Lab-3 Report

Akhilesh Gowda Mandya Ramesh

***Abstract*—The Lab-3 introduced SQL injection to exploit the vulnerabilities in WebGoat.**

I. Introduction

THE insecure application that provides functionalities to observe the vulnerabilities of applications, such as WebGoat can be used to exploit its vulnerabilities using SQL injection. SQL injection typically happens when you request user input, such as their username or userid, and instead of receiving a name or id, the user instead provides you with a SQL statement that you will unwittingly execute on your database. In the subsequent sections we will learn how to perform SQL injection and different ways to do it.

II. SQL Injection intro (WebGoat 8.2)

SQL injection intro was completed on WebGoat 8.2, where the basics of SQL was introduced such as DDL, DML. The section started by writing basic SQL queries to extract information from the database.The task in the below figure was to display the department of the employee who's name is "Bob Franco"

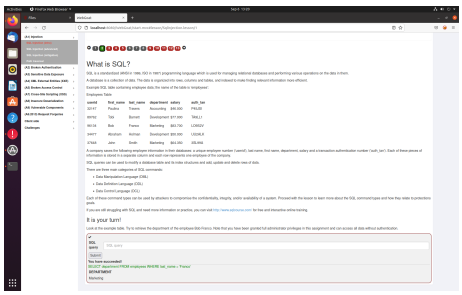

Fig. 1. SQL introduction by writing simple queries

Later we went on to explore string SQL Injection, Where the query is execute by joining strings making it vulnerable to SQL injection. The query was SELECT * FROM user data WHERE first name = 'John' AND last name = "Smith" or '1' = '1'; The 1=1 always evaluates to True, No mater the condition before it, hence we will be able to retrieve the data 8. Then we exploited the vulnerabilities of Numerical SQL Injection, where we concatenate numbers to form a query. The query concatenating numbers was SELECT * FROM user data WHERE login count =7 and userid = user or '1'='1'. Where the concatenating elements here are both numbers. The Numeric SQL Injection was performed by intercepting it through Zap and modifying it.

Advisor: Luyi Xing, Jiale Guan .

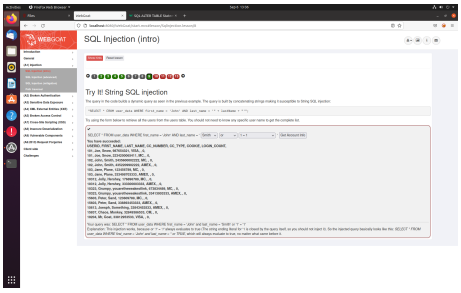

Fig. 2. sql injection joining strings

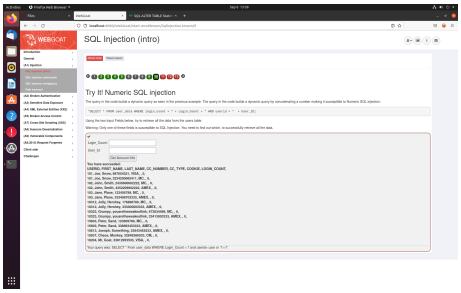

Fig. 3. Numeric SQL Injection

III. Compromising CIA triad using SQL Injection

Next we went on to explore how to compromise each aspect of the CIA triad using query chaining. Successful attacks from SQL injection can be used to view confidential information from the database such as credit card information. Successful attacks can compromise integrity to allow the attacker to change the information that they should not be able to access. There are different ways to violate, where the attacker can access the accounts and delete specific users making it unavailable for the users who have accounts in the systems.

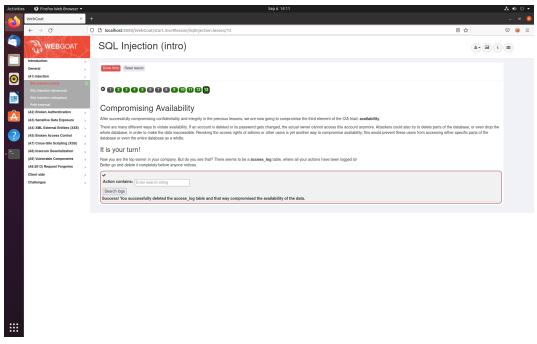

Fig. 4. Completion of SQL Intro

IV. WebGoat 7.1

WebGoat 7.1 was used to perform additional exercises that involve intercepting of requests through tolls such as ZAP, which also allows to modify the POST request. Numeric SQL Injection was performed on the webgoat 7.1 as well where we concatenate numbers in the query.

