


# RETR BRIDGE

Audit Report by Yevhenii Bezuhlyi

 @BezuglyiE

Date:

November 2023

# Table of Contents

- 01 Scope
- 02 Disclaimer
- 03 Risks Classification
- 04 Summary
- 05 Findings

## About Retrobridge

Retrobridge is an advanced cross-chain bridging solution designed to simplify the transfer of assets between blockchain networks.

By utilizing concentrated liquidity on each chain, we ensure rapid and direct asset cross-chain transfers, bypassing the complexities of smart contracts and consequently reducing bridging costs. This efficiency is further emphasized through its direct transfer mechanisms.

## Disclaimer

This document presents the findings of a security audit conducted on the specified backend application. The audit was performed based on the information provided by the client and the state of the application at the time of the audit.

- **Scope Limitation**

The findings are confined to the scope agreed upon with the client and may not cover all potential security risks. The audit was conducted on the application's version as provided, and any changes made to the application post-audit may affect the validity of these findings.

- **No Guarantee of Security**

While this audit aims to identify security vulnerabilities and provide recommendations for mitigation, it does not guarantee that the application is free from security risks. New vulnerabilities may emerge, and existing vulnerabilities may be exploited in ways not identified in this audit.

- **Limitation of Liability**

The auditor is not liable for any direct, indirect, incidental, consequential, or any other damages resulting from using the information provided in this audit. The client is responsible for the final decision on implementing the recommendations provided.

# Severities Definition

## Risks Classification

Impact Likelihood	High	Medium	Low
High	Critical	High	Medium
Medium	High	Medium	Low
Low	Medium	Low	Low

## Impact

### High

leads a significant financial losses or irrecoverable data consistency issues.

### Medium

leads a financial losses to only a subset of users, but still unacceptable.

### Low

leads to low financial losses or easily recoverable data consistency issues.

## Likelihood

### High

easy to prepare and execute or are always executing within the system during the regular flow.

### Medium

requires additional preparations and specific conditions.

### Low

requires very specific conditions and most likely will never be executed during the regular flow of the app.

# Summary

---

The audited code contains:

Critical

1 critical issue

Medium

2 medium issue

Low

1 low severity issue

Informational and architectural issues are omitted.

No	Title	Severity	Status
1	Order content substitution	Critical	✓ Fixed
2	Double-spending during an order fulfillment process	Medium	✓ Fixed
3	Insufficient number of confirmations	Medium	✓ Fixed
4	Non-validated response from the prices provider	Low	✓ Fixed

# Findings

---

**Critical**

## Risk Findings (1)

### 1. Order content substitution

<b>Impact</b>	Allows third parties to substitute order information such as the receiving party.	<b>High</b>
<b>Likelihood</b>	Easy to execute	<b>High</b>
<b>Description</b>	The description has been intentionally withheld to ensure the security and confidentiality of the app's architecture and codebase.	
<b>Recommendation</b>	Sessions management should be added to validate a request origin. Authentication could be done by the <b>eip-4361</b> or its variations.	
<b>Resolution</b>	<b>✓ Fixed</b>	

**Medium**

## Risk Findings (2)

### 1. Double-spending during an order fulfillment process

<b>Impact</b>	Can lead to multiple fulfillment of the same order.	<b>High</b>
<b>Likelihood</b>	The service execution must be aborted at its very specific stage of execution	<b>Low</b>
<b>Description</b>	The description has been intentionally withheld to ensure the security and confidentiality of the app's architecture and codebase.	

**Recommendation** Order execution flow should follow the principles of transactional systems. All the external calls should be made in an async mode with the possibility to verify whether the same request has been sent and/or processed before.

**Resolution** ✓ Fixed

## 2. Insufficient number of confirmations

**Impact** Orders that are not funded might be considered as funded. High

**Likelihood** The chances of a network reordering are low-to-impossible within the supported networks. Low

**Description** The description has been intentionally withheld to ensure the security and confidentiality of the app's architecture and codebase.

**Recommendation** Add the recommended number of confirmations for each network where it's required.

**Resolution** ✓ Fixed

Low

## Risk Findings (1)

### 1. Non-validated response from the provider of the price

**Impact** Outdated prices can be used, leading to the platform's financial losses. Low

**Likelihood** Likely to happen if the price provider experiences issues or shuts down. Medium

**Description** The description has been intentionally withheld to ensure the security and confidentiality of the app's architecture and codebase.

**Recommendation**

Implement fallback prices provider. Constantly evict the cache on timeout. Add correct procession for cases when the cache is empty.

**Resolution**

 Fixed