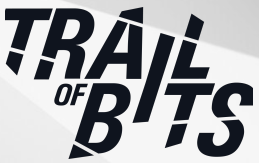

Hydrogen Labs Rover Protocol

Security Assessment (Summary Report)

May 30, 2024

Prepared for:
Meir Bank
Hydrogen Labs

Prepared by: **Michael Colburn**

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 100+ 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.
497 Carroll St., Space 71, Seventh Floor
Brooklyn, NY 11215
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

Project Summary

Contact Information

The following project manager was associated with this project:

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

The following engineering director was associated with this project:

Josselin Feist, Engineering Director, Blockchain
josselin.feist@trailofbits.com

The following consultant was associated with this project:

Michael Colburn, Consultant
michael.colburn@trailofbits.com

Project Timeline

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

Date	Event
May 1, 2024	Pre-project kickoff call
May 3, 2024	Delivery of report draft
May 3, 2024	Report readout meeting
May 30, 2024	Delivery of summary report

Project Targets

The engagement involved a review and testing of the following target. For a complete list of files and their individual SHA-256 hashes, see appendix B.

rover-contracts

Repository	https://github.com/Hydrogen-Labs/rover-contracts
Version	`b31502c225427259c6786f6c12a73abcd5bbe3ef`
Type	Solidity
Platform	EVM

Executive Summary

Engagement Overview

Hydrogen Labs engaged Trail of Bits to review the security of its Rover protocol Solidity smart contracts at commit `b31502c` of the `rover-contracts` repository. The protocol implements a liquid staking token on the Botanix network.

One consultant conducted the review from May 1 to May 2, 2024, for a total of two engineer-days of effort. With full access to source code and documentation, we performed static and dynamic testing of the codebase, using automated and manual processes.

Observations and Impact

The Rover protocol is made up of three core contracts. The `RoleManager` contract centrally manages all of the access controls for the system. We checked that these roles are granted properly and that there are no gaps in where they are enforced in the other components. The `RovBtcToken` contract implements the actual liquid staking token. It uses the OpenZeppelin ERC-20 implementation as a base. We checked that it is initialized correctly and that the pause functionality is integrated properly. The `StakeManager` contract is the primary entrypoint for end users. Currently it allows only deposits, not withdrawals, and does not actually stake deposited bitcoin. We checked that the contract's internal bookkeeping is sound, that the deposit cap is enforced properly, and that its pause functionality works as intended. We also reviewed the upgradeability pattern used by the contracts for any issues that could impact the upgrade process or cause a clash in storage.

We did not identify any security issues during our review. Overall, the system appears to be well designed and contains only the minimum viable functionality necessary at this time. The protocol is expected to evolve with the Botanix network, which is still in a nascent stage in development.

Recommendations

We recommend that the Hydrogen Labs team continue to develop a more flexible and advanced test suite, especially as more functionality is added to the protocol, and begin to document an incident response plan and a roadmap toward eventually further decentralizing control over the protocol. As new functionality is added, consider when it may warrant a follow-up security review.

Codebase Maturity Evaluation

Trail of Bits uses a traffic-light protocol to provide each client with a clear understanding of the areas in which its codebase is mature, immature, or underdeveloped. Deficiencies identified here often stem from root causes within the software development life cycle that should be addressed through standardization measures (e.g., the use of common libraries, functions, or frameworks) or training and awareness programs.

Category	Summary	Result
Arithmetic	The contracts use a version of the Solidity compiler with built-in overflow protection. The arithmetic is quite simple, limited to a bounds check in the `StakeManager` contract to ensure that deposited bitcoin will not exceed the total value locked (TVL) cap.	**Satisfactory**
Auditing	The `StakeManager` contract emits an event when bitcoin is deposited. Events related to roles, minting, and burning are handled by the parent contracts. Consider whether adding events for changes to the paused status or TVL cap would be useful for off-chain monitoring. We were not provided with an incident response plan.	**Moderate**
Authentication / Access Controls	The system has a set of roles for managing different functionality across the token and staking contracts; these roles are managed centrally in their own `RoleManager` contract. The capabilities of these roles are clearly documented in the code and, with the exception of the default admin role, in the repository's README.	**Satisfactory**
Complexity Management	The contracts are built using well-known base contracts with minimal custom logic. They are broken up into components with clear separation of concerns. Take care to maintain this practice as new functionality is added over time.	**Strong**
Decentralization	The current contracts are all upgradeable with no timelock, and the ability to upgrade them will be controlled by a multisignature account. Outside of an upgrade, balances and funds cannot currently be manipulated directly by an admin, though the ability to	**Weak**

