Assignment 2

Venkata Diwakar Reddy Kashireddy

***Abstract*—This report explains how to perform attacks on the types of vulnerabilities in the area of injection and authentication and how to perform XPATH Injection. The attacker accomplishes this by injecting malicious code into the application through a text field.**

I. Introduction

In this lab we were asked to perform attacks on the types of vulnerabilities in the area of injection and authentication: XPATH Injection. We learnt about injection flaws and performed the attacks on WebGoat 7.1.

II. Tools

• KVM (Kernel-based Virtual Machine): It is a full virtualization solution for Linux. [1]
• WebGoat: A deliberately insecure web application maintained by OWASP designed for teaching web application security concepts. [2]
• Overleaf: It is a collaborative cloud-based LaTeX editor that helps to create documents easily by providing standard formats. [3]
• GitHub: GitHub is an Internet hosting service for software development and version control using Git. [4]
• Red: – IU Research Desktop (RED) is a virtual desktop service for users with accounts on the Carbonate research supercomputer at IU. [5]
• ZAP: OWASP ZAP (Zed Attack Proxy) is an open-source web application security scanner. It is used by those new to application security and professional penetration testers. [6]
• Firefox DevTools: Firefox Developer Tools is a set of web developer tools built into Firefox. It can be accessed to inspect the web page. [7]

III. XPATH Injection

XPath injection is a type of security vulnerability that occurs when an application uses user-supplied input to construct an XPath query without properly validating or sanitizing the input. XPath (XML Path Language) is a query language used for selecting data from XML documents. XPath uses path expressions to select nodes or node-sets in an XML document. When XPath injection occurs, an attacker can manipulate the input in a way that allows them to retrieve sensitive information from the XML document or perform other malicious actions on the application.

IV. Injecting XPATH Attack on the WebGoat

To perform an XPath injection attack on WebGoat, I manipulated the user input in the ”user name” field by entering ”diwakar’ or 1=1 or ’a’=’a”. This input was passed to exploit the XPath query construction in a way similar to SQL injection. The single quote ’ served to end the original username string, and the ” or 1=1 and or ’a’=’a ” conditions were always true, bypassing any authentication checks.

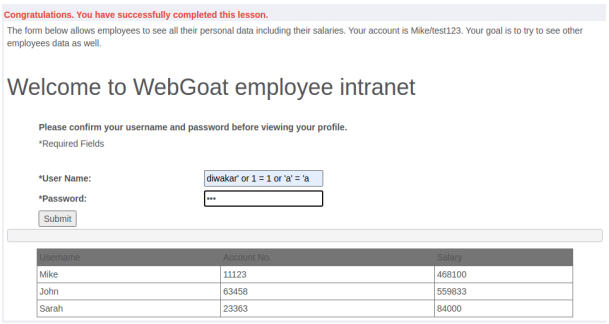

Fig. 1. Successfully performed XPATH injection to get user details.

By using these conditions, the XPath query returned all user values from the ”users” table, essentially ignoring any filtering logic. This shows the danger of XPath injection when applications do not properly validate and sanitize user input when constructing XPath queries, highlighting the importance of secure coding practices to prevent such vulnerabilities.

V. Conclusion

In this lab, XPath injection was performed in the context of injection and authentication vulnerabilities highlighted the significant risks associated with improper input validation and query construction. XPath injection, much like other injection attacks, poses a substantial threat to web applications. We demonstrated how an attacker can manipulate user input to form malicious XPath queries, leading to unauthorized access to sensitive data and bypassing authentication mechanisms.

Not only is it a threat easily instigated, it is also a threat that, with a little common-sense and forethought, can be almost totally prevented. This lesson showed us several examples of parameter injection. It is always good practice to sanitize all input data, especially data that will used in OS command, scripts, and database queries.

REFERENCES

[1] KVM
https://www.linux-kvm.org/page/MainPage..
[2] WebGoat
https://owasp.org/www-project-webgoat/..
[3] OverLeaf
https://www.overleaf.com/.
[4] Github
https://en.wikipedia.org/wiki/GitHub.
[5] RED
https://kb.iu.edu/d/apum.
[6] OWASP ZAP
https://www.zaproxy.org/
[7] Firefox DevTools
https://firefox-source-docs.mozilla.org/devtools-user/