Lab-10 Report

Akhilesh Gowda Mandya Ramesh

***Abstract*—The Lab-10 introduced us to additional sections on Cross site scripting(XSS) and how it can be used to perform tasks that were not the original intent of the developer. where phishing, stored XSS attacks, cross site scripting and reflected XSS attacks 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 where phishing, stored XSS attacks, cross site scripting and reflected XSS attacks were introduced.

II. Phishing with XSS

Using Phishing in XSS we can attach HTMl to the request credentials, Add javascript to actually collect the credential. The task here was to post the credentials to http://localhost:8080/WebGoat/catcher?PROPERTY=yes. For the task to be completed the credentials has to be posted to the catcher servlet. I did so by enclosing the script tag inside a form tag and wrote a javascript function to complete the task as in to post the credentials to the catch servlet.

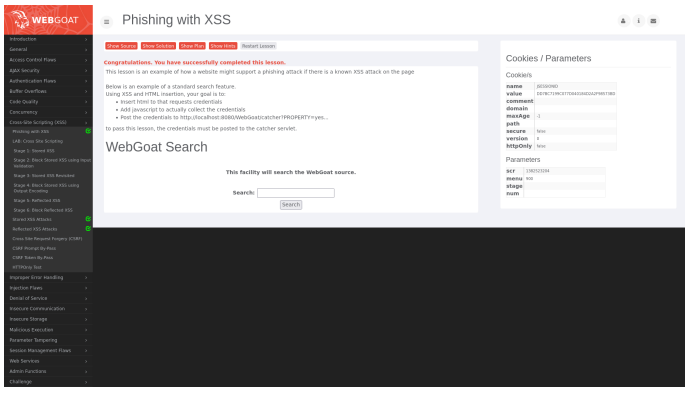

Fig. 1. phising with xss

III. Reflected XSS Attacks

Reflected cross-site scripting (or XSS) arises when an application receives data in an HTTP request and includes that data within the immediate response in an unsafe way. The given task here was to crack the url as an attacker using an attacker script and to post it in another website or to email it. I exploited the three digit access code field to complete the assignment i added a script tag and inserted an alert statement.

Advisor: Luyi Xing, Jiale Guan.

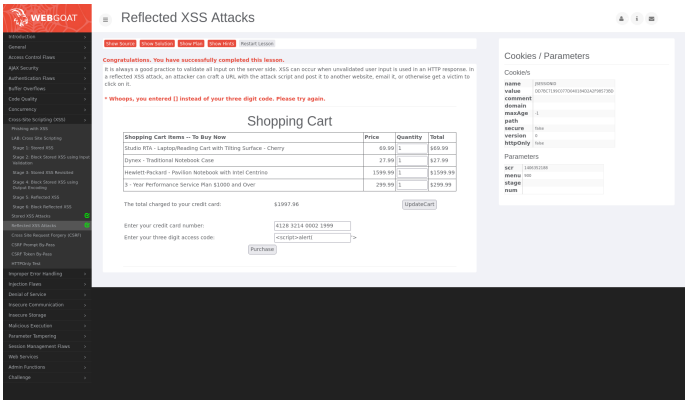

Fig. 2. reflected xss attacks

IV. Stored XSS Attacks

Stored cross-site scripting (also known as second-order or persistent XSS) arises when an application receives data from an untrusted source and includes that data within its later HTTP responses in an unsafe way. It is always a good practice to scrub all input, especially those inputs that will later be used as parameters to OS commands, scripts, and database queries. It is particularly important for content that will be permanently stored somewhere in the application. Users should not be able to create message content that could cause another user to load an undesirable page or undesirable content when the user's message is retrieved.

