Experiment 10

AIM - Perform Cross-Site Scripting attack and analyze its impact on security

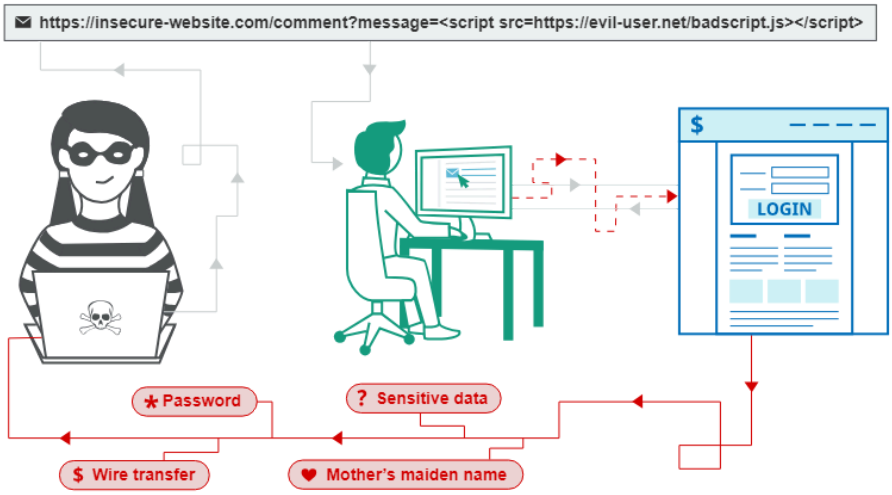
Learning Objective: Students should be able to perform XSS and analyze its impact on security.

Tools: Virtual Machine, Linux

Theory:

Cross-site scripting (also known as XSS) is a web security vulnerability that allows an attacker to compromise the interactions that users have with a vulnerable application. It allows an attacker to circumvent the same origin policy, which is designed to segregate different websites from each other. Cross-site scripting vulnerabilities normally allow an attacker to masquerade as a victim user, to carry out any actions that the user is able to perform, and to access any of the user's data. If the victim user has privileged access within the application, then the attacker might be able to gain full control over all of the application's functionality and data.

Cross-site scripting works by manipulating a vulnerable web site so that it returns malicious JavaScript to users. When the malicious code executes inside a victim's browser, the attacker can fully compromise their interaction with the application.



There are three main types of XSS attacks. These are:

- Reflected XSS
- Stored XSS
- DOM-based XSS

Reflected cross-site scripting :

Reflected XSS is the simplest variety of cross-site scripting. It arises when an application receives data in an HTTP request and includes that data within the immediate response in an unsafe way.

Stored cross-site scripting :

Stored XSS (also known as persistent or second-order XSS) arises when an application receives data from an untrusted source and includes that data within its later HTTP responses in an unsafe way. The data in question might be submitted to the application via HTTP requests; for example,

comments on a blog post, user nicknames in a chat room, or contact details on a customer order. In other cases, the data might arrive from other untrusted sources; for example, a webmail application displaying messages received over SMTP, a marketing application displaying social media posts, or a network monitoring application displaying packet data from network traffic.

DOM-based cross-site scripting

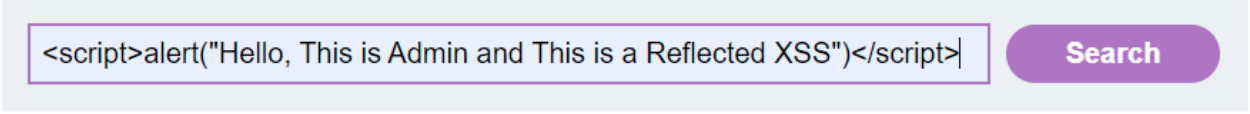
DOM-based XSS (also known as DOM XSS) arises when an application contains some client-side JavaScript that processes data from an untrusted source in an unsafe way, usually by writing the data back to the DOM.

