

Leave the Cookies, Take the Payload

Overview

Room URL: <https://tryhackme.com/room/xss-aoc2025-c5j8b1m4t6>

Difficulty: Easy

Category: Cross-Site-Scripting

Date Completed: 12/11/2025

Objectives

- Understand how XSS works
 - Learn to prevent XSS attacks
-

Table of Contents

[Introduction](#)

[Walk Through](#)

[Lessons Learned](#)

[Resources](#)

Introduction

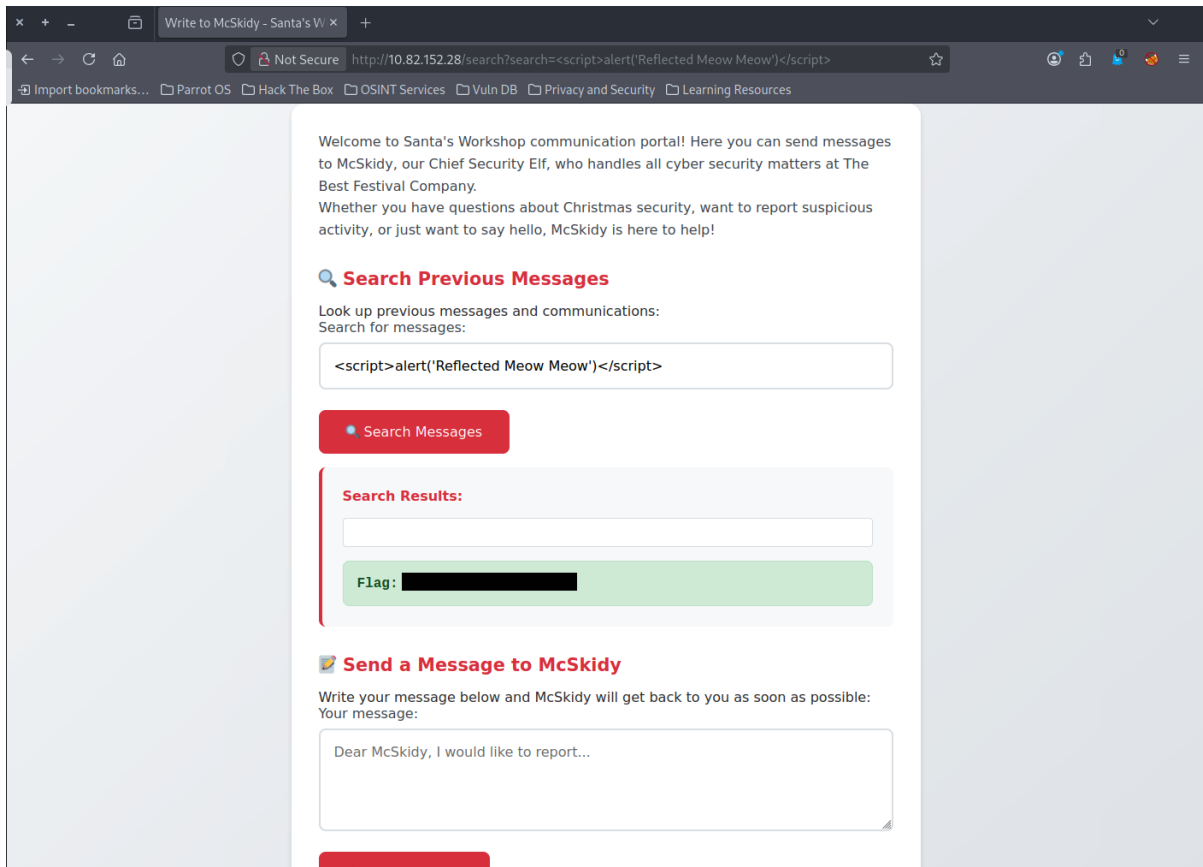
The TBFC messaging portal has been flagged for potential security vulnerabilities, and your mission is to identify and exploit two critical Cross-Site Scripting (XSS) attack vectors. As part of the defensive security team, understanding how attackers abuse unvalidated user input is essential to hardening web applications against injection attacks. In this challenge, you'll navigate a vulnerable web application and demonstrate both **Reflected XSS** and **Stored XSS** attacks—two of the most

dangerous web application vulnerabilities. By successfully injecting malicious payloads through the search functionality and messaging system, you'll uncover how poor input validation can allow attackers to execute arbitrary JavaScript in users' browsers, potentially stealing session cookies, hijacking accounts, or spreading malware. This hands-on exercise reinforces the critical importance of input sanitization and output encoding in secure application design.

