Ethical Hacking

Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 52
Part 1: SQL Injection Authentication Bypass

In this session, we will discuss about SQL Injection. SQL injection is a type of an injection attack that makes it possible to execute malicious SQL statements. These statements control a database server behind a web application. Attackers can use SQL injection vulnerabilities to bypass application security measure. They can go around authentication and authorization of a web page or web application and retrieve the content of the entire SQL database.

They can also use SQL injection to add, modify and delete records in the database and SQL injection vulnerability may affect any website or web application that uses an SQL database such as MySQL, Oracle, SQL server or others. Criminals may use it to gain unauthorized access to a sensitive data, customer information, personal data, trade secret, intellectual property and many more. SQL injection attacks are one of the oldest most prevalent and most dangerous type of web application attack. The OWASP organization which full form is: Open Web Application Security Project, list injection in their OWASP top 10 2017 document as the number 1 threat to web application security.

Now, I am discussing how and why is an SQL injection attack performed. To make an SQL injection attack, an attacker must first find vulnerable user inputs within the web page or web application. Web page or web application that has an SQL injection vulnerability uses such user input directly in an SQL query. The attacker can create input content, such content is often called a malicious payload and is the key part of the attack.

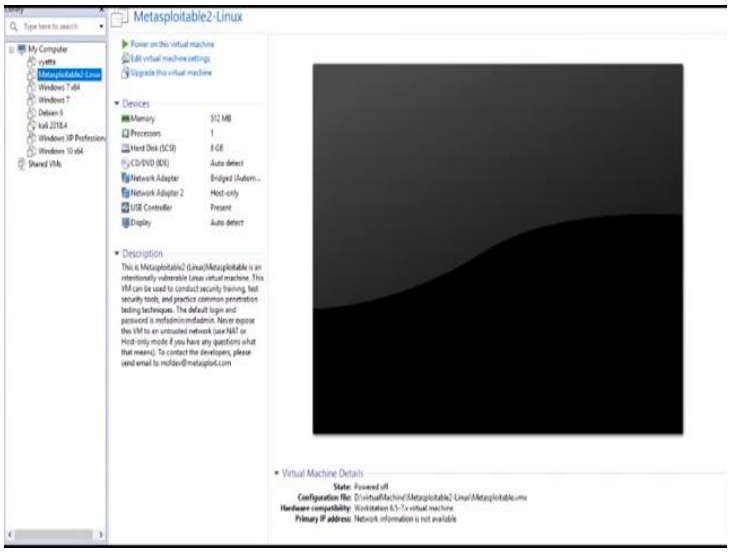
After the attacker sends this content, malicious SQL command are executed in the database. SQL is a query language that was designed to manage data stored in relational database. You can use it to access, modify and delete data. Many web applications and websites store all the data in SQL database. In some cases, you can also use SQL commands to run operating system commands. Therefore, a successful SQL injection attack can have very serious consequence.

Attackers can use SQL injections to find the credential of other users in the database; they can then impersonate these users. The impersonated user may be a database administrator with all database privileges. SQL lets you select and output data from the database. An SQL injection vulnerability would allow the attacker to gain complete access to all data in a database server. SQL also led to alter data in a database and add new data; for example, in a financial application an attacker could use SQL injection to alter balance, void transactions or transfer money to their account.

You can use SQL to delete records from a database, even drop tables, even if application availability until the database is restored; also backups may not cover the most recent data. In some databases servers, you can access the operating system using the database server; this may be intentional or accidental. In such case, an attacker could use an SQL injection as the initial vector and then attack the internal network behind the firewall. There are several types of SQL injection attack are available like SQLi using database error or union command, blind SQL injection, authentication bypass.

So, I will show you some of the SQL injection attack and starting from the bypass authentication. Now in this part, first we will discuss about authentication bypass using SQL injection. First we will discuss about some SQL query. So, to run SQL query we will use a operating system Metasploitable2 where some web application are hosted which are use some SQL database. So, first I am showing you some SQL query which related to SQL injection attack mainly authentication bypass attack using that particular operating system Metasploitable2.

