Lab-9 Report

Akhilesh Gowda Mandya Ramesh

***Abstract*—The Lab-9 introduced us to Cross site scripting(XSS) and how it can be used to perform tasks that were not the original intent of the developer. Concepts such as Reflected XSS injection an DOM-based XSS injection were introduced.**

I. Introduction

THE insecure application that provides functionalities to observe the vulnerabilities of applications, such as WebGoat can be used to learn and get exposure to different types of vulnerabilities in the area of injection and cross site scripting such as Reflected XSS injection and DOM-based XSS injection.

II. Cross Site Scripting

Cross-Site scripting which is also known as XSS is a vulnerability or flaw hat combines the allowances of html or script tags as input that are rendered into a browser without encoding or sensitization. In Web security this is one of the most prevalent and pernicious issue. If not properly protected against XSS sensitive data can be lost such as authentication cookies and these sensitive information can be used for someone else's purpose. The task 2 in the XSS was to open another tab of the same page and check if the have the same cookie, The answer was yes they contain the same cookie. Most common locations of Cross site scripting is in Search fields that echo the string back to the user, inputs fields tha echo strings back to the user, error message that returns user supplied text, hidden fields that contain user supplied data, pages that display user supplied data such as message boards and free form comments and HTTP headers

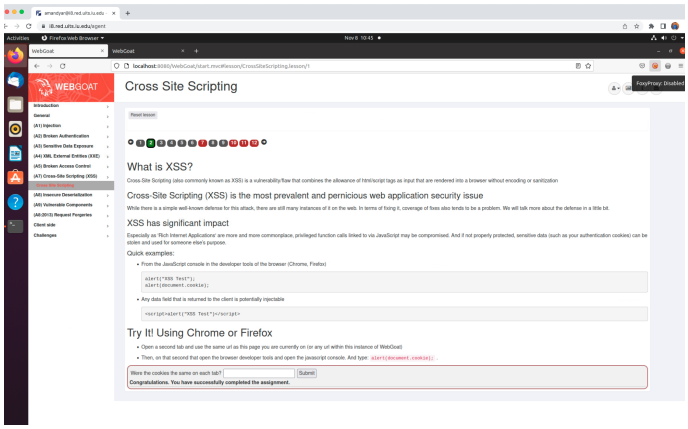

Fig. 1. XSS-task-2

III. XSS attacks effects

XSS attacks may result in stealing of session cookie, creating of false requests, creating false fields on a page to collect credentials, redirecting the page to a non-friendly site, creating requests hat masquerade as a valid user, stealing of confidential information, execution of malicious code on an end-user system also called as active scripting, insertion of hostile and inappropriate content.

IV. Types of XSS attacks

There are differen types of XSS attacks such as Reflected XSS where Malicious content is displayed in the web browser using a user request, malicious content is written into the page as a server response, social engineering is required, runs with browser privileges inherited from user in browser. DOM-based XSS which is also technically reflected where malicious content fom a user request is used by client-side scripts to write HTML to its own page, runs with browser privileges inherited fom user in the browser similiar to reflected XSS. Stored or persistent XSS is where malicious content is stored on the server and later displayed in the web browser, social engineering is not required here. The task 7 was to find the vulnerable field, credit card number field was vulnerable and i typed ¡script¿alert('XSS')¡/script¿ into the credit card field and the task was completed. The difference between DOM-based and refelcted XSS is that in DOM-based the payload will never got ot the server and it will only ever be processed by client. For task-10 we have to look at the javaScript source code o find out the route, after ispecting i found that the correct answer is start.mvctest/. Fot task 11 we need to activate webgoat.customjs.phoneHome() in a new tab using URL

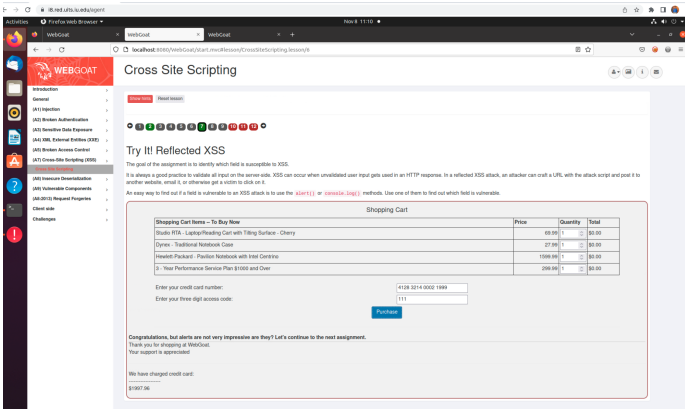

Fig. 2. XSS-task-7

Advisor: Luyi Xing, Jiale Guan.

