Exploit a service using Metasploit

7/11/25 1:41 AM

Setting up Metasploit :

Step 1 : Start PostgreSQL DB use msfdb init command to start the DB

Metasploit Framework uses a PostgreSQL database to store and manage large amounts of information during a penetration test. And even in order to keep track

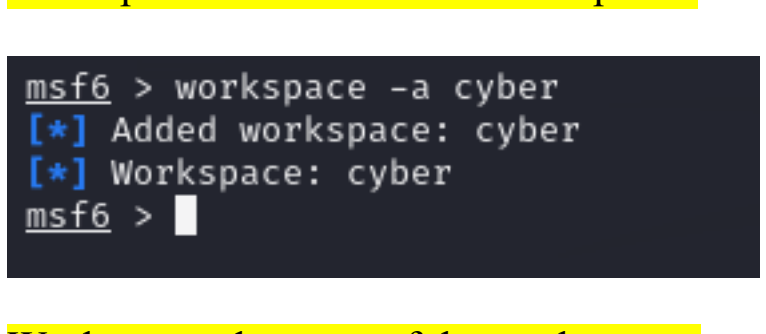
db_status --> Shows if Metasploit is connected to the DB

Step 2 : sudo msfconsole to start Metasploit.. If u don't want banner use -q flag

Step 3 : creating a workspace -- >A **workspace** is like a **project folder inside Metasploit's database** that stores all your data for a specific engagement or lab.

Command :

Workspace -a <name of the workspace> -- > for creating a workspace



Workspace -d <name of the workspace> -- > for deleting a workspace

Workspace <name of the workspace> -- > selecting workspace

Workspace -- > will show you which workspace is selected..

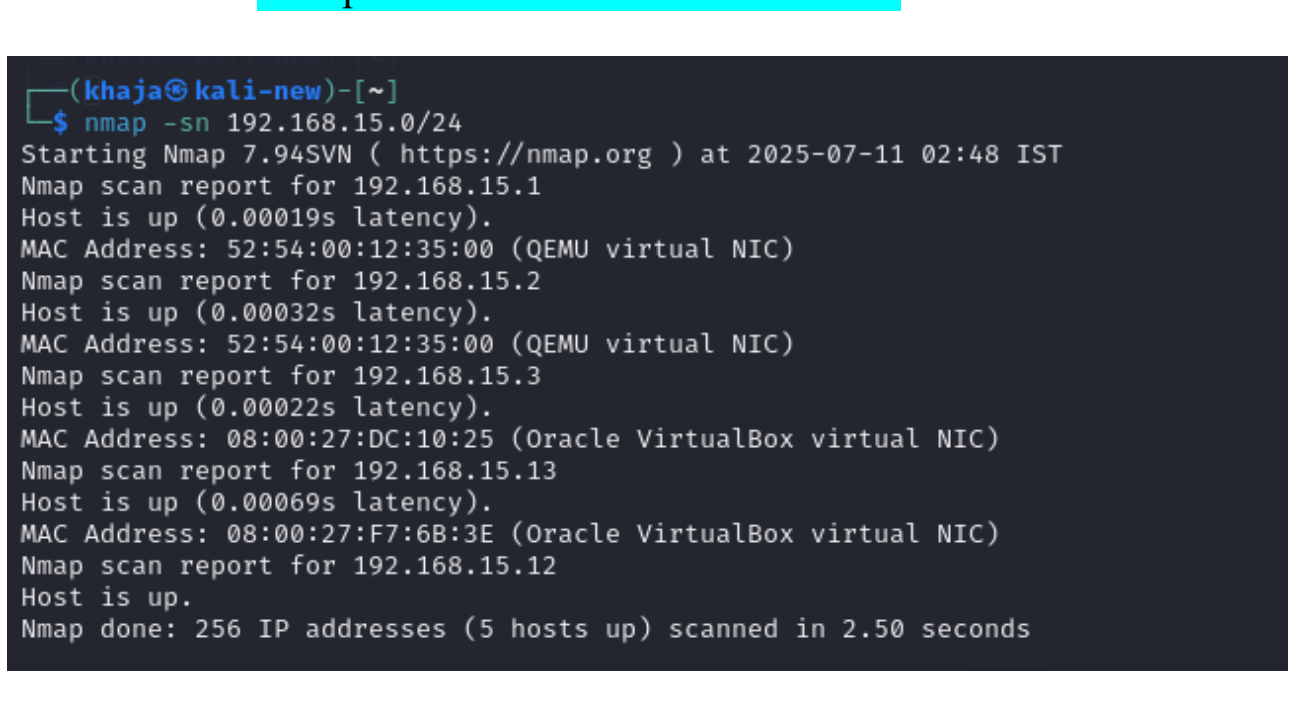
Step 4 : search for any exploit , payload , auxiliary by using search command

