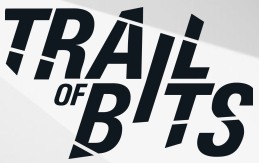

# Roll Staking

## Security Assessment (Summary Report)

**October 4, 2022**

*Prepared for:*
**Andres Aiello**
Roll

*Prepared by:* **Michael Colburn, Anish Naik, and Vara Prasad Bandaru**

# About Trail of Bits

Founded in 2012 and headquartered in New York, Trail of Bits provides technical security assessment and advisory services to some of the world's most targeted organizations. We combine high-end security research with a real-world attacker mentality to reduce risk and fortify code. With 80+ employees around the globe, we've helped secure critical software elements that support billions of end users, including Kubernetes and the Linux kernel.

We maintain an exhaustive list of publications at https://github.com/trailofbits/publications, with links to papers, presentations, public audit reports, and podcast appearances.

In recent years, Trail of Bits consultants have showcased cutting-edge research through presentations at CanSecWest, HCSS, Devcon, Empire Hacking, GrrCon, LangSec, NorthSec, the O'Reilly Security Conference, PyCon, REcon, Security BSides, and SummerCon.

We specialize in software testing and code review projects, supporting client organizations in the technology, defense, and finance industries, as well as government entities. Notable clients include HashiCorp, Google, Microsoft, Western Digital, and Zoom.

Trail of Bits also operates a center of excellence with regard to blockchain security. Notable projects include audits of Algorand, Bitcoin SV, Chainlink, Compound, Ethereum 2.0, MakerDAO, Matic, Uniswap, Web3, and Zcash.

To keep up to date with our latest news and announcements, please follow @trailofbits on Twitter and explore our public repositories at https://github.com/trailofbits. To engage us directly, visit our "Contact" page at https://www.trailofbits.com/contact, or email us at info@trailofbits.com.

**Trail of Bits, Inc.**
228 Park Ave S #80688
New York, NY 10003
https://www.trailofbits.com
info@trailofbits.com

# Notices and Remarks

## Copyright and Distribution



## Test Coverage Disclaimer

All activities undertaken by Trail of Bits in association with this project were performed in accordance with a statement of work and agreed upon project plan.

Security assessment projects are time-boxed and often reliant on information that may be provided by a client, its affiliates, or its partners. As a result, the findings documented in this report should not be considered a comprehensive list of security issues, flaws, or defects in the target system or codebase.

Trail of Bits uses automated testing techniques to rapidly test the controls and security properties of software. These techniques augment our manual security review work, but each has its limitations: for example, a tool may not generate a random edge case that violates a property or may not fully complete its analysis during the allotted time. Their use is also limited by the time and resource constraints of a project.

# Table of Contents

# Executive Summary

## Engagement Overview

Roll engaged Trail of Bits to review the security of its Staking smart contracts. From July 25 to July 29, 2022, a team of three consultants conducted a security review of the client-provided source code, with two person-weeks of effort. Details of the project's timeline, test targets, and coverage are provided in subsequent sections of this report.

## Project Scope

Our testing efforts were focused on the identification of flaws that could result in a compromise of confidentiality, integrity, or availability of the target system. We conducted this audit with full knowledge of the target system, including access to the source code and documentation. We performed static and dynamic testing of the target system and its codebase, using both automated and manual processes.

## Summary of Findings

The audit did not uncover significant flaws that could impact system confidentiality, integrity, or availability. A summary of the findings is provided below.

### EXPOSURE ANALYSIS

| *Severity* | *Count* |
|---|---|
| High | 0 |
| Medium | 1 |
| Low | 3 |
| Informational | 2 |
| Undetermined | 0 |

### CATEGORY BREAKDOWN

| *Category* | *Count* |
|---|---|
| Access Controls | 0 |
| Auditing and Logging | 1 |
| Data Validation | 4 |
| Timing | 1 |
| Undefined Behavior | 0 |

# Project Summary

## Contact Information

The following managers were associated with this project:

**Dan Guido**, Account Manager
dan@trailofbits.com

**Sam Greenup**, Project Manager
sam.greenup@trailofbits.com

The following engineers were associated with this project:

**Michael Colburn**, Consultant
michael.colburn@trailofbits.com

**Anish Naik**, Consultant
anish.naik@trailofbits.com

**Vara Prasad Bandaru**, Consultant
vara.bandaru@trailofbits.com

## Project Timeline

The significant events and milestones of the project are listed below.

| Date | Event |
|---|---|
| **June 9, 2022** | Pre-project onboarding architecture call |
| **July 21, 2022** | Pre-project kickoff call |
| **August 2, 2022** | Delivery of report draft |
| **August 2, 2022** | Report readout meeting |
| **August 19, 2022** | Delivery of final report |
| **September 27, 2022** | Review of fixes implemented by Roll |
| **October 4, 2022** | Delivery of final report with fix log appendix |

# Project Targets

The engagement involved a review and testing of the following target.