(Refer Slide Time: 06:18)

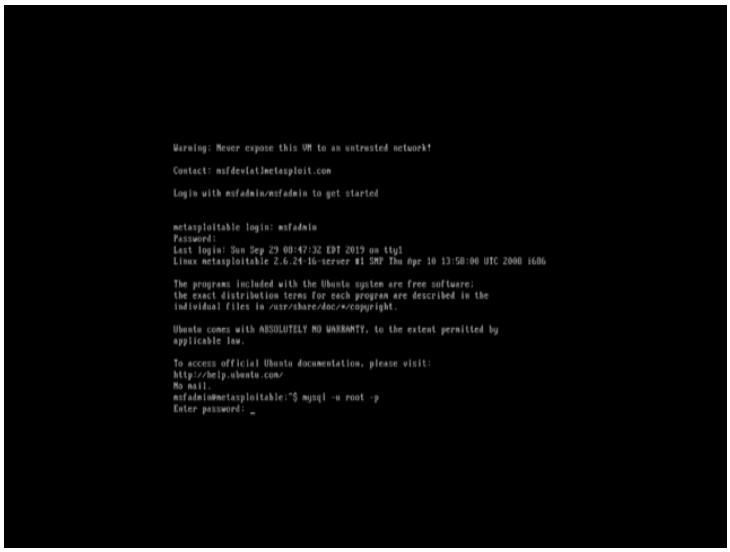


So, now open the metasploitable2, it is a Linux operating system.

(Refer Slide Time: 06:49)

Metasploitable login user id is msf admin and password is also msf admin.

(Refer Slide Time: 07:00)

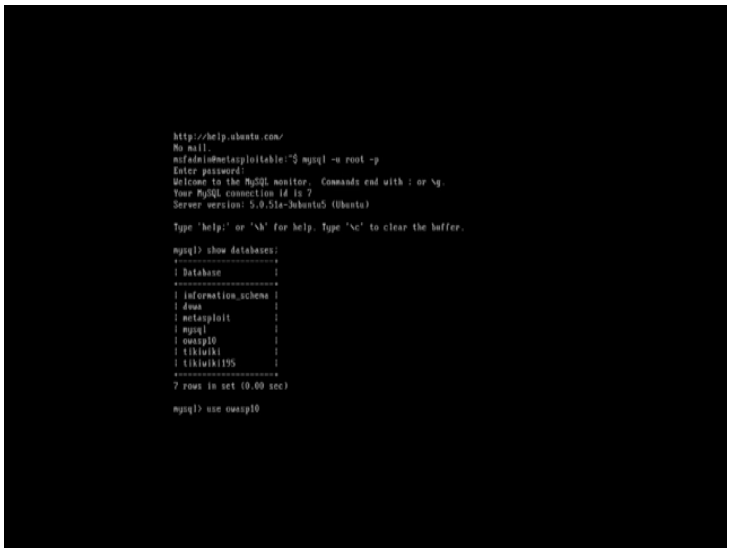


So, to run SQL query, first we need to go to the SQL prompt. So, to go to my SQL prompt, we need to use the command **mysql**, then -u specify the username that is root and -p is for password. So, it asking for the password and there is no password actually. So, just hit an enter.

(Refer Slide Time: 07:34)

So, now I am inside the mysql prompt. So, now, to see all the available databases, we need to use the command **show databases** and then to terminate the query use semicolon.

(Refer Slide Time: 08:01)



So, few databases are available here like dvwa, metasploit, mysql, owasp10, all these. So, now, suppose we want to check a particular database, suppose owasp10. So, first we need to use that particular database. So, to use a particular database, we need to use the command **use** then database name owasp10, then semicolon.

(Refer Slide Time: 08:42)

So, database changed; now to check all the available table inside this database, we need to use the command **show tables**. So, all these tables are available; accounts, blogs_table, captured_data, credit_cards, hitlog, then pen_test_tools. Further to check a

particular table, we can use the **select** command; to select all the row inside a particular table, we can use the command **select * from <table name>**. So, suppose now we want to check the table accounts. So, **select * from accounts**.

(Refer Slide Time: 09:58)

Now, see we got all the row inside the table accounts. Now, suppose we want the result for a particular username and password. So, we can use the command **select * from accounts where username** = suppose “test” **and password =** suppose “12345”, then use the semicolon.

(Refer Slide Time: 11:01)

So, this is an empty set. Why? Because no valid credential is there; so that is why this is an empty set. So, there is no username test and password 12345; so, that is why we got a empty set. Now, suppose further you want to select that particular row where username is admin and password is adminpass. So, use the command **select * from accounts where username = "admin" and password = "adminpass"**. So, there is an entry with this particular username and password. So, we got the result; so, that is why it returned valid entry from the database.

