Lab-7 Report

Akhilesh Gowda Mandya Ramesh

***Abstract*—The Lab-7 introduced types of vulnerabilities in the area of Injection and Authentication.**

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 authentication such as command injection, Log spoofing, XPATH Injection, Common authentication flaws. For this Lab task we will be using Webgoat 7.1 version

II. Command Injection

command injection is a serious threat to any parameter driven site. The methods behind the attacks are simple to learn but the amount of damage it causes is significant which sometimes might lead to system compromise. Despite this risk a significant number of systems on the internet are susceptible to this attack. Even though this threat can be easily prevented using common sense there are still systems that are open for this kind of attack. I converted the dropdown to text using Web developer tools and then entered ” and netstat -an and ipconfig then it retreived all the information and the

III. Log Spoofing

The Log Spoofing task was to login as an admin with the username field and password field i used aki with string ending with Login succeeded for username admin. This was the print log below prints login for aki failed but it prints login succeeded for username: admin. This way we can log spoof to get what is desired out of the fields. The below image depicts the task completion of the log spoofing task in webgoat 7.1

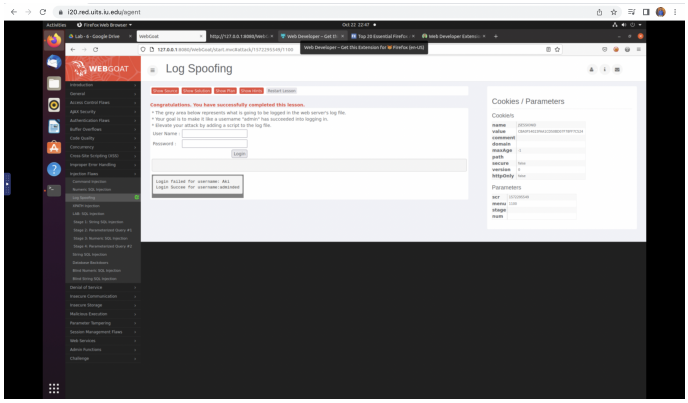

Fig. 1. Log Spoofing Task completion

Advisor: Luyi Xing, Jiale Guan.

IV. XPATH Injection

XPath Injection attacks are similar to SQL Injection attacks but occur when a web site uses user-supplied information to construct an XPath query for XML data. By sending intentionally malformed information into the web site, an attacker can find out how the XML data is structured, or access data that they may not normally have access to. They may even be able to elevate their privileges on the web site if the XML data is being used for authentication. for the Xpath injection task i entered Aki’ or 1=1 or ’a’=’a in the username field and i entered a random password. This was i was able to complete the task, The screenshot indicates the task completion validity.

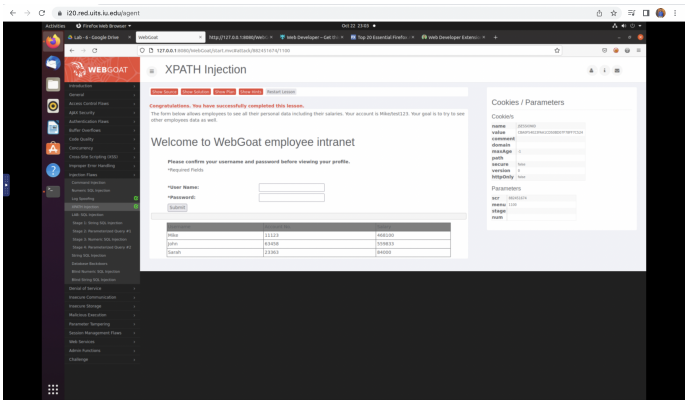

Fig. 2. XPATH Injection Task

V. Authentication Flaws

Authentication flaws are the flaws in authentication. The first task was to determine the password strength using a website. The amount of time it would take to crack that password. The second task was the forgot password task where we have to guess the favourite color of the admin to get his password. I kept guessing the password and eventually guessed it right and the colour was green. The next task was multilevel login -1. The task was to use the username jane and password tarzan and tan-1 is given. For the second attempt when tan-1 is already used we have to break in any way. I used the web developer tool to change the tan-2 to tan-1 and used the tan-1 code again to complete the assignment. The next task was to sign in as user Joe and his password and tan, But should then get the details of Jane. I once again used to web developer tool to alter the username to Jane before i entered the tan-2 code. This way i was able to complete multi-level login -2 task.

