

MixBytes()

# Euler AccessControlKeyring Security Audit Report

JUNE 02, 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	5
1.5 Risk Classification	7
1.6 Summary of Findings	8
<b>2. Findings Report</b>	<b>9</b>
2.1 Critical	9
2.2 High	9
2.3 Medium	9
2.4 Low	9
<b>3. About MixBytes</b>	<b>10</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

The `HookTargetAccessControlKeyring` is a security-focused hook for EVK vaults that enforces both function-level access control and credential-based authentication via Keyring. It ensures that only users with appropriate roles or valid credentials can perform sensitive operations like borrowing, depositing, or withdrawing. If the caller is acting on another account's behalf, the hook verifies credentials for both parties.

This security audit was conducted over a 1-day period.

During the audit of the `HookTargetAccessControlKeyring` contract, we thoroughly reviewed its logic, paying special attention to the credential verification mechanisms, the function-specific access control, and the interaction with the Keyring credential registry.

The contract demonstrates **strong code quality, a clean and minimalistic design, and adherence to access control best practices**. No security vulnerabilities were identified. Only minor observations were made regarding operational risks:

**Keyring dependency:** The hook heavily relies on the availability and correctness of the Keyring registry. If the Keyring system becomes unavailable or misconfigured, it could unintentionally lock vaults or freeze protocol operations.

**Careful role configuration:** The system requires precise setup of access permissions to avoid breaking core lending flows. For example, if a borrower successfully calls `borrow()` but later loses permission to call `repay()`, it could lead to a stuck, unclosable debt.

**Hooked operations validation:** We verified that all expected vault operations (`OP_DEPOSIT`, `OP_MINT`, `OP_WITHDRAW`, `OP_REDEEM`, `OP_SKIM`, `OP_BORROW`, `OP_REPAY`, `OP_REPAY_WITH_SHARES`, `OP_PULL_DEBT`, `OP_LIQUIDATE`, and `OP_FLASHLOAN`) are properly hooked and that all critical address fields (`owner`, `receiver`, `from`) are consistently validated through credential checks.

**Role-based bypass mechanism:** The `WILD_CARD` role correctly bypasses credential verification, allowing privileged addresses to operate independently of Keyring.

The audit specifically focused on:

- Verifying the security of the Keyring-based credential checks,
- Ensuring correct enforcement of per-function access control,
- Assessing fallback behavior for unauthorized calls,

- Evaluating resilience to Keyring inconsistencies,
- Checking the operational safety of lending-related workflows under dynamic access control changes,
- Confirming full support and correct validation of all critical hooked operations.

Overall, the `HookTargetAccessControlKeyring` contract is secure, thoughtfully designed, and safe for deployment, provided that operational risks related to Keyring availability and role management are carefully monitored.

## 1.3 Project Overview

### Summary

Title	Description
Client Name	Euler
Project Name	AccessControlKeyring
Type	Solidity
Platform	EVM
Timeline	24.04.2025 - 20.05.2025

### Scope of Audit

File	Link
<code>src/HookTarget/ HookTargetAccessControlKeyring.sol</code>	<a href="#">HookTargetAccessControlKeyring.sol</a>

### Versions Log

Date	Commit Hash	Note
24.04.2025	5e4ed9e3ed1fd406c9987a9c1257290fef29a4e9	Initial Commit
15.05.2025	e9fa4bd2ce21d0746cdeacb94ec089d8ac6118a0	Commit with Updates
20.05.2025	8000f7ef4e6999a5bf118bdf4702ba554405c429	Commit with Updates

## Mainnet Deployments

File	Address	Blockchain
HookTargetAccessControlKeyring.sol	0xC375f...da0f1c6a	Avalanche

## 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>