

MixBytes()

# Gearbox Mellow Adapters Security Audit Report

JULY 15, 2025

# Table of Contents

<b>1. Introduction</b>	<b>2</b>
1.1 Disclaimer	2
1.2 Executive Summary	2
1.3 Project Overview	3
1.4 Security Assessment Methodology	4
1.5 Risk Classification	6
1.6 Summary of Findings	7
<b>2. Findings Report</b>	<b>8</b>
2.1 Critical	8
2.2 High	8
2.3 Medium	8
2.4 Low	8
<b>3. About MixBytes</b>	<b>9</b>

# 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 is an on-chain credit protocol that allows users to borrow funds and deploy leverage across major DeFi protocols using portable "Credit Accounts".

This audit focused on the new Mellow Finance integration, reviewing its adapters to ensure they comply with Gearbox safety rules, prevent asset lock-ups, and preserve the risk assumptions of both protocols.

The codebase was audited in one day by three auditors through a combination of manual review and automated tooling.

During the audit, we verified that the Mellow adapters were implemented according to the Gearbox adapter architecture and adhered to the following guidelines:

1. Every external method that touches a Credit Account is protected with the `creditFacadeOnly` modifier.
2. The contract does not store any end-user state and only keeps immutable parameters and whitelisted settings maintained by the configurator.
3. It implements the Gearbox `IVersion` requirements, exposing immutable `version` and `contractType` constants.
4. Each public function correctly returns the `useSafePrices` flag depending on whether the call leaves the Credit Account with newly-created assets that lack a safe oracle price.
5. Token allowances follow the `safe-approve` pattern: grant `type(uint256).max` before the external call and reset to `1` immediately afterwards.
6. The adapter provides a `serialize()` function that outputs all parameters needed to reconstruct its state off-chain.
7. All configuration actions are grouped in dedicated functions and gated by the `configuratorOnly` modifier.

We also ensured that the implementation does not result in the permanent locking of user assets.

No vulnerabilities were found during the audit. The code is well-structured and demonstrates a high standard of security.

## 1.3 Project Overview

### Summary

Title	Description
Client Name	Gearbox Protocol
Project Name	Mellow Adapters
Type	Solidity
Platform	EVM
Timeline	09.07.2025 – 10.07.2025

### Scope of Audit

File	Link
contracts/adapters/mellow/ MellowWrapperAdapter.sol	<a href="#">MellowWrapperAdapter.sol</a>
contracts/adapters/mellow/ MellowDVVAdapter.sol	<a href="#">MellowDVVAdapter.sol</a>

### Versions Log

Date	Commit Hash	Note
09.07.2025	a4b208e50695536fb9346f981b1588dda5d866e6	Initial 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	<b>Project Architecture Review:</b> <ul style="list-style-type: none"><li>• Review project documentation</li><li>• Conduct a general code review</li><li>• Perform reverse engineering to analyze the project's architecture based solely on the source code</li><li>• Develop an independent perspective on the project's architecture</li><li>• Identify any logical flaws in the design</li></ul> <p><b>OBJECTIVE: UNDERSTAND THE OVERALL STRUCTURE OF THE PROJECT AND IDENTIFY POTENTIAL SECURITY RISKS.</b></p>
	<b>Code Review with a Hacker Mindset:</b> <ul style="list-style-type: none"><li>• Each team member independently conducts a manual code review, focusing on identifying unique vulnerabilities.</li><li>• Perform collaborative audits (pair auditing) of the most complex code sections, supervised by the Team Lead.</li><li>• Develop Proof-of-Concepts (PoCs) and conduct fuzzing tests using tools like Foundry, Hardhat, and BOA to uncover intricate logical flaws.</li><li>• Review test cases and in-code comments to identify potential weaknesses.</li></ul> <p><b>OBJECTIVE: IDENTIFY AND ELIMINATE THE MAJORITY OF VULNERABILITIES, INCLUDING THOSE UNIQUE TO THE INDUSTRY.</b></p>
	<b>Code Review with a Nerd Mindset:</b> <ul style="list-style-type: none"><li>• Conduct a manual code review using an internally maintained checklist, regularly updated with insights from past hacks, research, and client audits.</li><li>• Utilize static analysis tools (e.g., Slither, Mythril) and vulnerability databases (e.g., Solodit) to uncover potential undetected attack vectors.</li></ul> <p><b>OBJECTIVE: ENSURE COMPREHENSIVE COVERAGE OF ALL KNOWN ATTACK VECTORS DURING THE REVIEW PROCESS.</b></p>

Stage	Scope of Work
	<p><b>Consolidation of Auditors' Reports:</b></p> <ul style="list-style-type: none"> <li>• Cross-check findings among auditors</li> <li>• Discuss identified issues</li> <li>• Issue an interim audit report for client review</li> </ul> <p><b>OBJECTIVE: COMBINE INTERIM REPORTS FROM ALL AUDITORS INTO A SINGLE COMPREHENSIVE DOCUMENT.</b></p>
Re-audit	<p><b>Bug Fixing &amp; Re-Audit:</b></p> <ul style="list-style-type: none"> <li>• The client addresses the identified issues and provides feedback</li> <li>• Auditors verify the fixes and update their statuses with supporting evidence</li> <li>• A re-audit report is generated and shared with the client</li> </ul> <p><b>OBJECTIVE: VALIDATE THE FIXES AND REASSESS THE CODE TO ENSURE ALL VULNERABILITIES ARE RESOLVED AND NO NEW VULNERABILITIES ARE ADDED.</b></p>
Final audit	<p><b>Final Code Verification &amp; Public Audit Report:</b></p> <ul style="list-style-type: none"> <li>• Verify the final code version against recommendations and their statuses</li> <li>• Check deployed contracts for correct initialization parameters</li> <li>• Confirm that the deployed code matches the audited version</li> <li>• Issue a public audit report, published on our official GitHub repository</li> <li>• Announce the successful audit on our official X account</li> </ul> <p><b>OBJECTIVE: PERFORM A FINAL REVIEW AND ISSUE A PUBLIC REPORT DOCUMENTING THE AUDIT.</b></p>

## 1.5 Risk Classification

### Severity Level Matrix

Severity	Impact: High	Impact: Medium	Impact: Low
Likelihood: High	Critical	High	Medium
Likelihood: Medium	High	Medium	Low
Likelihood: 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	0
Low	0



## 2. Findings Report

### 2.1 Critical

Not Found

### 2.2 High

Not Found

### 2.3 Medium

Not Found

### 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](https://github.com/mixbytes/audits_public)



[hello@mixbytes.io](mailto:hello@mixbytes.io)



<https://x.com/mixbytes>