### Social Staking

| | |
|---|---|
| Repository | https://github.com/TuringAdvisoryGroup/social-money-staking |
| Version | `b04879976529f902a5d860737f38f2987eeed81f` |
| Type | Solidity |
| Platform | EVM |

# Project Coverage

This section provides an overview of the analysis coverage of the review, as determined by our high-level engagement goals. Our approaches and their results include the following:

- We reviewed the code related to making state changes to the `periodStart` and `periodFinish` variables to identify any edge cases that could cause undefined system behavior. This review uncovered one issue: a specific state change to `periodStart` could cause the staking contract to become unusable (issue #28).
- We reviewed the data validation performed on token balances and reward rates. This review uncovered one issue that could allow a malicious `RollStakingRewards` contract owner to send fewer rewards to the contract than expected (issue #29).
- We investigated whether a user or owner could steal staked tokens. This investigation did not uncover any findings.
- We investigated whether a user could receive more rewards than expected. This investigation did not uncover any findings.
- We reviewed the sequential state changes performed during the `stake`, `withdraw`, and `getReward` operations to identify whether these changes could allow users to steal contract funds or receive more rewards. This investigation did not uncover any findings.
- We checked for front-running vulnerabilities in the contract initialization processes and in updates to reward durations and amounts. We identified one issue that could allow a malicious attacker to front-run the initialization of the `RollStakingFactory` contract *if* the `@openzeppelin/hardhat-upgrades` library fails to function as expected during production deployment (issue #31).
- We reviewed the use of arbitrary ERC20 tokens and their impact on system behavior. We discovered one issue related to the insufficient use of the `SafeERC20` library (issue #30).
- We reviewed whether the system is vulnerable to reentrancy attacks. This review did not uncover any findings.
- We reviewed whether a user could receive additional rewards after contract expiration. This investigation did not uncover any findings.
- We reviewed the arithmetic used in the protocol and changes made to the `rewards` and `freeTokens` arrays. This investigation did not uncover any findings.

## Coverage Limitations

Because of the time-boxed nature of testing work, it is common to encounter coverage limitations. The following list outlines the coverage limitations of the engagement and indicates system elements that may warrant further review:

- We were able to conduct only minimal fuzzing of the Staking codebase. The protocol would benefit from more extensive dynamic testing.

# Summary of Findings

The table below summarizes the findings of the review, including type and severity details.

| ID | Title | Type | Severity |
|---|---|---|---|
| 1 | Incorrect balance check in notifyRewardAmount | Data Validation | **Medium** |
| 2 | If periodStart is set to block.timestamp, the staking contract becomes unusable | Data Validation | **Low** |
| 3 | Insufficient use of the SafeERC20 library for ERC20 tokens | Data Validation | **Low** |
| 4 | Incorrect and insufficient use of event logging | Auditing and Logging | **Low** |
| 5 | State variable shadowing | Data Validation | **Informational** |
| 6 | Initialization functions can be front-run | Timing | **Informational** |

# A. Vulnerability Categories

The following tables describe the vulnerability categories, severity levels, and difficulty levels used in this document.

| Vulnerability Categories | |
|---|---|
| **Category** | **Description** |
| **Access Controls** | Insufficient authorization or assessment of rights |
| **Auditing and Logging** | Insufficient auditing of actions or logging of problems |
| **Authentication** | Improper identification of users |
| **Configuration** | Misconfigured servers, devices, or software components |
| **Cryptography** | A breach of system confidentiality or integrity |
| **Data Exposure** | Exposure of sensitive information |
| **Data Validation** | Improper reliance on the structure or values of data |
| **Denial of Service** | A system failure with an availability impact |
| **Error Reporting** | Insecure or insufficient reporting of error conditions |
| **Patching** | Use of an outdated software package or library |
| **Session Management** | Improper identification of authenticated users |
| **Testing** | Insufficient test methodology or test coverage |
| **Timing** | Race conditions or other order-of-operations flaws |
| **Undefined Behavior** | Undefined behavior triggered within the system |

| Severity Levels | |
|---|---|
| **Severity** | **Description** |
| **Informational** | The issue does not pose an immediate risk but is relevant to security best practices. |
| **Undetermined** | The extent of the risk was not determined during this engagement. |
| **Low** | The risk is small or is not one the client has indicated is important. |
| **Medium** | User information is at risk; exploitation could pose reputational, legal, or moderate financial risks. |
| **High** | The flaw could affect numerous users and have serious reputational, legal, or financial implications. |

| Difficulty Levels | |
|---|---|
| **Difficulty** | **Description** |
| **Undetermined** | The difficulty of exploitation was not determined during this engagement. |
| **Low** | The flaw is well known; public tools for its exploitation exist or can be scripted. |
| **Medium** | An attacker must write an exploit or will need in-depth knowledge of the system. |
| **High** | An attacker must have privileged access to the system, may need to know complex technical details, or must discover other weaknesses to exploit this issue. |

# B. Fix Log

| ID | Title | Type | Severity | Fix Status |
|---|---|---|---|---|
| 1 | Incorrect balance check in notifyRewardAmount | Data Validation | Medium | Fixed (PR 41) |
| 2 | If periodStart is set to block.timestamp, the staking contract becomes unusable | Data Validation | Low | Fixed (PR 40) |
| 3 | Insufficient use of the SafeERC20 library for ERC20 tokens | Data Validation | Low | Fixed (PR 38) |
| 4 | Incorrect and insufficient use of event logging | Auditing and Logging | Low | Fixed (PR 39) |
| 5 | State variable shadowing | Data Validation | Informational | Fixed (PR 37) |
| 6 | Initialization functions can be front-run | Timing | Informational | Fixed |