Security+[SY0–601] Lab Walkthrough

Lab 5 – Conducting a Cross Site Scripting (XSS) attack

Hassen Hannachi

November 20, 2025

Contents

List of Figures

1 Introduction

XXS is a common vulnerability in web applications and is frequently listed as a top vulnerability in the OWASP top ten. XXS occurs when web applications execute JavaScript, which is input into the form sections of a web application. The applications perform no security checks on the entered data. It simply passes it straight to the server, causing inputted JavaScript to execute. [1]

2 Environment Setup

Please follow these labs to get hands-on experience for CompTIA Security+ exam [SY0–601]. All the labs use free tools. I STRONGLY suggest you use a virtual machine[2] such as VMware or Virtualbox for these labs to avoid exposing your home PC or laptop. [3]

3 Lab Walkthrough

3.1 Task 1

We will begin this lab by opening a web browser of your choice. There are numerous sites on the web that have been setup for the purpose of practising attacks like XSS. We will be using this site: **https://xss-game.appspot.com** The site has several levels of XSS which vary in difficulty. It also offers you several hints on how to proceed if stuck on a level. This is a great way to advance your knowledge of this type of web application attack.

3.2 Task 2

Let's begin by navigating to the following URL:

https://xss-game.appspot.com/level1

This is the first level. We are presented with a simple search box for a web page.

Figure 1: xss-game.appspot.com

[1] You can use any web browser of your choosing for this lab.

[2] You can use Kali Linux in a virtual machine for the purpose of this lab.

[3] NEVER configure these labs at work using your employers' PCs.

To be able to execute JavaScript in a web application like this one, a basic understanding of the syntax for JavaScript and HTML is required.

For example, `<h1>"Header here"</h1>` will create a header. Enter this value into the search box and see what result you get

Figure 2: create header

Great! We know this application is vulnerable to XSS now, as our input is directly reflected in the output of the search result. Note, what we just did is not XSS as there is no JavaScript involved. We simply know now that the web application is most likely vulnerable to XSS.

3.3 Task 3

Now, to execute the XSS attack. Try to figure it out yourself using the hints the site provides you. The answer is the following:

`<script>alert(1)</script>`

This will cause an alert text box to pop up on our screen with “1” on it.

We have successfully executed an XSS attack.

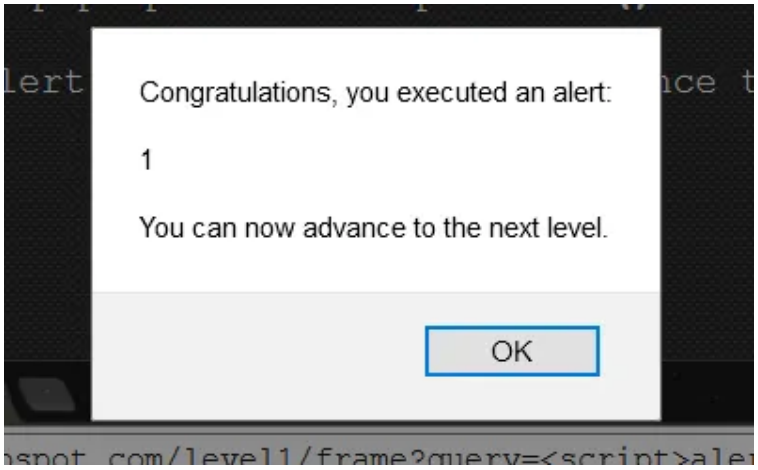

Figure 3: XSS attacl & alert box

3.4 Task 4

For level 2, we will only be talking about it in brief. In this level, we are presented with a forum page.

The script we entered for level 1 will not work here. We need to first enter a HTML tag which will adopt the script we entered in level 1, so that every time this page is visited and the tag is loaded, the XSS attack will run. This is a method of achieving a persistent XXS attack on a site.

```
<img src="http://inexist.ent" onerror="javascript:alert(1)"/>
```

Figure 4: load of an image

This bit of HTML is loading an image, which doesn't exist into the forum. Every time there is an error, the JavaScript alert will run. Considering that the image doesn't exist and that it will be loaded every time a user visits the forum, the JavaScript alert will always run.

Great! We now have all the information we need and are ready to open Hydra and begin the attack.

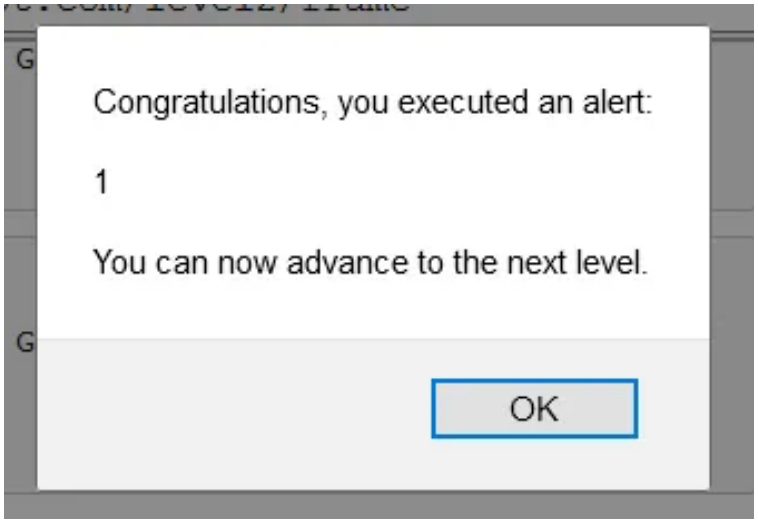

Figure 5: javascript alert

3.5 Task 5

To deepen your understanding of different levels of XSS, you should attempt the next few levels and see how far you get.

4 Conclusion

Cross-Site Scripting (XSS) vulnerabilities can be exploited to inject malicious scripts into web applications. By simulating a reflected or stored XSS attack, we demonstrated how attackers can execute unauthorized scripts in a user's browser, steal session cookies, redirect users, or manipulate webpage content.

This lab reinforced the importance of secure web development practices, such as input validation, output encoding, and proper use of security headers. Understanding how XSS works is essential for identifying and mitigating this common vulnerability during security assessments.

References

[1] XSS game