MixBytes()

Gearbox InfiniFi Integration Security Audit Report

NOVEMBER 14, 2025

Table of Contents

1. Introduction

1.1 Disclaimer

The audit makes no statements or warranties regarding the utility, safety, or security of the code, the suitability of the business model, investment advice, endorsement of the platform or its products, the regulatory regime for the business model, or any other claims about the fitness of the contracts for a particular purpose or their bug-free status.

1.2 Executive Summary

Gearbox Protocol's InfiniFi integration enables credit account users to deposit assets into the InfiniFi protocol and receive iUSD (receipt tokens). Users can further stake iUSD to receive siUSD (staked tokens) or create locked positions with various unwinding epochs to earn additional rewards. Users can redeem iUSD for assets through the redemption process, which may be processed immediately or enqueued depending on available liquidity in InfiniFi. The `InfinifiUnwindingGateway` contract manages unwinding positions and tracks user balances during the unwinding period. Pending unwinding positions can be valued as collateral by Gearbox during the unwinding period and after withdrawal via the `InfinifiUnwindingPhantomToken` contract.

The audit was conducted over 4 days by 3 auditors, involving an in-depth manual code review and automated analysis within the scope.

During the audit, we reviewed the following potential attack vectors and security aspects:

- We verified that the claimable timestamp calculation is consistent with InfiniFi's implementation.
- We validated that the ID generation logic matches InfiniFi's implementation and produces identical results.
- We confirmed that `InfinifiUnwindingGateway.withdraw()` indexes the `userToUnwindingData` mapping by `msg.sender` and transfers iUSD to `msg.sender` only.
- We verified that the adapters properly adhere to the Gearbox adapter architecture constraints, including the enforcement of the `onlyCreditFacade` and `onlyConfigurator` restrictions.
- We verified that credit accounts do not lose their tokens when InfiniFi has insufficient liquidity for immediate redemption, and that these tokens can be claimed later via `claimRedemption()` once liquidity becomes available in InfiniFi.

The codebase is well-written, well-documented, and of high quality. Its adapter and gateway contract architecture is cleanly designed, demonstrating a clear separation of concerns between components.

1.3 Project Overview

Summary

Title	Description
Client Name	Gearbox
Project Name	InfiniFi Integration
Type	Solidity
Platform	EVM
Timeline	04.11.2025 - 11.11.2025

Scope of Audit

File	Link
contracts/adapters/infinifi/InfinifiUnwindingGatewayAdapter.sol	InfinifiUnwindingGatewayAdapter.sol
contracts/adapters/infinifi/InfinifiGatewayAdapter.sol	InfinifiGatewayAdapter.sol
contracts/helpers/infinifi/InfinifiUnwindingPhantomToken.sol	InfinifiUnwindingPhantomToken.sol
contracts/helpers/infinifi/InfinifiUnwindingGateway.sol	InfinifiUnwindingGateway.sol

Versions Log

Date	Commit Hash	Note
04.11.2025	54221b884babfe43d4c09de9d5411833ced627be	Initial Commit
11.11.2025	b3cc54453225b0f163aaa48f812dbd5ff5c9a148	Re-audit Commit

Mainnet Deployments

Deployment verification will be conducted via https://permissionless.gearbox.foundation/bytecode/.

1.4 Security Assessment Methodology

Project Flow

Stage	Scope of Work
Interim audit	**Project Architecture Review:** · Review project documentation · Conduct a general code review · Perform reverse engineering to analyze the project's architecture based solely on the source code · Develop an independent perspective on the project's architecture · Identify any logical flaws in the design OBJECTIVE: UNDERSTAND THE OVERALL STRUCTURE OF THE PROJECT AND IDENTIFY POTENTIAL SECURITY RISKS.
	Code Review with a Hacker Mindset: · Each team member independently conducts a manual code review, focusing on identifying unique vulnerabilities. · Perform collaborative audits (pair auditing) of the most complex code sections, supervised by the Team Lead. · Develop Proof-of-Concepts (PoCs) and conduct fuzzing tests using tools like Foundry, Hardhat, and BOA to uncover intricate logical flaws. · Review test cases and in-code comments to identify potential weaknesses. OBJECTIVE: IDENTIFY AND ELIMINATE THE MAJORITY OF VULNERABILITIES, INCLUDING THOSE UNIQUE TO THE INDUSTRY.
	Code Review with a Nerd Mindset: · Conduct a manual code review using an internally maintained checklist, regularly updated with insights from past hacks, research, and client audits. · Utilize static analysis tools (e.g., Slither, Mythril) and vulnerability databases (e.g., Solodit) to uncover potential undetected attack vectors. OBJECTIVE: ENSURE COMPREHENSIVE COVERAGE OF ALL KNOWN ATTACK VECTORS DURING THE REVIEW PROCESS.