Implementation :

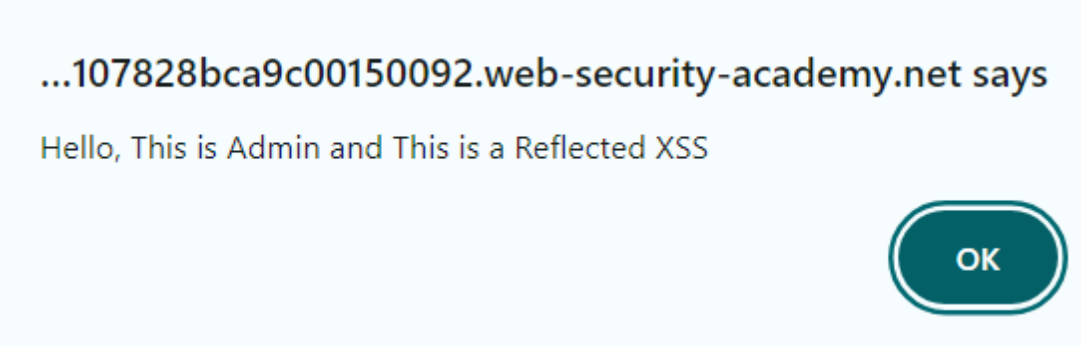
Reflected cross-site scripting :

Step 1 : Go to portswigger.net/web-security/all-labs#cross-site-scripting and click on the lab

Step 2 : In search box search **<script>alert("Hello, This is Umang and This is a Reflected XSS")</script>** and click on search



Step 3 : We get a popup of alert and no search results are found.

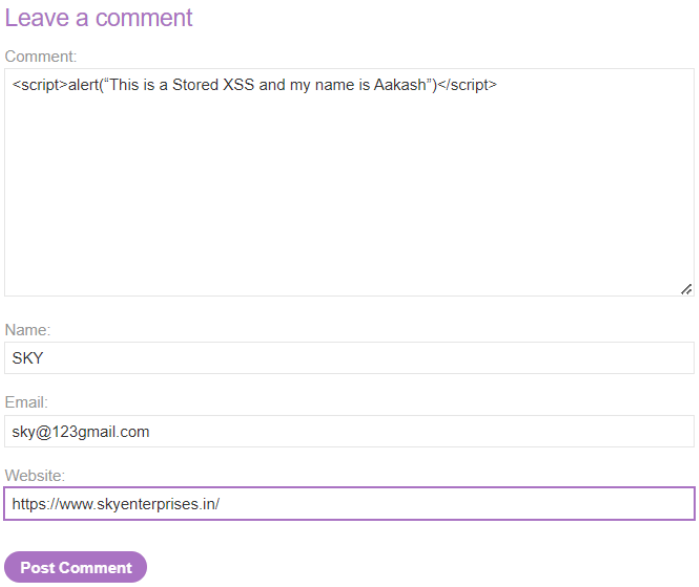


Stored cross-site scripting :

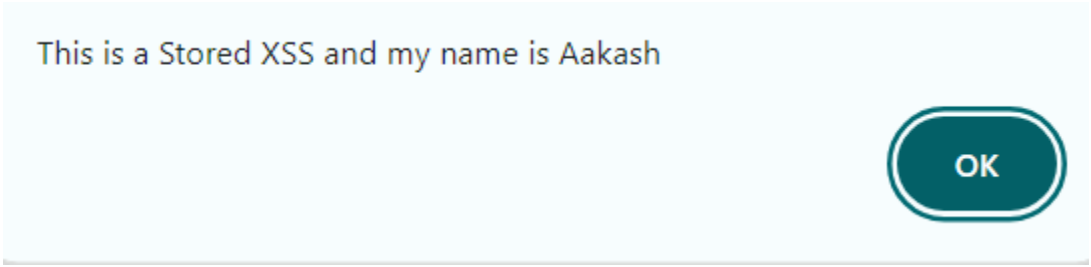
Step 1 : Click on View Post and go to the comment section

Step 2 : Enter the following into the comment box:

- **<script>alert(“This is a Stored XSS and my name is Aakash”)</script>**
- Enter a name, email and website.
- Click "Post comment".