Walk Through

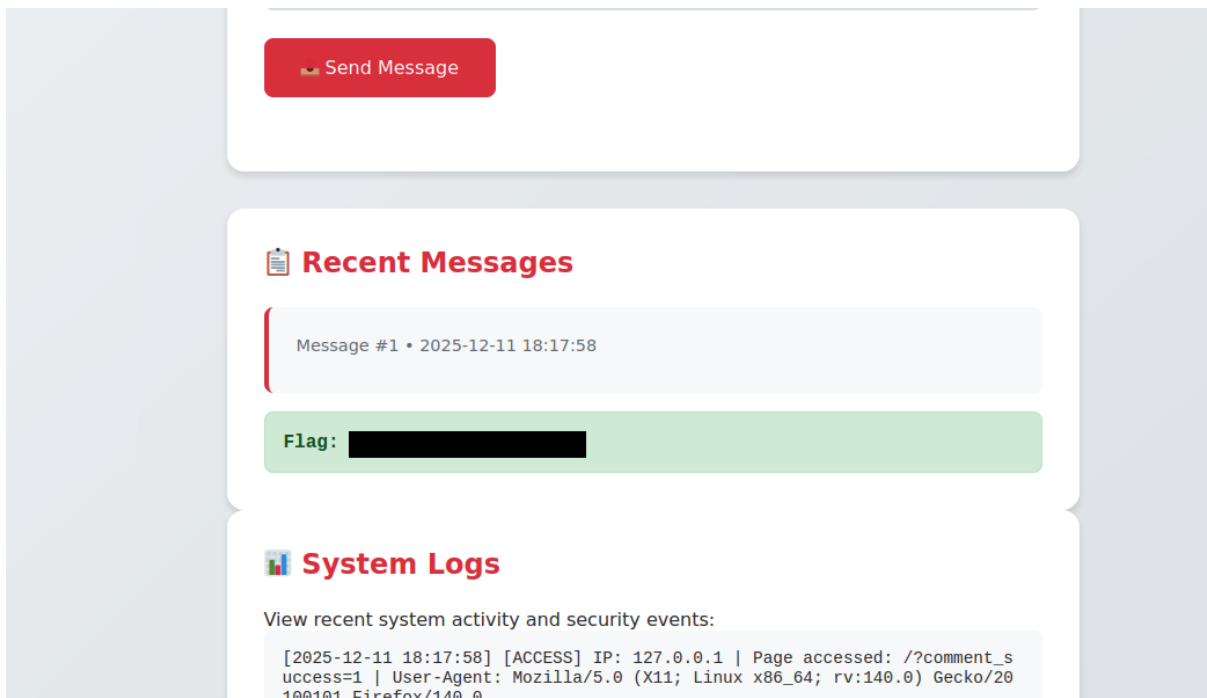
1. Start the target machine
2. Go to the web interface of the target machine
3. Which type of XSS attack requires payloads to be persisted on the backend?
 1. stored
4. What's the reflected XSS flag?
 1. Search messages sends a request. Using the request form can add to test a reflected

XSS.



5. What's the stored XSS flag?
 1. The send a message field stores the input in the database. By using you can see if it is susceptible to a stored attack.

2.



Lessons Learned

- **Input Validation is Non-Negotiable:** Both vulnerabilities stem from the application's failure to validate, encode, or sanitize user input. The difference between Reflected and Stored XSS lies in *persistence*—Reflected XSS targets individual victims through phishing links, while Stored XSS compromises all users who access the affected page.
- **Defense Strategies Are Essential:** Organizations must implement layered defenses including: using `textContent` instead of `innerHTML`, setting `HttpOnly` and `SameSite` cookie attributes to limit JavaScript access, and properly encoding all output to escape potentially dangerous characters before rendering them in the browser. These practices directly prevent the JavaScript injection attacks demonstrated in this challenge.

Resources

[TryHackMe](#)

[OWASP XSS Prevention Cheat Sheet](#)

[PortSwigger Web Security Academy - XSS](#)

Revision #2

Created 2025-12-11 18:28:53 UTC by David Rizzo

Updated 2025-12-11 18:32:33 UTC by David Rizzo