Stage	Scope of Work
	Consolidation of Auditors' Reports: · Cross-check findings among auditors · Discuss identified issues · Issue an interim audit report for client review OBJECTIVE: COMBINE INTERIM REPORTS FROM ALL AUDITORS INTO A SINGLE COMPREHENSIVE DOCUMENT.
Re-audit	**Bug Fixing & Re-Audit:** · The client addresses the identified issues and provides feedback · Auditors verify the fixes and update their statuses with supporting evidence · A re-audit report is generated and shared with the client OBJECTIVE: VALIDATE THE FIXES AND REASSESS THE CODE TO ENSURE ALL VULNERABILITIES ARE RESOLVED AND NO NEW VULNERABILITIES ARE ADDED.
Final audit	**Final Code Verification & Public Audit Report:** · Verify the final code version against recommendations and their statuses · Check deployed contracts for correct initialization parameters · Confirm that the deployed code matches the audited version · Issue a public audit report, published on our official GitHub repository · Announce the successful audit on our official X account OBJECTIVE: PERFORM A FINAL REVIEW AND ISSUE A PUBLIC REPORT DOCUMENTING THE AUDIT.

1.5 Risk Classification

Severity Level Matrix

Severity	Impact: High	Impact: Medium	Impact: Low
Likehood: High	Critical	High	Medium
Likehood: Medium	High	Medium	Low
Likehood: Low	Medium	Low	Low

Impact

- **High** – Theft from 0.5% OR partial/full blocking of funds (>0.5%) on the contract without the possibility of withdrawal OR loss of user funds (>1%) who interacted with the protocol.
- **Medium** – Contract lock that can only be fixed through a contract upgrade OR one-time theft of rewards or an amount up to 0.5% of the protocol's TVL OR funds lock with the possibility of withdrawal by an admin.
- **Low** – One-time contract lock that can be fixed by the administrator without a contract upgrade.

Likelihood

- **High** – The event has a 50–60% probability of occurring within a year and can be triggered by any actor (e.g., due to a likely market condition that the actor cannot influence).
- **Medium** – An unlikely event (10–20% probability of occurring) that can be triggered by a trusted actor.
- **Low** – A highly unlikely event that can only be triggered by the owner.

Action Required

- **Critical** – Must be fixed as soon as possible.
- **High** – Strongly advised to be fixed to minimize potential risks.
- **Medium** – Recommended to be fixed to enhance security and stability.
- **Low** – Recommended to be fixed to improve overall robustness and effectiveness.

Finding Status

- **Fixed** – The recommended fixes have been implemented in the project code and no longer impact its security.
- **Partially Fixed** – The recommended fixes have been partially implemented, reducing the impact of the finding, but it has not been fully resolved.
- **Acknowledged** – The recommended fixes have not yet been implemented, and the finding remains unresolved or does not require code changes.

1.6 Summary of Findings

Findings Count

Severity	Count
Critical	0
High	0
Medium	2
Low	0

Findings Statuses

ID	Finding	Severity	Status
M-1	Griefing Attack in `startUnwinding()` Allows Blocking of Legitimate Users	Medium	Fixed
M-2	Queued Redemption Funds Excluded from Collateral Valuation	Medium	Acknowledged

2. Findings Report

2.1 Critical

Not Found

2.2 High

Not Found

2.3 Medium

M-1	Griefing Attack in `startUnwinding()` Allows Blocking of Legitimate Users		
Severity	Medium	**Status**	Fixed in b3cc5445

Description

The `InfinifiUnwindingGateway.startUnwinding()` function can be called by any address and is limited to one call per block through a `block.timestamp` check. An attacker can front-run legitimate transactions by invoking the function with minimal amounts at the beginning of each block, effectively preventing other users from starting unwinding operations during that block. This issue is classified as **Medium** severity because it enables a low-cost griefing attack that can temporarily block legitimate users from calling the function, degrading user experience without causing direct fund loss.

