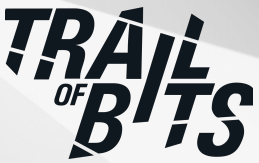

Gondi Retainer

Security Assessment (Summary Report)

July 28, 2023

Prepared for:
0xEND
Gondi

Prepared by: **Bo Henderson**

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.
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

Gondi engaged Trail of Bits to review the security of recent changes made to the `florida-contracts` repository.

A team of one consultant conducted the review on July 24, 2023, for a total of six engineer-hours of effort. Our testing efforts focused on the safety and security of multisource loans and partial refinancing features added since our previous review of the same codebase. With full access to source code and documentation, we performed static and dynamic testing of the target codebase, using automated and manual processes.

Observations and Impact

The changes reviewed during this engagement represent a thorough overhaul of the previously reviewed code. Deprecation of the `Vault` contract layer has simplified the flow of funds and decreased the attack surface. The `MultiSourceLoan` contract that replaces the previous `SingleSourceLoan` contract generalizes lending activity, which allows smaller actors to participate while still maintaining safety guarantees similar to the previous iteration. Due to the large scope of changes and the small amount of time available, this review was conducted on a best-effort basis, and some aspects of the newly introduced logic were investigated only lightly. However, the newly introduced documentation explains the new features clearly and allowed us to ramp up more quickly than we otherwise could have.

The following tables provide the number of findings by severity and category.

EXPOSURE ANALYSIS

Severity	*Count*
High	0
Medium	0
Low	0
Informational	1
Undetermined	0

CATEGORY BREAKDOWN

Category	*Count*
Timing	1

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 engineer was associated with this project:

Bo Henderson, Consultant
bo.henderson@trailofbits.com

Project Timeline

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

Date	Event
July 24, 2023	Retainer review conducted
July 25, 2023	Delivery of draft report
July 28, 2023	Delivery of summary report

Project Targets

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

`florida-contracts`

Repository	https://github.com/pixeldaogg/florida-contracts
Prior Version	`9cc2fe559994fd29736d504e2f4a10ed423884d1`
New Version	`d1b0f5a292c7b4454d845a708d5fd2e92b1583a6`
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 included the following:

- **`Vault`**. The ERC-4626 `Vault` component has been deprecated and removed from the Gondi system. Instead, the `MultiSourceLoan` contract transfers funds directly to and from user addresses. We verified that required `Vault` functionality was properly ported to other contracts, such as by enforcing whitelists for valid loan contracts in the `AuctionLoanLiquidator` contract and by managing protocol fees in the `BaseLoan` contract.

- **`MultiSourceLoan`**. This contract represents a generalization of the deprecated `SingleSourceLoan` contract and allows multiple lenders to supply assets to the borrower. If a single lender is initially present, new lenders can supply part of the loan, with their funds being sent to pay back part of the original lender's loan. We reviewed the full and partial refinancing functionality, verified that funds can flow safely through the system, and analyzed the risk of timing attacks.

- **`AuctionLoanLiquidator`**. If a loan features only a single lender, then the collateral NFT is transferred to the lender upon borrower default. If a loan features more than one lender, then the NFT is transferred to the `AuctionLoanLiquidator` contract and auctioned off, with the proceeds split proportionally among lenders. We reviewed the newly introduced "quiet-ending" mechanism, loan contract validity checks, and the application of protocol fees to ensure these behaviors are in line with the behavior described by the documentation.

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:

- While reviewing the logic in the new `MultiSourceLoan` contract, we focused primarily on the newly added features described above. Our coverage of a narrowly defined set of added features should not be misinterpreted as a thorough review of all logic contained in these contracts. However, an earlier version of these contracts was audited by us. Refer to our previous security assessment for a more in-depth report regarding the logic not covered here.

- Due to the very short timeframe available, this was a best-effort review and reported issues may feature misunderstandings, or other issues may be present that were not reported.

Summary of Findings

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

ID	Title	Type	Severity
1	Loan liquidation can be front-run with a refinance	Timing	**Informational**

Detailed Findings

1. Loan liquidation can be front-run with refinancing	
Severity: **Informational**	Difficulty: **High**
Type: Timing	Finding ID: TOB-GONDI2-1
Target: `src/lib/loans/MultiSourceLoan.sol`	

Description

Last-second refinancing allows a quick user to take advantage of mispriced loans, without giving other lenders an opportunity to respond.

NFT loans on the Gondi platform achieve economic efficiency through an informal auction mechanism in which lenders compete against each other to provide bigger, cheaper, or longer-duration loans in exchange for the opportunity to seize NFT collateral if the borrower fails to repay. In the case of borrower default where multiple lenders are present, this informal lender auction transitions to a formal bidder auction dictated by the `AuctionLoanLiquidator` contract. During the formal auction process, last-second bids result in a "quiet ending" by extending the auction until a predefined period has passed without any activity.

This quiet-ending mechanism is not present during the informal lender-auction phase of the loan. Full refinance offers will be instantly accepted if the APR and/or principal are significantly better than the current loan.

```
if (_renegotiationOffer.duration != 0) {
    revert PartialOfferCannotChangeDurationError();
}
```

Figure 1.1: Partial refinance operations are prevented from changing the loan duration. (florida-contracts/src/lib/loans/MultiSourceLoan.sol#L551–L553)

Additionally, as shown in figure 1.1, partial refinancing ensures that the loan duration does not change. As a result, there are no limits to prevent refinancing from occurring at the last second.

Exploit Scenario

Eve configures a sniper bot that scans all outstanding loans to find those where the loan value is less than the predicted value of selling the NFT. Any time one of these loans is about to default, her bot submits a last-second full refinancing offering 0% APR with the

same duration and principal. Because the loan is about to default, she experiences near-zero opportunity cost for providing a loan at 0% and she becomes the only lender, without any time available for other lenders to respond. As a result, honest lenders are disappointed when they do not receive the NFT collateral that they expected to receive, and they gradually stop lending.

Recommendations

Short term, consider introducing a quiet-ending mechanism to the refinancing process, similar to that present in the `AuctionLoanLiquidator` contract. This would require a predefined period to have passed with no lender activity before the liquidation process can be started. This would allow honest lenders to respond to last-minute refinancing and make such sniper bots less enticing to produce and maintain.

Before implementing this recommendation, carefully consider ways in which this new mechanism could be misused. Due to the time-boxed nature of this retainer, the implications of this change have not been fully mapped out.

Long term, identify and document all time-sensitive actions and the impacts of taking such actions at the last second. This will facilitate a review of similar timing-attack vectors and help inform users of the risks, leading to fewer unpleasant surprises.

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. Code Quality Recommendations

The following recommendations are not associated with specific vulnerabilities. However, they enhance code readability and may prevent the introduction of vulnerabilities in the future.

- **Remove references to the deprecated `Vault` contract.** A few references to the `Vault` are present in code comments and tests. A recursive `grep` will help Gondi identify these.
- **Remove the `requiresLiquidation` parameter from the `LoanOffer` function**. With the transition from the `SingleSourceLoan` contract to the `MultiSourceLoan` contract, the `requiresLiquidation` parameter no longer has any effect.