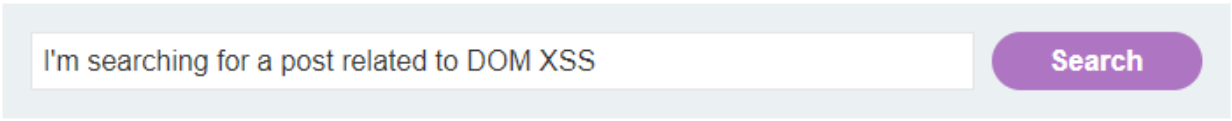


Step 3 : Go back to the blog and we notice a alert that we commented

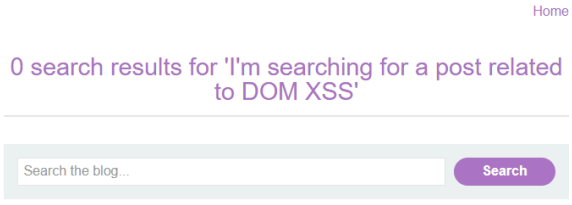


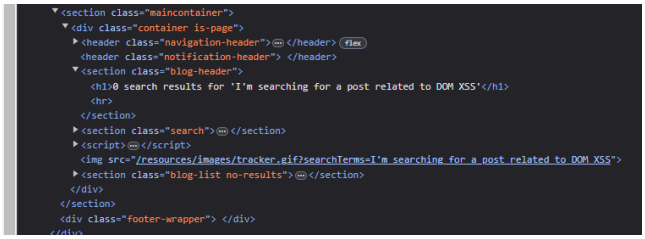
DOM-based cross-site scripting

Step 1 : Go to the website and search anything on it and click on Search

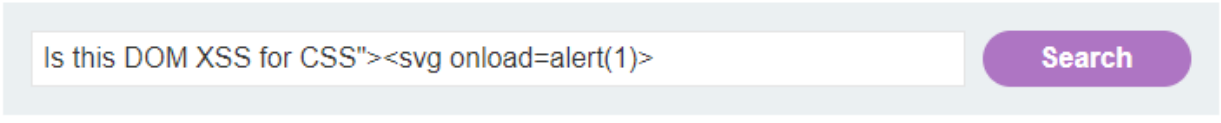


Step 2 : We see there are 0 search result and on Inspect we can see the content we searched is placed inside an image tag attribute

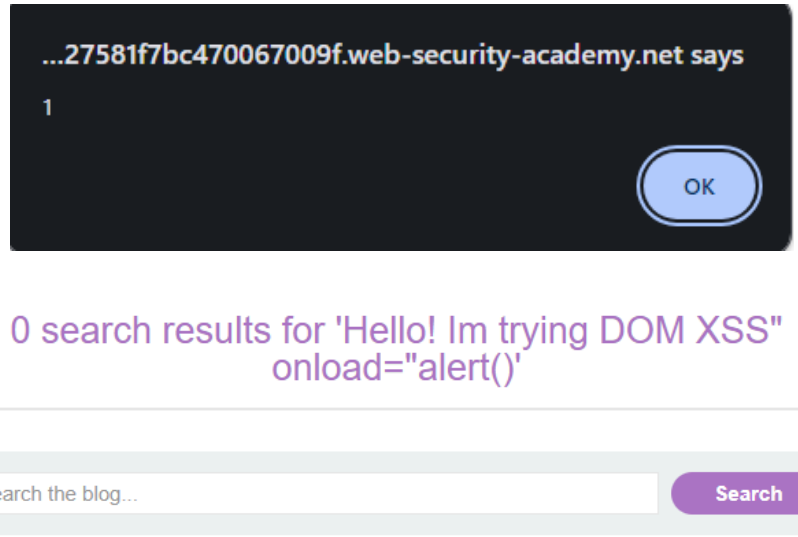




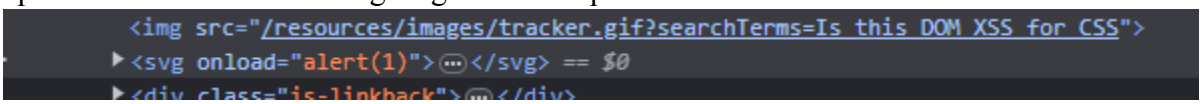
Step 3 : Breakout of the image tag and update as **Is this DOM XSS for CSS"><svg onload=alert(1)>** and click on Search



Step 4 : Alert pop up appears, click on OK and Inspect again



Step 5 : We notice that the image tag has been updated



Learning Outcomes:
The student should be able to understand and perform XSS
LO1: To describe & understand XSS Scripting
LO2: To perform XSS Scripting

Course Outcomes: Upon completion of the course students will be able to understand & perform Cross-Site Scripting

Conclusion:
From this experiment, we were able to understand what Cross-Site Scripting is, how it is done and the different types of Cross-Site Scripting. Hence we were able to successfully perform the different types of XSS.

For Faculty Use:

Correction Parameters	Formative Assessment [40%]	Timely completion of Practical [40%]	Attendance / Learning Attitude [20%]	
Marks Obtained				