Syntax -- > search <keyword> -- > it will search that keyword in module

Exploiting a service:

Step 1 : identifying available host in the network using Nmap scan

Command -- > nmap -sN <network address/subnet>



Step 2 : identifying all open port

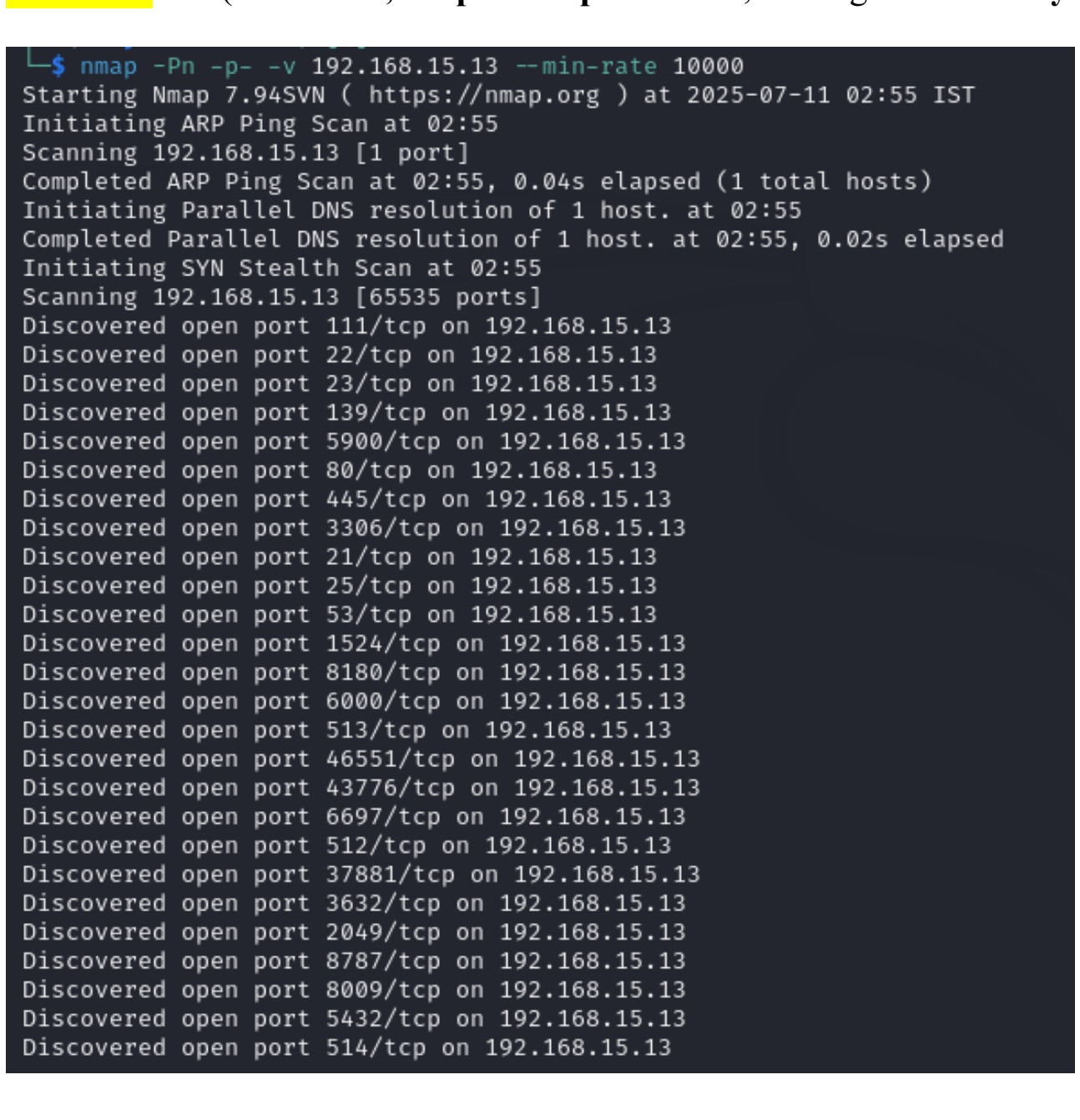
Command -- > nmap -Pn -p- -v 192.168.15.13 --min-rate 10000

-Pn --> don't send icmp request consider all the host are live.

