ROCKETPOOL

Houston Hotfix

Smart Contract Security Review

Version: 2.0

September, 2024

Contents

Introduction **2**
- Disclaimer . . . 2
- Document Structure . . . 2
- Overview . . . 2

Security Assessment Summary **3**
- Scope . . . 3
- Approach . . . 3
- Coverage Limitations . . . 3
- Findings Summary . . . 3

Detailed Findings **4**

Summary of Findings **5**
- Inconsistent Value On Estimated Block Per Day . . . 6
- Miscellaneous General Comments . . . 7

A Vulnerability Severity Classification **8**

Introduction

Sigma Prime was commercially engaged to perform a time-boxed security review of the Rocketpool smart contract changes in scope. The review focused solely on the security aspects of the Solidity implementation of the contract, though general recommendations and informational comments are also provided.

Disclaimer

Sigma Prime makes all effort but holds no responsibility for the findings of this security review. Sigma Prime does not provide any guarantees relating to the function of the smart contract. Sigma Prime makes no judgements on, or provides any security review, regarding the underlying business model or the individuals involved in the project.

Document Structure

The first section provides an overview of the functionality of the Rocketpool smart contract changes contained within the scope of the security review. A summary followed by a detailed review of the discovered vulnerabilities is then given which assigns each vulnerability a severity rating (see Vulnerability Severity Classification), an *open/closed/resolved* status and a recommendation. Additionally, findings which do not have direct security implications (but are potentially of interest) are marked as *informational*.

The appendix provides additional documentation, including the severity matrix used to classify vulnerabilities within the Rocketpool smart contracts.

Overview

This assessment focused on reviewing several changes made to existing features based on community feedback and external reports received after Houston launch.

The changes include bug fixes related to legacy minipools, onchain voting, improvements to default values and DAO parameters, as well as code refactoring and miscellaneous housekeeping tasks.

Security Assessment Summary

Scope

This review was conducted on the files hosted on the Rocketpool repository and were assessed at commit 6f07f2a.

The scope of the review was strictly limited to the following diff: fb52ec9...6f07f2.

Note: third party libraries and dependencies, such as OpenZeppelin, were excluded from the scope of this assessment.

Approach

The manual review focused on identifying issues associated with the business logic implementation of the contracts. This includes their internal interactions, intended functionality and correct implementation with respect to the underlying functionality of the Ethereum Virtual Machine (for example, verifying correct storage/memory layout).

Additionally, the manual review process focused on identifying vulnerabilities related to known Solidity anti-patterns and attack vectors, such as re-entrancy, front-running, integer overflow/underflow and correct visibility specifiers.

For a more detailed, but non-exhaustive list of examined vectors, see [1, 2].

To support this review, the testing team also utilised the following automated testing tools:

- Mythril: `https://github.com/ConsenSys/mythril`
- Slither: `https://github.com/trailofbits/slither`
- Surya: `https://github.com/ConsenSys/surya`
- Aderyn: `https://github.com/Cyfrin/aderyn`

Output for these automated tools is available upon request.

Coverage Limitations

Due to a time-boxed nature of this review, all documented vulnerabilities reflect best effort within the allotted, limited engagement time. As such, Sigma Prime recommends to further investigate areas of the code, and any related functionality, where majority of critical and high risk vulnerabilities were identified.

Findings Summary

The testing team identified a total of 2 issues during this assessment. Categorised by their severity:

- Low: 1 issue.
- Informational: 1 issue.

Detailed Findings

This section provides a detailed description of the vulnerabilities identified within the Rocketpool smart contract changes in scope. Each vulnerability has a severity classification which is determined from the likelihood and impact of each issue by the matrix given in the Appendix: Vulnerability Severity Classification.

A number of additional properties of the contracts, including gas optimisations, are also described in this section and are labelled as “informational”.

Each vulnerability is also assigned a **status**:

- ***Open:*** the issue has not been addressed by the project team.
- ***Resolved:*** the issue was acknowledged by the project team and updates to the affected contract(s) have been made to mitigate the related risk.
- ***Closed:*** the issue was acknowledged by the project team but no further actions have been taken.