Recommendation

We recommend implementing access control to restrict function calls so it cannot be called by anyone, and adding a minimum amount threshold (e.g., `require(shares >= MIN_SHARES)`) to reduce the risk of potential griefing attacks.

Client's Commentary:

***Client**: We've added a minimal share limit of 1e18 (equal to ~1 USD) in b3cc54453225b0f163aaa48f812dbd5ff5c9a148 and we believe this will sufficiently deter potential griefers. We believe that adding access control is not justified by the magnitude of the problem, as this will introduce significant additional complexity and trust assumptions into the contract.*

***MixBytes**: The issue was partially fixed. The introduced minimum share limit significantly reduces the probability of griefing attacks, as an attacker would now need to lock approximately 50k worth of funds for at least one unwinding period (1 week). While the attack vector still theoretically exists, the required capital commitment makes such an attack highly impractical.*

M-2	Queued Redemption Funds Excluded from Collateral Valuation		
Severity	Medium	**Status**	Acknowledged

Description

When `InfinifiGatewayAdapter.redeem()` is called under certain conditions, part (or all) of the requested amount can be enqueued into the protocol's redemption queue and only be claimable later. In the current integration, such enqueued redemptions are explicitly *not considered collateral*, even though these funds are economically secured and held by the protocol during the queueing period. Practically, this forces the position owner to supply additional collateral or face liquidation risk.

Recommendation

Even though this behavior is documented, we recommend counting the pending (queued) redemption liquidity as collateral until it is claimed.

Client's Commentary:

Introducing delayed redemption capability into the contracts would greatly increase their scope and may not be possible due to certain properties of the redemption queue (primarily, due to inability to directly retrieve pending redemptions for a specific account). As this functionality is not strictly required in the business sense, we believe that adding this is not justified at the moment.

On user side, there are means to mitigate unexpected delayed redemptions, such as the slippage check in Credit Facade. This is always enforced on Gearbox UI, so realistically only power users are subject to the risk of having some funds temporarily locked into a non-collateral position (and even then, this will only materialize in the specific case where the amount locked into a delayed redemption is not large enough to trigger a collateral check failure).

2.4 Low

Not Found

3. About MixBytes

MixBytes is a leading provider of smart contract audit and research services, helping blockchain projects enhance security and reliability. Since its inception, MixBytes has been committed to safeguarding the Web3 ecosystem by delivering rigorous security assessments and cutting-edge research tailored to DeFi projects.

Our team comprises highly skilled engineers, security experts, and blockchain researchers with deep expertise in formal verification, smart contract auditing, and protocol research. With proven experience in Web3, MixBytes combines in-depth technical knowledge with a proactive security-first approach.

Why MixBytes

- **Proven Track Record:** Trusted by top-tier blockchain projects like Lido, Aave, Curve, and others, MixBytes has successfully audited and secured billions in digital assets.
- **Technical Expertise:** Our auditors and researchers hold advanced degrees in cryptography, cybersecurity, and distributed systems.
- **Innovative Research:** Our team actively contributes to blockchain security research, sharing knowledge with the community.

Our Services

- **Smart Contract Audits:** A meticulous security assessment of DeFi protocols to prevent vulnerabilities before deployment.
- **Blockchain Research:** In-depth technical research and security modeling for Web3 projects.
- **Custom Security Solutions:** Tailored security frameworks for complex decentralized applications and blockchain ecosystems.

MixBytes is dedicated to securing the future of blockchain technology by delivering unparalleled security expertise and research-driven solutions. Whether you are launching a DeFi protocol or developing an innovative dApp, we are your trusted security partner.

Contact Information

 https://mixbytes.io/

 https://github.com/mixbytes/audits_public

 hello@mixbytes.io

 https://x.com/mixbytes