-p- -- > for all port from **1 to 65535**

-v **-- >** verbose (**show more detailed output** while the scan is running)

--min-rate -- > (**at least 10,000 packets per second**, making the scan **very fast** and **aggressive**.) doest scan like this in real world



Step 3 : identifying the vulnerable service running

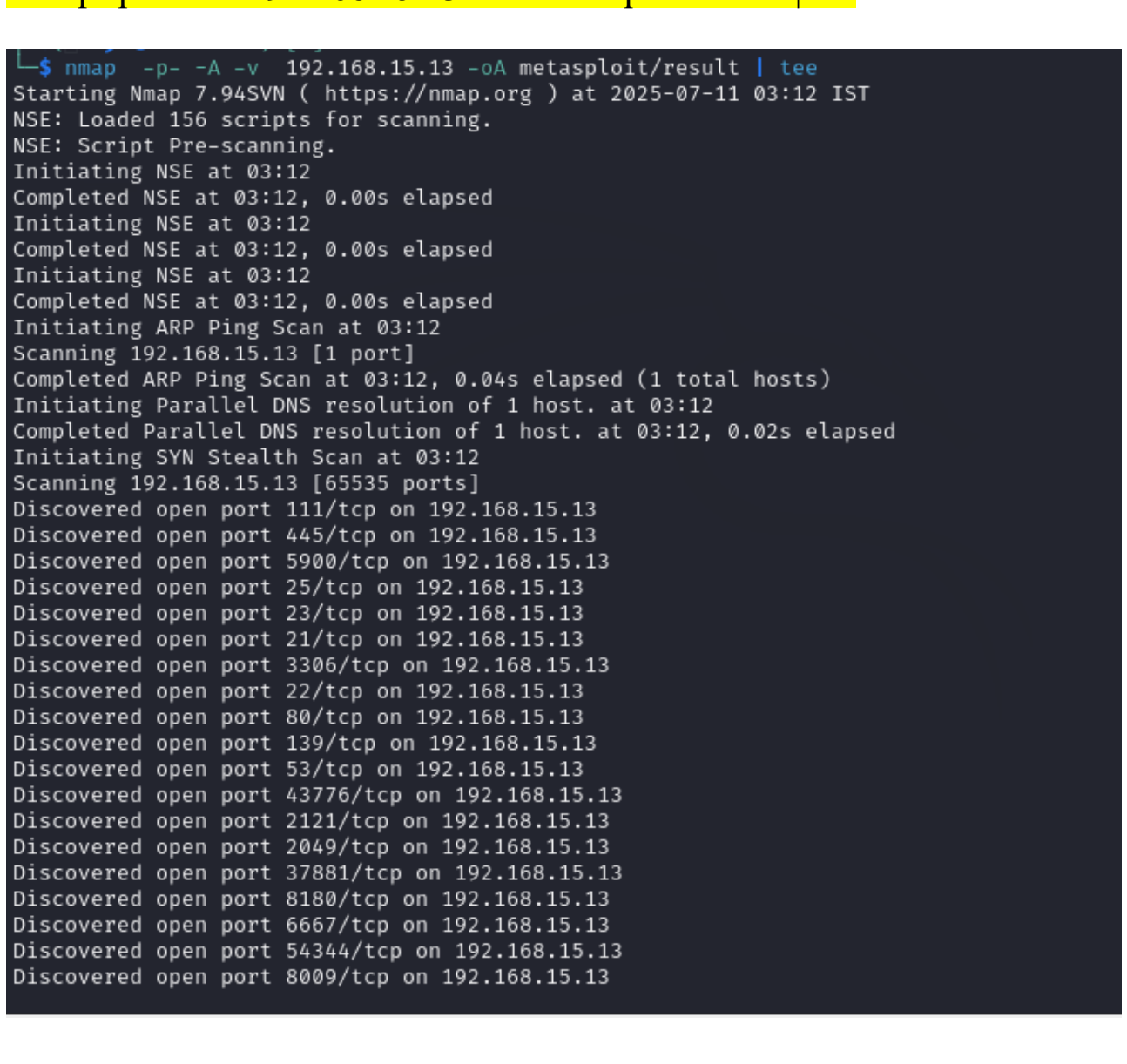
Command -- > nmap -p111,21,22,139…. Etc -sV <ip address> (this method is safer in real world for evading security solutions)