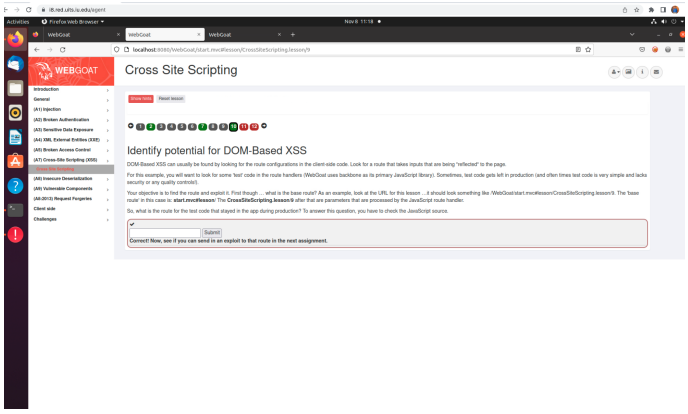

Fig. 3. XSS task -10

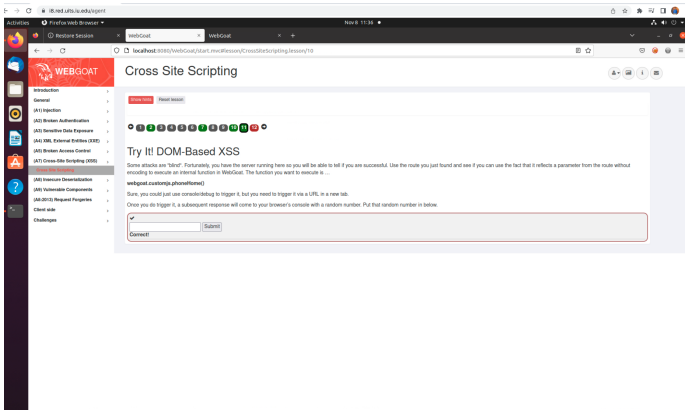

Fig. 4. XSS task -11

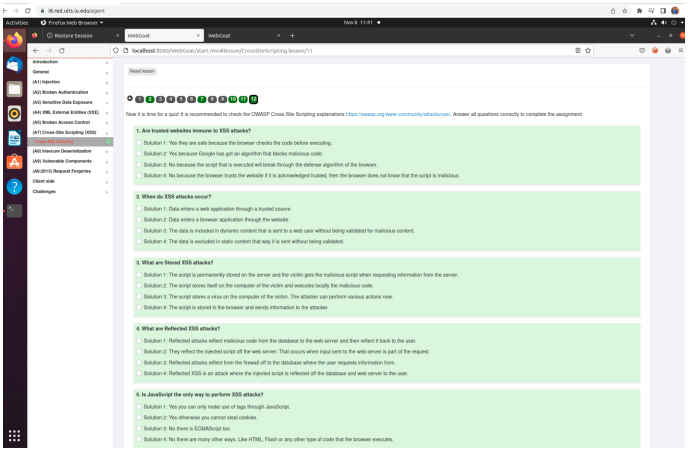

Fig. 5. XSS task -12 quiz completion

V. Conclusion

In Summary as WebGoats vulnerabilities can be exploited to study and bolster the attacks, We can also use it to study and mitigate the attacks such caused by SQL injection as the CIA Triads can be compromised by performing SQL Injection. This lab we studied XSS attacks, its effects and different types of XSS attacks. XSS attacks may result in stealing of session cookie, creating of false requests, creating false fields on a page to collect credentials, redirecting the page to a non-friendly site, creating requests hat masquerade as a valid user, stealing of confidential information, execution of malicious code on an end-user system also called as active scripting, insertion of hostile and inappropriate content. Hence we need to make sure that we keep our websites not vulnerable to XSS attacks

References

[1] https://github.com/WebGoat/WebGoat
[2] https://owasp.org/
[3] https://www.zaproxy.org