Summary of Findings

ID	Description	Severity	Status
RPHF-01	Inconsistent Value On Estimated Block Per Day	Low	Resolved
RPHF-02	Miscellaneous General Comments	Informational	Resolved

RPHF-01	Inconsistent Value On Estimated Block Per Day		
Asset	`RocketDAOProtocolSettingsAuction.sol`		
Status	**Resolved:** See Resolution		
Rating	Severity: Low	Impact: Low	Likelihood: Low

Description

The code on lines [**39-40**] indicates that an approximation of `7200 blocks` is used to replace the previous `1 day` for measuring time.

```
// >= 1 day (RPIP-33) (approximated by blocks)
require(_value >= 7200, "Value must be >= 7200");
```

There is an inconsistency on line [**19**] where `40320` is meant to represent 7 days.

```
setSettingUint("auction.lot.duration", 40320);          // 7 days
```

If `7200` blocks represent one day, then seven days should be `50400` blocks. The value `40320` represents roughly 5.6 days.

Recommendations

Consider replacing the value of `40320` on line [**19**] with `50400` to ensure consistency.

Resolution

The issue has been resolved in fbdeaa1.

RPHF-02	Miscellaneous General Comments
Asset	All contracts
Status	**Resolved:** See Resolution
Rating	Informational

Description

This section details miscellaneous findings discovered by the testing team that do not have direct security implications:

1. **In-line Comment For Consistency**
   Related Asset(s): Contract.sol
   The changed code on lines [**56-60**] reduced the guardrail from `RPIP-33` to be consistent with `RPIP-4`. Consider adding `RPIP-4` in the in-line comments of line [**56**] and line [**59**] for consistency with other code in function `setSettingUint()`.
2. **Inaccurate Natspec Comment**
   Related Asset(s): RocketDAOProtocolVerifier.sol
   Function `claimBondChallenger()` burns the `totalReward` by `bondBurnPercent`. This behaviour is not reflected in the Natspec comment on line [**273**].

   ```
   /// @notice Called by a challenger to claim bonds (both refunded bonds and any rewards paid)
   ```

   Consider adding information about the reward burn on the Natspec comment.
   A similar issue is also found on the Natspec comment of function `claimBondProposer()`.

Recommendations

Ensure that the comments are understood and acknowledged, and consider implementing the suggestions above.

Resolution

The issues have been resolved in fbdeaa1.

Appendix A Vulnerability Severity Classification

This security review classifies vulnerabilities based on their potential impact and likelihood of occurance. The total severity of a vulnerability is derived from these two metrics based on the following matrix.

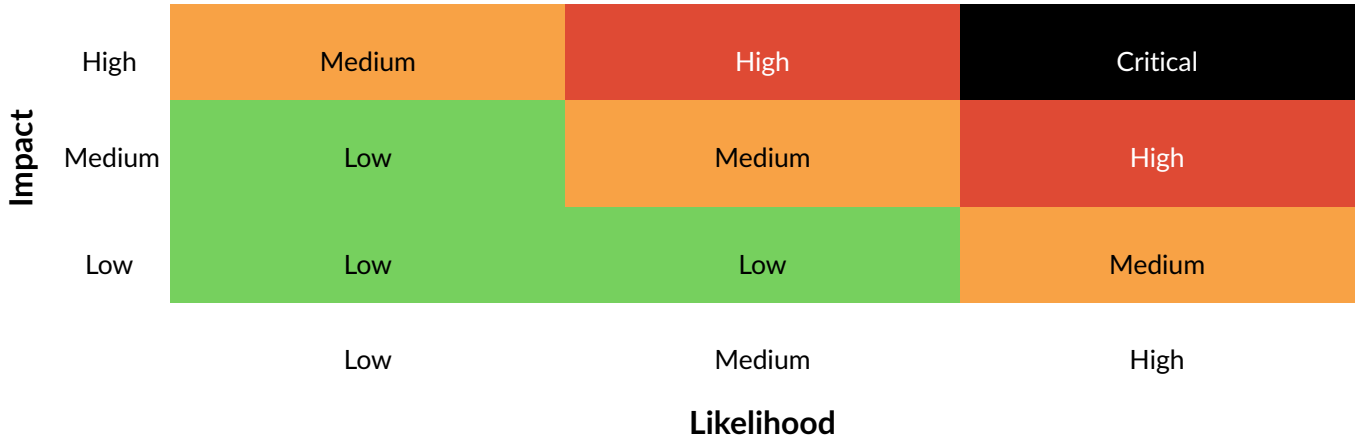

Table 1: Severity Matrix - How the severity of a vulnerability is given based on the *impact* and the *likelihood* of a vulnerability.

References

[1] Sigma Prime. Solidity Security. Blog, 2018, Available: https://blog.sigmaprime.io/solidity-security.html. [Accessed 2018].

[2] NCC Group. DASP - Top 10. Website, 2018, Available: http://www.dasp.co/. [Accessed 2018].