But for lab scenarios we can use -- > nmap -p- -A <ip>

-p- -- > for all port

-A -- > aggressive scan include (-O -sV -sC)

Nmap -p- -A -v 192.168.15.13 -oA metasploit/result | tee



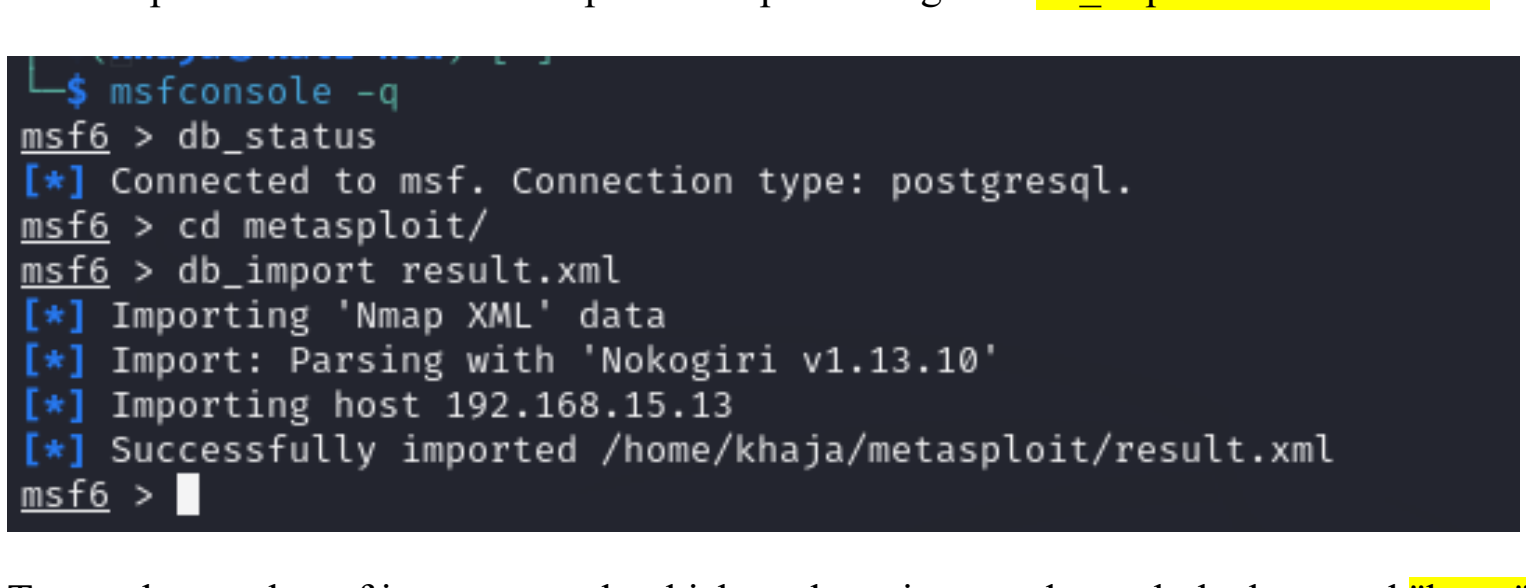
Saving the result in result files.. Tee is used to save the output and Displays the output live in the terminal