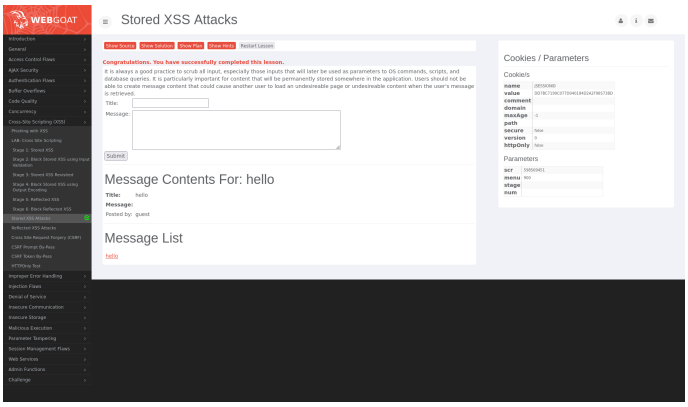

Fig. 3. reflected xss attacks

V. 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. Most common locations of Cross site scripting is in Search fields that echo the string back to the user, inputs fields thay 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. There were 3 tasks to be completed task 1,3,5. The task was similar to each other where we insert alert inside a script tag to complete the tasks

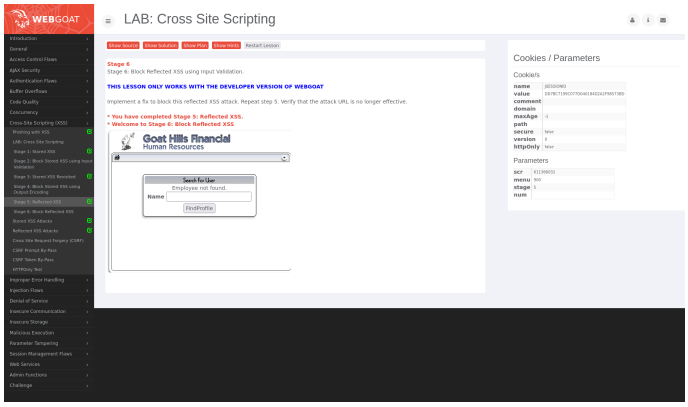

Fig. 4. Cross site scripting Tasks 1, 3, 5 completion

VI. Cross Site Request Forgeries

Cross site request forgeries happens usually with one click because of which it is also known as one click attack. Cross site request forgeries makes use of the trust the user has on a browser. it usually involves the sites that rely on a user's identity and exploits its trust. It tricks the user's browser by sending a HTTP request to a target site. It involves HTTP requests that have side effects.

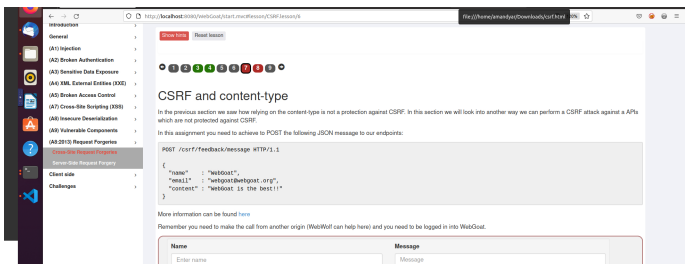

Fig. 5. Cross Site Request Forgeries task completion

VII. Webpage payload task

Tested several ways of inserting the payload methods such as scripts tag which has a src attribute where we can insert the alert statements or malicious codes. We can also make use of IMG tag which also has a SRC attribute in order to inject malicious code to the browser. For case sensitive attacks we can vary the cases of the javascript functions to confuse the browsers. There are many other tags which can be used similarly such as the anchor tag which is the a tag in the HTML. This way i used several methods from the cheatsheet to learn more about its functionality

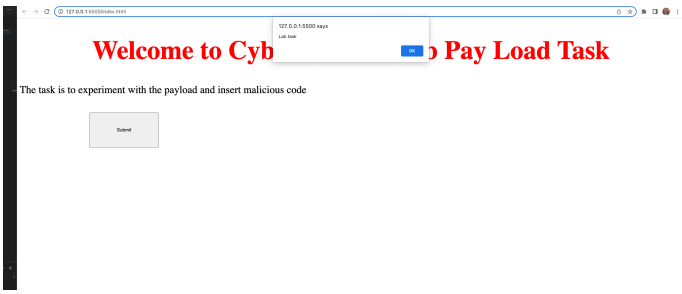

Fig. 6. Experimenting with HTML and Javascript code by trying different payloads

VIII. 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. We can additionally make use of the payloads to inject malicious code to the browser by making use of the src attribute in the html tags.

References

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