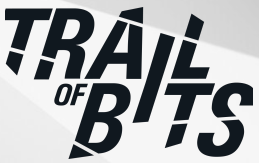

Roll Memberships

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 Membership 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 uncovered significant flaws that could impact system confidentiality, integrity, or availability. A summary of the findings is provided below.

EXPOSURE ANALYSIS

Severity	*Count*
High	4
Medium	1
Low	3
Informational	2
Undetermined	0

CATEGORY BREAKDOWN

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

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.

Memberships

Repository	https://github.com/TuringAdvisoryGroup/memberships
Version	`b604542affe10920a773d7dfb5a9ced1db25037d`
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 access controls and roles used in the protocol to determine whether privileged users could have unexpected access to user funds. We identified several ways in which a malicious actor with the privileges of one of the various protocol roles could access user funds (issue #62).
- We investigated whether an unprivileged user could steal tokens from the contracts. This investigation did not uncover any findings.
- We reviewed the protocol's interactions with arbitrary ERC20 tokens and their impact on system behavior. We discovered one issue related to the insufficient use of the `SafeERC20` library (issue #63). We also identified an issue related to an incorrect argument that is passed to calls to grant an allowance (issue #64).
- We investigated whether the system is vulnerable to reentrancy attacks. This investigation did not uncover any findings.
- We reviewed how parameter changes at various points in a campaign's lifecycle could result in unexpected behavior. This review did not uncover any findings.
- We reviewed the campaign creation process and the update logic to ensure that data is validated adequately and consistently. This review uncovered one issue: if a global fee limit is set outside of the expected range, a denial of service could occur (issue #60).
- We also investigated whether a campaign created by one user could influence campaigns created by other users. We identified one issue that could cause campaign creator funds to become locked in the contract after the campaign concludes (issue #66) and another issue that could cause the campaign state to be overwritten as a result of a hash collision (issue #61).

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 unable to perform dynamic fuzz testing on the memberships codebase. We recommend that the Roll team employ such testing to ensure that system properties are preserved across various system states. .

Summary of Findings

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

ID	Title	Type	Severity
1	Insufficient event generation	Auditing and Logging	Informational
2	Missing validation when updating the minimum Roll fee	Data Validation	Medium
3	Potential collision in schedule IDs of an owner's different campaigns	Undefined Behavior	High
4	Undocumented access privileges of contract deployers	Access Controls	High
5	Memberships is incompatible with nonstandard ERC20 tokens	Undefined Behavior	Low
6	Incorrect token allowance arguments could disrupt existing campaigns	Data Validation	High
7	Referred campaigns cannot be removed from a referral party's campaignsByAddress	Undefined Behavior	Low
8	Users can buy more lots than the predefined amount of a minting schedule	Data Validation	High
9	Memberships contract could lock ether	Data Validation	Low
10	State variable shadowing	Undefined Behavior	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	Insufficient event generation	Auditing and Logging	Informational	Fixed (PR 79)
2	Missing validation when updating the minimum Roll fee	Data Validation	Medium	Fixed (PR 73)
3	Potential collision in schedule IDs of an owner's different campaigns	Undefined Behavior	High	Fixed (PR 77)
4	Undocumented access privileges of contract deployers	Access Controls	High	Fixed (PR 83)
5	Memberships is incompatible with nonstandard ERC20 tokens	Undefined Behavior	Low	Fixed (PR 74)
6	Incorrect token allowance arguments could disrupt existing campaigns	Data Validation	High	Fixed (PR 74)
7	Referred campaigns cannot be removed from a referral party's campaignsByAddress	Undefined Behavior	Low	Fixed (PR 78)
8	Users can buy more lots than the predefined amount of a minting schedule	Data Validation	High	Fixed (PR 57)
9	Memberships contract could lock ether	Data Validation	Low	Fixed (PR 75)
10	State variable shadowing	Undefined Behavior	Informational	Fixed (PR 76)