Note : we can import Nmap scanning result in Metasploit

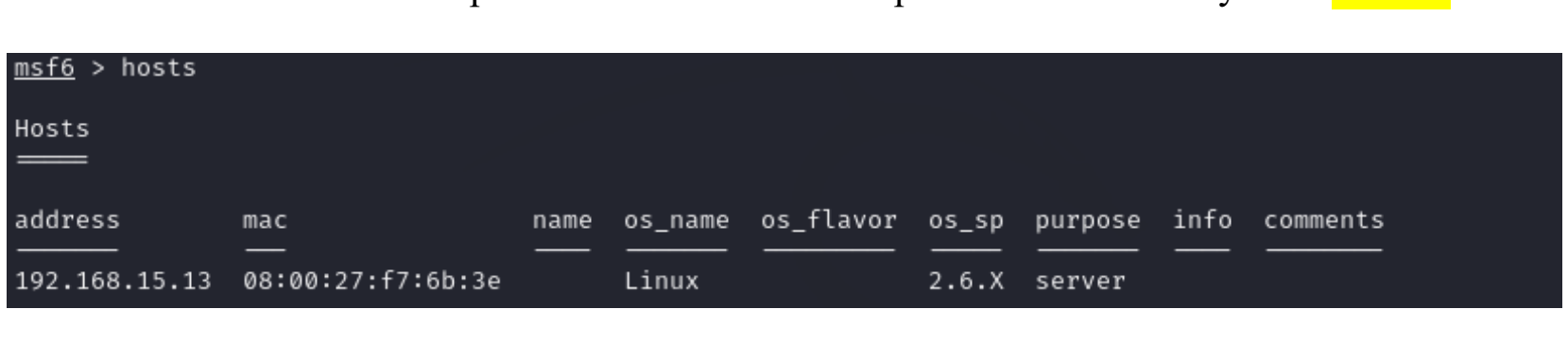
Step 4 : Integrating Nmap result in Metasploit

Settup the metasploit by looking the above steps

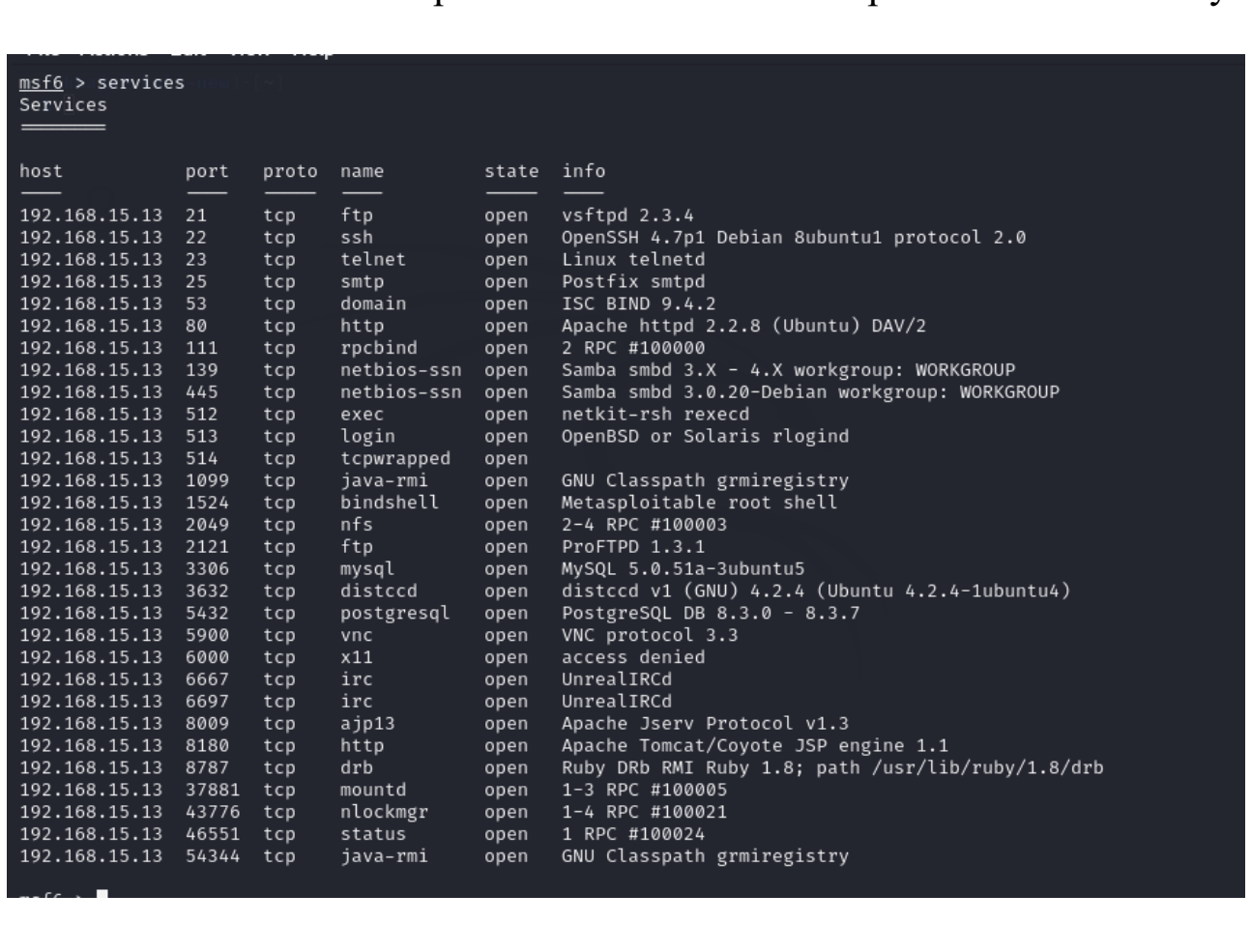
Then import the result file of nmap in metasploit using -- > db_import "filename.xml"