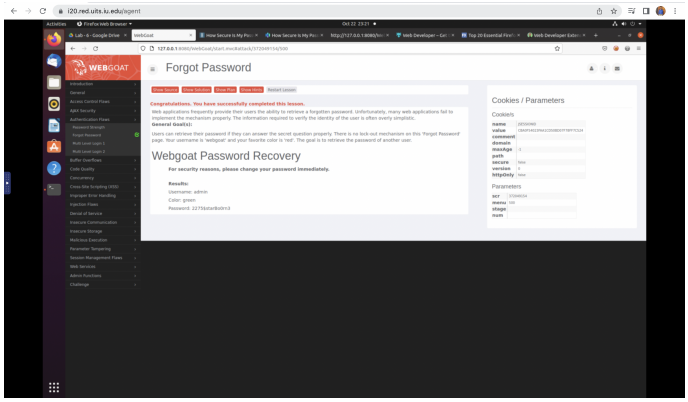

Fig. 3. authentication flaws-forgot password task

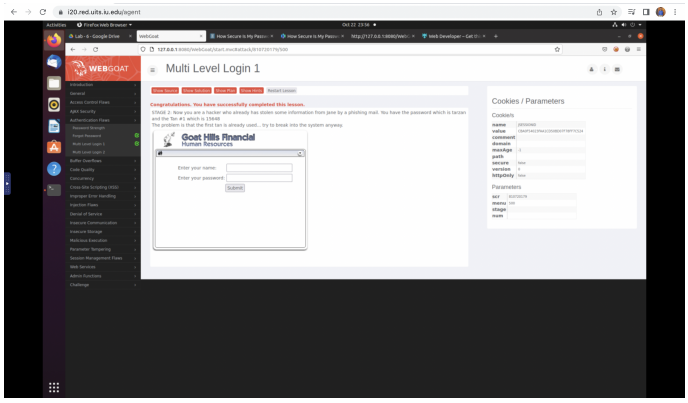

Fig. 4. multilevel login-1 completion

VI. Bonus Task: Content in /etc/passwd and /etc/shadow

etc/passwd file stores the important information needed for logging in. /etc/passwd file is a plain text file that contains a list of the system's accounts, which for each account useful information like user ID, group ID, home directory, shell, and more are contained. The /etc/passwd file should have general read permission as many command utilities use it to map user IDs to user names. However, write access to the /etc/passwd must only limit for the superuser/root account. The /etc/passwd contains one entry per line for each user (user account) of the system. All fields are separated by a colon (:) symbol. There are totally seven fields separated by : and those are username, password, userID, Group Id, User ID Info, Home directory and Command/Shell. our encrypted password hashes are in the /etc/shadow file. The /etc/shadow file contains one entry per line, each representing a user account. Each line of the /etc/shadow file contains nine comma-separated fields and those are username, encrypted password, Last password change, Minimum password age, Maximum password age, Warning period, Inactivity period, Expiration date, Unused which is reserved for future use. The /etc/shadow file should not be edited by hand unless we know what you are doing. Always use a command that is designed for the purpose. For example, to change a user password, use the passwd command, and to change the password aging information, use the chage command. The /etc/shadow file keeps records

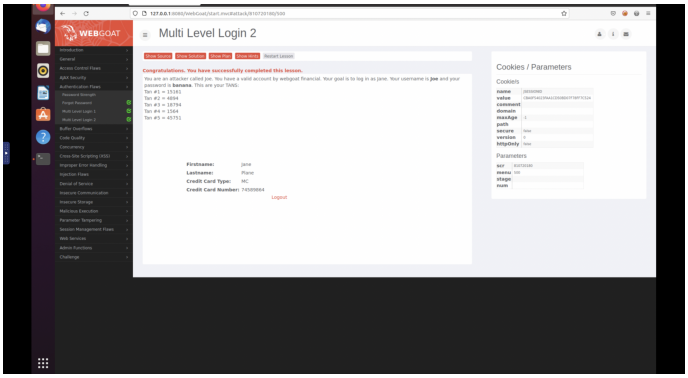

Fig. 5. multilevel login-2 completion

about encrypted users' passwords, as well as other passwords related information. A salt is added to the hashing process to force their uniqueness, increase their complexity without increasing user requirements, and to mitigate password attacks like hash tables. the unique hash produced by adding the salt can protect us against different attack vectors, such as hash table attacks, while slowing down dictionary and brute-force offline attacks. However, there are limitations in the protections that a salt can provide. If the attacker is hitting an online service with a credential stuffing attack, a subset of the brute force attack category, salts won't help at all because the legitimate server is doing the salting+hashing. Hashed passwords are not unique to themselves due to the deterministic nature of hash function: when given the same input, the same output is always produced. If Alice and Bob both choose nopassword as a password, their hash would be the same. The attacker can better predict the password that legitimately maps to that hash. Once the password is known, the same password can be used to access all the accounts that use that hash.

VII. 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 different types of vulnerabilities in the area of injection and authentication such as command injection, Log spoofing, XPATH Injection, Common authentication flaws. This was we can decide to try and develop a website that are less vulnerable to all of these. This way we can try to develop websites that do not compromise a CIA triads.

References

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