Lab-4 Report

Akhilesh Gowda Mandya Ramesh

***Abstract*—The Lab-4 introduced advanced techniques for 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 previous lab we had done basic SQL injection, This lab we took another step towards it by having and exposure to advanced techniques. Which includes Combing SQL injection techniques and Blind SQL injection.

II. SQL Injection Advanced (WebGoat 8.2)

SQL injection Advanced was completed on WebGoat 8.2. There was two topics that was introduced such as Combining SQL injection techniques and Blind SQL Injection. In the upcoming sections we will discuss about combining SQL statements and Blind SQL Injection.

III. Combining SQL statements

Multiple SQL statements can be combined to take advantages of the vulnerabilities. This can be done using SQL statements such as JOIN and UNION. JOIN Statements in SQL is used to combine rows from two or more tables, based on a related column. The UNION statement is required to combine the statements of two or more SELECT statements. The union ALL syntax allows duplicate values. The lab task was to get the account information by using union statements in the SQL Injection. The query used was ' UNION SELECT 1, username, password, cookie, 'A', 'B', 1 from usersystemdata. This retrieves userid , username, password, cookie from the tables.

IV. Blind Numeric SQL Injection

Blind SQL Injection is a type of SQL Injection that validates the database with TRUE or FALSE queries, based on this response we will be able to guess the answers.The blind Numeric SQL injection is when we are expecting a number or numeric value as an output. This kind of attacks can be done when the application is designed to display generic error messages.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')

Advisor: Luyi Xing, Jiale Guan.

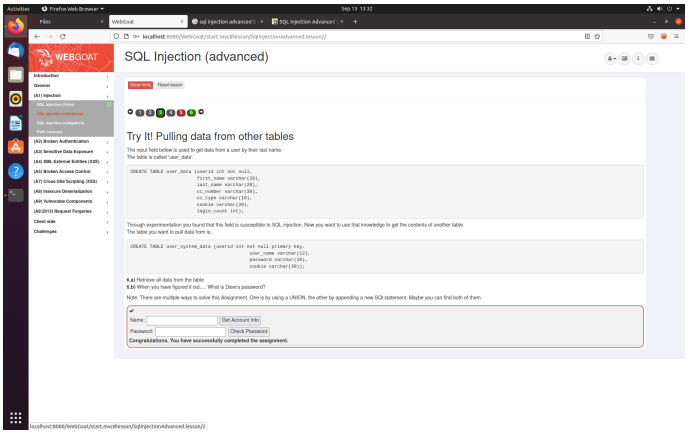

Fig. 1. Task 1 completion with UNION

greater than 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.

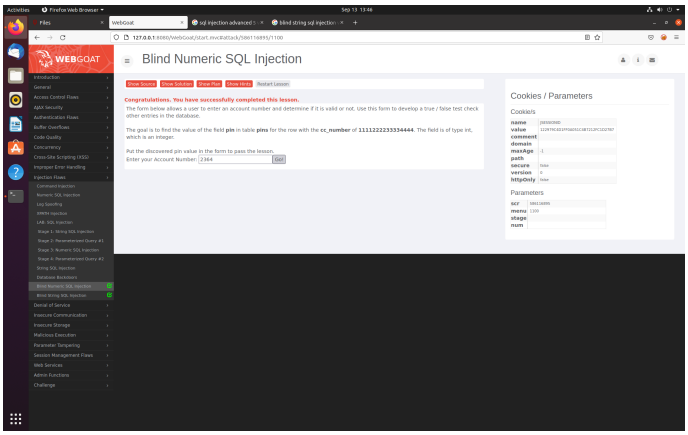

Fig. 2. Blind Numeric SQL Injection

V. Blind String SQL Injection

The Blind String SQL injection is done when we are expecting the string as an output. The typical example would be to guess the username of a particular field. The blind String SQL Injection can be done in a similar way to that as numeric one, its just that we test or query in the range of an Alphabet

for each position. Here in the lab task the sub-string syntax of the query was used to test the upper and lower bound of the alphabet. 101 AND (SUBSTRING((SELECT name FROM pins WHERE cc number='4321432143214321'), 1, 1) less than 'L'). was the query used to check the bound for the same character. Once we get the first character we can change the sub-string parameter to 2 to check for the second character and at the end the resulting name would be "Jill".

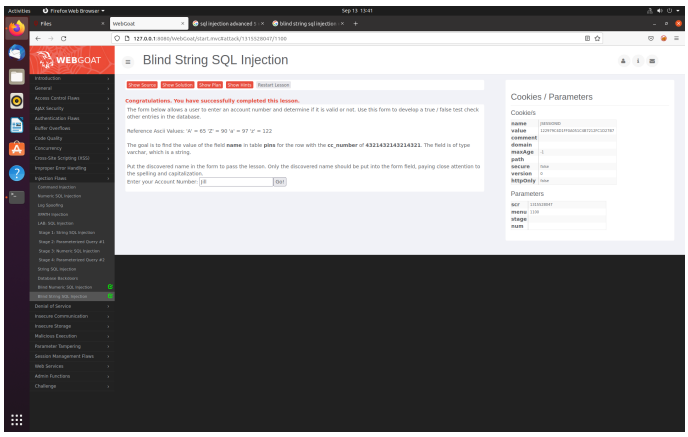

Fig. 3. Blind String SQL Injection

VI. Conclusion

In Summary WebGoats vulnerabilities can be exploited to study and bolster the attacks. The CIA Triads can be compromised by performing SQL Injection in various was such as string SQL Injection and Numeric SQL injection and also by combining SQL statements with UNION or JOIN. This can be very harmful or costly for the organizations as they will result in a very harsh damage. This was we can used the TRUE or FALSE result from SQL Injection technique to appropriately guess the range of the user input, leading a way to get the value. This can be avoided by making the application to not display the generic error output, without which we cannot guess the value using SQL injection. We can also write scripts/programs that can run to perform SQL Injection.

References

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