To see the number of in nmap result which we have imported search the keyword "hosts":



To see all services in nmap result which we have imported search the keyword "services" :



Step 5 : finding exploits for the services which we have got from nmap result

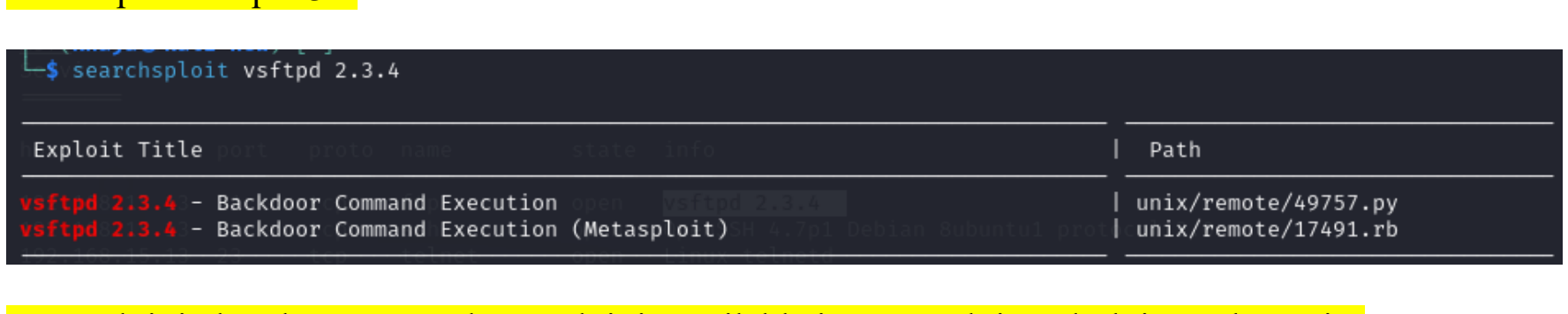
We have to find it manually which resources like exploitDB , GitHub , searchploit (cli of exploitDB)

Using searchploit :

Syntax -- > searchsploit <service version>

Example : finding the exploit of the first service in nmap result which is vsftp 2.3.4