Now, suppose we do not have any idea about the username and password. So, how can we find out some result without knowing the username and password? So, suppose we use the command **select * from the table name accounts where username = "any"**. So, I think there is no such username and use **or** operation **1 = 1**. So, here **1 = 1** is always true and we add two condition **username = "any" or 1 = 1** by using the **or** operator. So, **1 = 1** is always true. So, the condition is always true; so that is why it gives us all the result from the table accounts.

(Refer Slide Time: 13:52)

See this is an malicious query; it returns all the entry from that particular table accounts. Now, I am showing you another query, **select * from table name accounts where username = "any" or 1 = 1;**, then use **# and password = "123456"**, then semicolon.

(Refer Slide Time: 15:03)

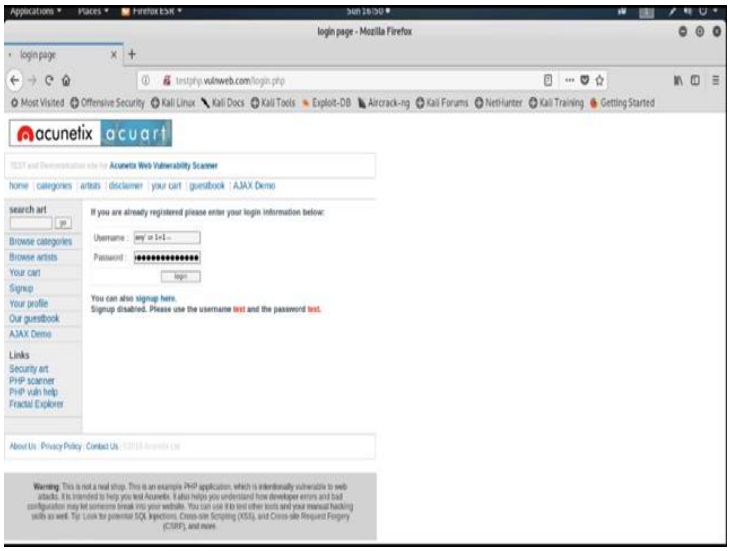
See it will also gives us all the result. So, basically hash is used to terminate the query. So, where you use the hash after that nothing is executed. So, in this query it only execute **select * from accounts where username = “any” or 1 = 1**. So, we use this concept further from a web application form where we need to keep some valid credentials. Now we can also limit our result by using limit; **select * from accounts where username = “any” or 1 = 1 limit 1**; see it limit the result in one entry.

So, if you use limit 2, then it will limit the result in two entry. So, this way we can also restrict our result with the number of entry.

(Refer Slide Time: 16:21)

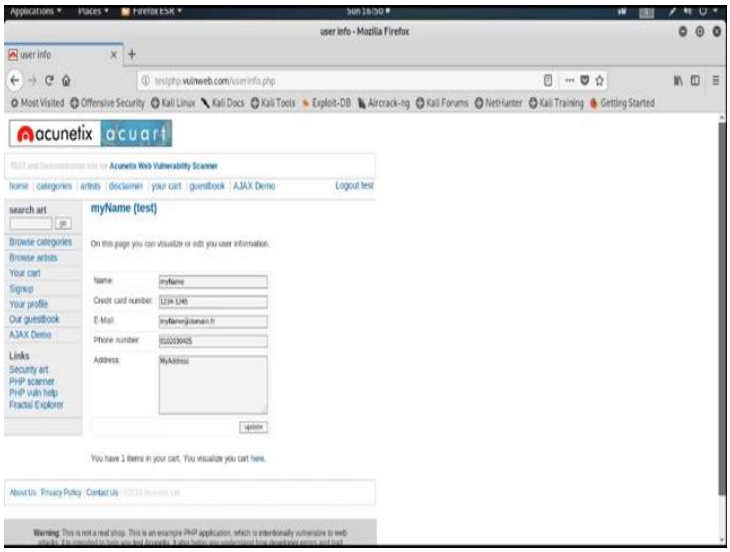
Now, we use this concept; now particular web application form to bypass the authentication. Now open our test web application **testphp.vulnweb.com**.

(Refer Slide Time: 17:07)



Now, go to sign up and it asking for username and password. So, put the username **any** then **" or 1 = 1**, then -- space that will also terminate the query; put the same thing in the password field also **any" or 1 = 1 --** .

(Refer Slide Time: 18:00)



Then hit enter and see successfully we are able to login inside the account. So, this way we can bypass the authentication in a particular web application and we can penetrate inside the web application. Further we will show you how to use the error based SQL injection.

Thank you.