4th05

Liquid RON

Security Review Report

4 February 2025

Security Review Report

4th05

February 4, 2025

Table of Contents

- Table of Contents
- Protocol Summary
- Disclaimer
- Risk Classification
- Overview
- Scope
- Issues found
- Findings
  - [L1] Functions reverted because of the big length of the returned arrays

Protocol Summary

Liquid Ron is a Ronin staking protocol that automates user staking actions. Deposit RON, get liquid RON, a token representing your stake in the validation process of the Ronin Network. Liquid RON stakes and harvests rewards automatically, auto compounding your rewards and ensuring the best yield possible.

Disclaimer

A smart contract security review can never verify the complete absence of vulnerabilities. This is a time, resource and expertise bound effort where I try to find as many vulnerabilities as possible. I can not guarantee 100% security after the review or even if the review will find any problems with your smart contracts. Subsequent security reviews, bug bounty programs and on-chain monitoring are strongly recommended.

Risk Classification

		Impact		
		High	Medium	Low
	High	H	H/M	M
Likelihood	Medium	H/M	M	M/L
	Low	M	M/L	L

Overview

Contest platform	Code4rena
LOC	386
Language	Solidity
Commit	e4b0b7c256bb2fe73b4a9c945415c3dcc935b61d
Previous audits	4naly3er, Slither

Scope

- /src/ValidatorTracker.sol
- /src/RonHelper.sol
- /src/Pausable.sol
- /src/LiquidRon.sol
- /src/LiquidProxy.sol
- /src/Escrow.sol

Issues found

Severity	Number of issues found
High	0
Medium	0
Low	1
Info	0

Findings

[L1] Functions reverted because of the big length of the returned arrays

Relevant GitHub Links

https://github.com/code-423n4/2025-01-liquid-ron/blob/main/src/LiquidRon.sol#L227

https://github.com/code-423n4/2025-01-liquid-ron/blob/main/src/LiquidRon.sol#L263

https://github.com/code-423n4/2025-01-liquid-ron/blob/main/src/LiquidRon.sol#L277

https://github.com/code-423n4/2025-01-liquid-ron/blob/main/src/LiquidRon.sol#L409

https://github.com/code-423n4/2025-01-liquid-ron/blob/main/src/ValidatorTracker.sol#L24

https://github.com/code-423n4/2025-01-liquid-ron/blob/main/src/ValidatorTracker.sol#L30

Finding description and impact

Several functions in the `LiquidRonin` and the `ValidatorTracker` contracts return arrays that could potentially be big enough to cause `OOG` issues.

```
for (uint256 j = 0; j < proxies.length; j++) {
                rewards[j] = IRoninValidator(roninStaking).getReward(vali, proxies[j]);
                valis[j] = vali;
            }
@>            uint256[] memory stakingTotals = IRoninValidator(roninStaking).getManyStakingAmounts(valis, proxies);
            bool canPrune = true;
            for (uint256 j = 0; j < proxies.length; j++)
                if (rewards[j] != 0 || stakingTotals[j] != 0) {
                    canPrune = false;
                    break;
                }
```

```
for (uint256 i = 0; i < _consensusAddrs.length; i++) users[i] = user;
@>        uint256[] memory stakedAmounts = IRoninValidator(roninStaking).getManyStakingAmounts(_consensusAddrs, users);
        for (uint256 i = 0; i < stakedAmounts.length; i++) totalStaked += stakedAmounts[i];
        return totalStaked;
```

```
@>    function getValidators() external view returns (address[] memory) {
        return validators;
    }

    /// @dev Get the list of validators, internal function
    /// @return validators The list of validators
@>    function _getValidators() internal view returns (address[] memory) {
        return validators;
    }
```

All these returned arrays will keep growing overtime by design. However, it has been not put in place any measure to avoid the `OOG` risk which would cause all the aforementioned functions to revert generating this way a `DOS`.

Recommended mitigation steps

Change these functions limiting the size of the array they can **`return`** by using `array indexes`.