	deposit bitcoin into the system or transfer rovBTC in general can be blocked by pausing either of the respective contracts. The project documentation suggests that a transition to a more decentralized governance system is an ultimate goal as the Botanix ecosystem matures.	
Documentation	The repository has adequate documentation and diagrams to describe the protocol, and the contracts and their interfaces have thorough code comment coverage.	**Satisfactory**
Low-Level Manipulation	There are no instances of inline assembly or low-level calls in the contracts.	**Not Applicable**
Testing and Verification	The contracts have basic hard-coded tests in place for all functionality. This is adequate given the limited capabilities of the current system. Making the test suite more flexible and beginning to integrate more advanced techniques like fuzzing will facilitate testing as the scope of the system expands.	**Moderate**
Transaction Ordering	We did not identify any instances in which transaction ordering would seriously impact the protocol or its users.	**Not Applicable**

A. Code Maturity Categories

The following tables describe the code maturity categories and rating criteria used in this document.

Code Maturity Categories	
Category	**Description**
Arithmetic	The proper use of mathematical operations and semantics
Auditing	The use of event auditing and logging to support monitoring
Authentication / Access Controls	The use of robust access controls to handle identification and authorization and to ensure safe interactions with the system
Complexity Management	The presence of clear structures designed to manage system complexity, including the separation of system logic into clearly defined functions
Cryptography and Key Management	The safe use of cryptographic primitives and functions, along with the presence of robust mechanisms for key generation and distribution
Decentralization	The presence of a decentralized governance structure for mitigating insider threats and managing risks posed by contract upgrades
Documentation	The presence of comprehensive and readable codebase documentation
Low-Level Manipulation	The justified use of inline assembly and low-level calls
Testing and Verification	The presence of robust testing procedures (e.g., unit tests, integration tests, and verification methods) and sufficient test coverage
Transaction Ordering	The system's resistance to transaction-ordering attacks

Rating Criteria	
Rating	**Description**
Strong	No issues were found, and the system exceeds industry standards.
Satisfactory	Minor issues were found, but the system is compliant with best practices.
Moderate	Some issues that may affect system safety were found.
Weak	Many issues that affect system safety were found.
Missing	A required component is missing, significantly affecting system safety.
Not Applicable	The category is not applicable to this review.
Not Considered	The category was not considered in this review.
Further Investigation Required	Further investigation is required to reach a meaningful conclusion.

B. Detailed Scope

The table below lists the specific files that were in scope for this assessment, as well as the `sha256sum` of each file's contents. All of the paths are relative to the `packages/foundry/contracts/` directory in the repository listed in Project Targets.

File Name	SHA-256 Hash
`./AccessControl/IRoleManager.sol`	`445930f46e4ae7d7f5b4d47a42654626b5b80a919f6b93a1f015f7f5e537f275`
`./AccessControl/RoleManager.sol`	`7c2949096ffb4fe846f94766b2dbbe7ab06de374844cecd188d51da900ebb2ee`
`./AccessControl/RoleManagerStorage.sol`	`786cd847d9ea05fe9c2f70430f02eda8ddac2b4eef2761900e4ffd195880f696`
`./Errors/Errors.sol`	`2c46654dd39ecad4c3adba4a2b7d056a1be159e0fc85bd049365089db50926f7`
`./RoverBtcToken/IRovBtcToken.sol`	`b7ed51bfaef5d31c05858bd9f3f6790f67174b6bfcbb2fead59e7f44f7f92fb5`
`./RoverBtcToken/RovBtcToken.sol`	`376ea620c722b82abef7cee814949fc2b2bf97ef242c7acd1e2c8e8327363af9`
`./RoverBtcToken/RovBtcTokenStorage.sol`	`596247b790647707d49ded84ac741d246c8fe5df24bdd4dc19f3ddf148820b3c`
`./StakeManager/IStakeManager.sol`	`2c0405a0ea968c06c7e4e7a4e30cbf6ab36d605660979d26da23478b9a3e8280`
`./StakeManager/StakeManager.sol`	`ea22f92cfdec7e3efafd8f8a757966e42e2f25e45a79b03b0d744a8b354d20f8`
`./StakeManager/StakeManagerStorage.sol`	`6b14b63da6e46ffa21c68c654cf05afd2d5046916a8d90d299b49878cf40615f`