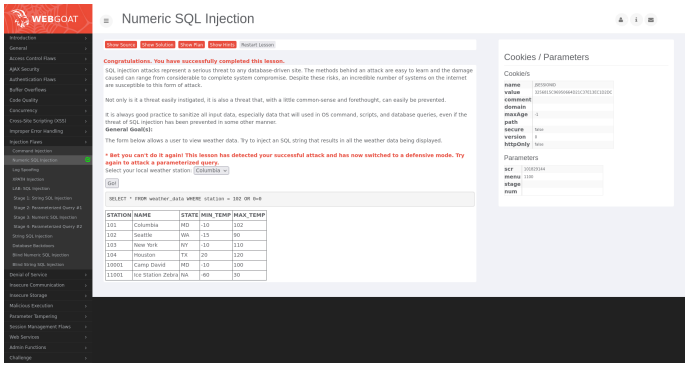

Fig. 5. Numeric SQL Injection With WebGoat 7.1

Further in a similar was string SQL injection was performed on the database to login through the login page as we did not know the password. Then we performed Numerical SQL injection on the database to view login profile Page of Neville by logging in as Larry. After which we explored database back doors where the stage-1 of the task was to update the salary of the employees and stage-2 was to create a trigger whenever a new employee is inserted, which updates the emails of employees. These types of attacks are very bad as they compromise the integrity of the system.

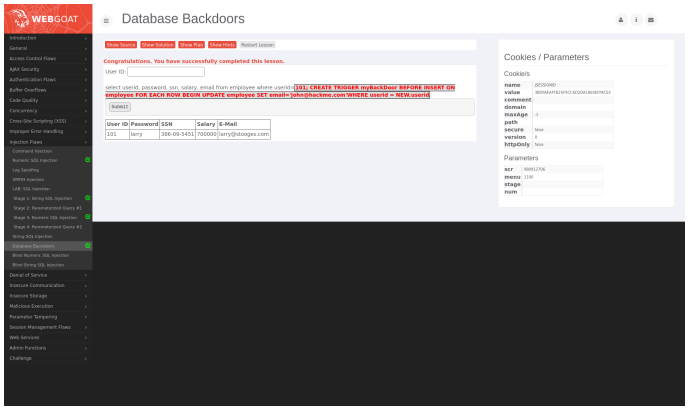

Fig. 6. Database Back Doors

V. Bonus: Blind Numeric SQL Injection

For Blind Numeric Injection i started of with a query that return or evaluates to True or False. Executed 100 AND ((SELECT pin FROM pins WHERE cc number='1111222233334444')¿10000) which results in an invalid message which suggests and infers that the value is not greater than 10000. Then checked for a lower bound and resulted that the value is between 1000-10000. Then checked for the next range which resulted in the value between 1000-5000. The same way o decreased the bound from the value evaluated to be between 2000-3000. Then i narrowed down to 2300-2400. The value at the end turned out to be 2364. This was we can exploit Blind Numeric SQL Injection by guessing and interpreting the values.

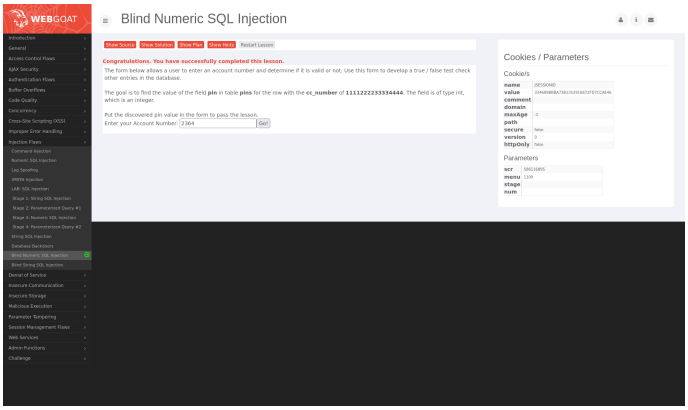

Fig. 7. Blind Numeric SQL Injection

VI. Conclusion

In Summary WebGoats vulnerabilities can be exploited to study and bolster the attacks. ZAP can be used to intercept HTTP requests, and the traffic can be changed using proxies. The CIA Triads can be compromised by performing SQL Injection in various was such as string SQL Injection and Numeric SQL injection. This can be very harmful or costly for the organizations as they will result in a very harsh damage. The Database backdoor is also a very harmful tool as it results in a trigger that can be triggered whenever a user performs a certain task without even him noticing

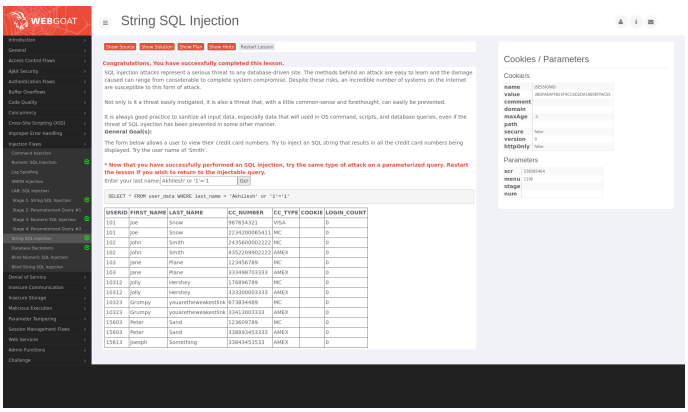

Fig. 8. Completion of All Tasks from Web Goat 7.1 screenshot

References

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