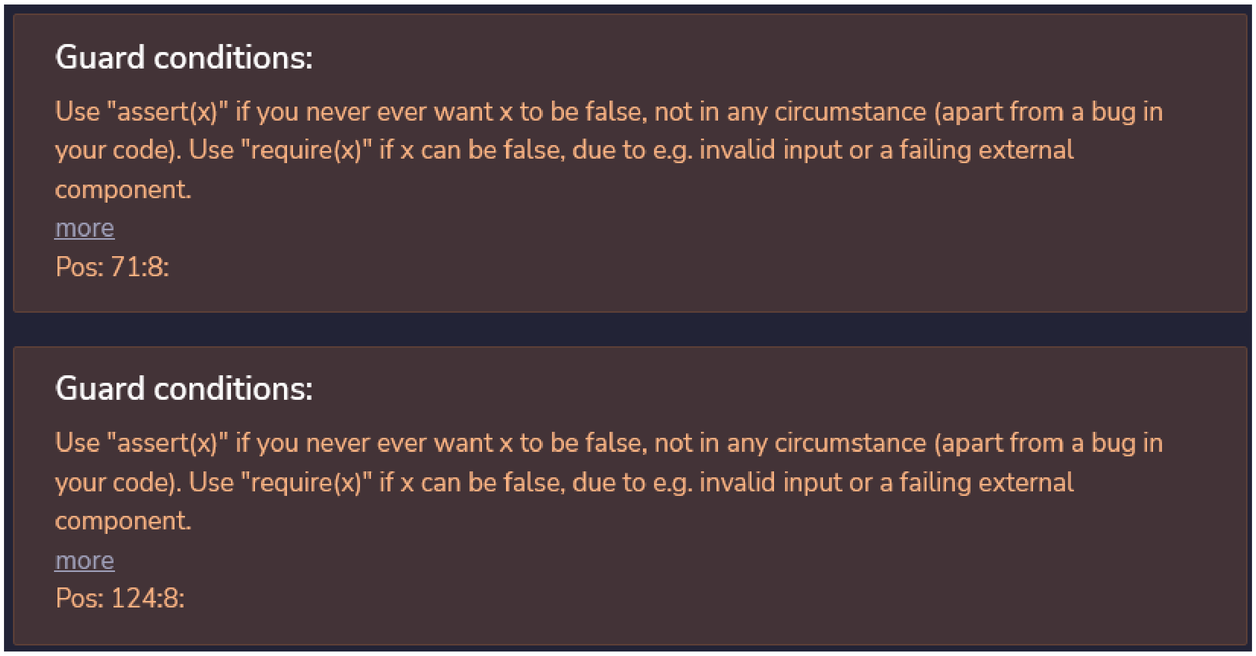
Miscellaneous

Constant/View/Pure functions:

OVRVestingV2.calcAmountToWithdraw(address) : Is constant but potentially should not be. Note: Modifiers are currently not considered by this static analysis.
more
Pos: 85:4:

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.
more
Pos: 56:8:

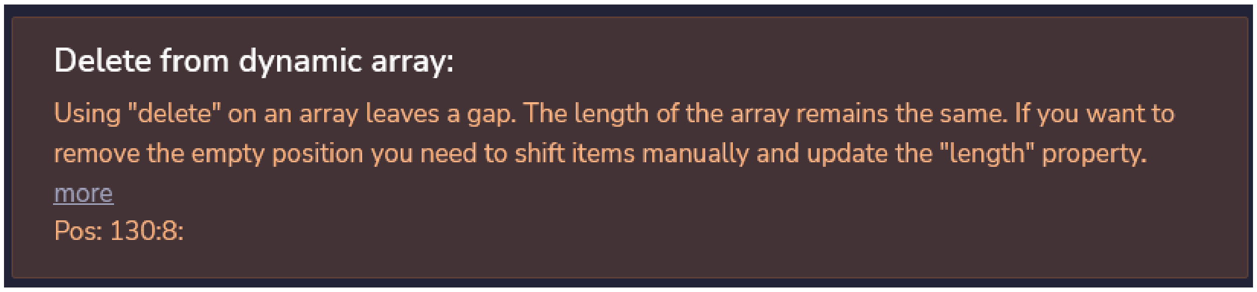


Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.
more
Pos: 71:8:

Guard conditions:

Use "assert(x)" if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use "require(x)" if x can be false, due to e.g. invalid input or a failing external component.
more
Pos: 124:8:



Delete from dynamic array:

Using "delete" on an array leaves a gap. The length of the array remains the same. If you want to remove the empty position you need to shift items manually and update the "length" property.
more
Pos: 130:8:

Severity Definitions

Risk Level	Description
Critical	Critical vulnerabilities are usually straightforward to exploit and can lead to lost tokens etc.
High	High level vulnerabilities are difficult to exploit; however, they also have a significant impact on smart contract execution, e.g. public access to crucial functions.
Medium	Medium level vulnerabilities are important to fix; however, they cannot lead to lost tokens.
Low	Low level vulnerabilities are most related to outdated, unused etc. These code snippets cannot have a significant impact on execution.
Lowest Code Style/ Best Practice	Lowest level vulnerabilities, code style violations and information statements cannot 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:

No low severity vulnerabilities were found.

Very Low:

No very low severity vulnerabilities were found.

Conclusion

We were given a contract file and have used all possible tests based on the given object. The contract is written systematically, so it is ready to go for production.

We have used all the latest static tools and manual observations to cover maximum possible test cases to scan everything.

The security state of the reviewed contract is now “well-secured”.

Note For Contract Users

There are several owner only functions. Those can be called by the owner's wallet only. So, if the owner's wallet is compromised, then it carries the risk of the contract becoming vulnerable.

Owner has full control over the smart contract. Thus, technical auditing does not guarantee the project's ethical side.

Please do your due diligence before investing. Our audit report is never an investment advice.

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 analyse 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 scrutinised 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

RD Auditors Disclaimer

The smart contracts given for audit have been analysed 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).

Because the total number of test cases are unlimited, the audit makes no statements or warranties on the 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 recommend proceeding with several independent audits and a public bug bounty program to ensure security of smart contracts.

Technical Disclaimer

Smart contracts are deployed and executed on the blockchain. 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.

RD
AUDITORS

Email: info@rdauditors.com

Website: www.rdauditors.com