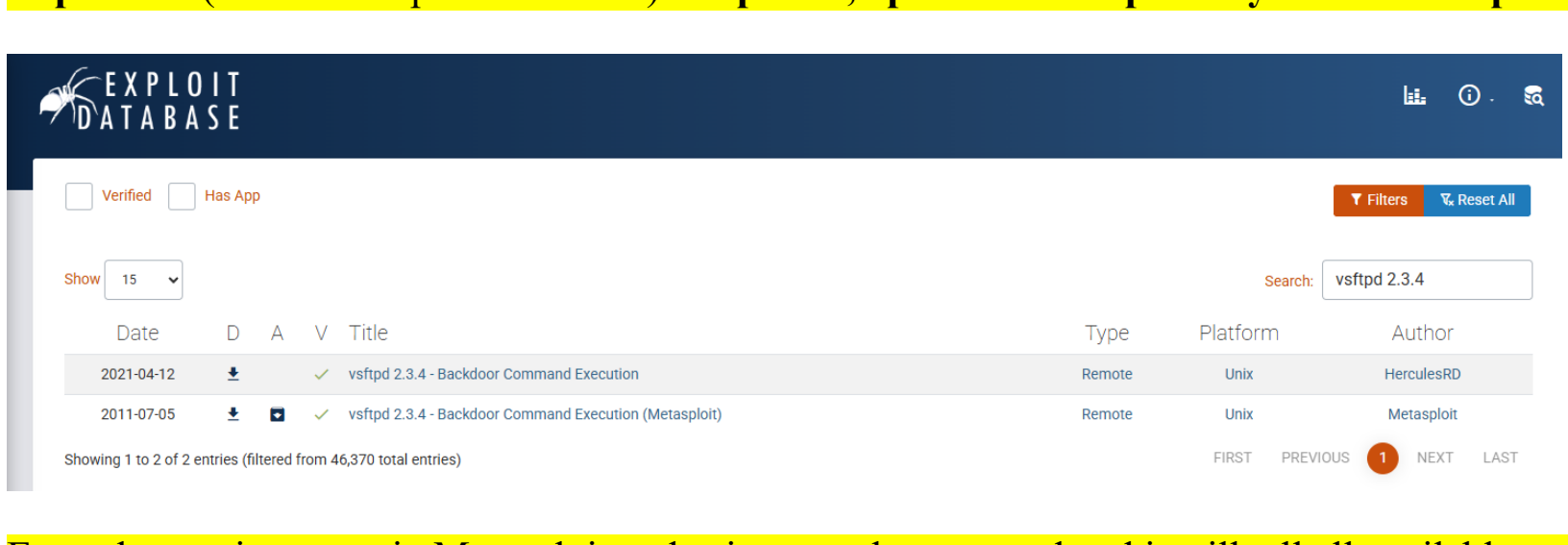
searchsploit vsftp 2.3.4



Metasploit in brackets means that exploit is available in Metasploit and .rb is a ruby script

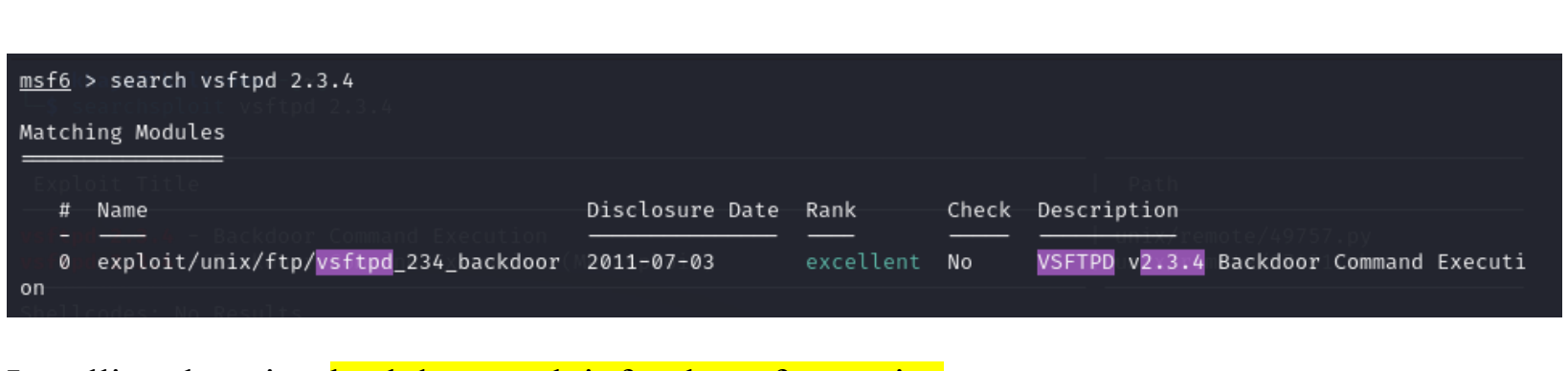
Using exploitDB :

ExploitDB (short for Exploit Database) is a **public, open-source repository of known exploits and vulnerabilities**



Enter the service name in Metasploit and using search command and it will tell all available exploit , payload , auxiliary etc. by searching in the module

Ex :

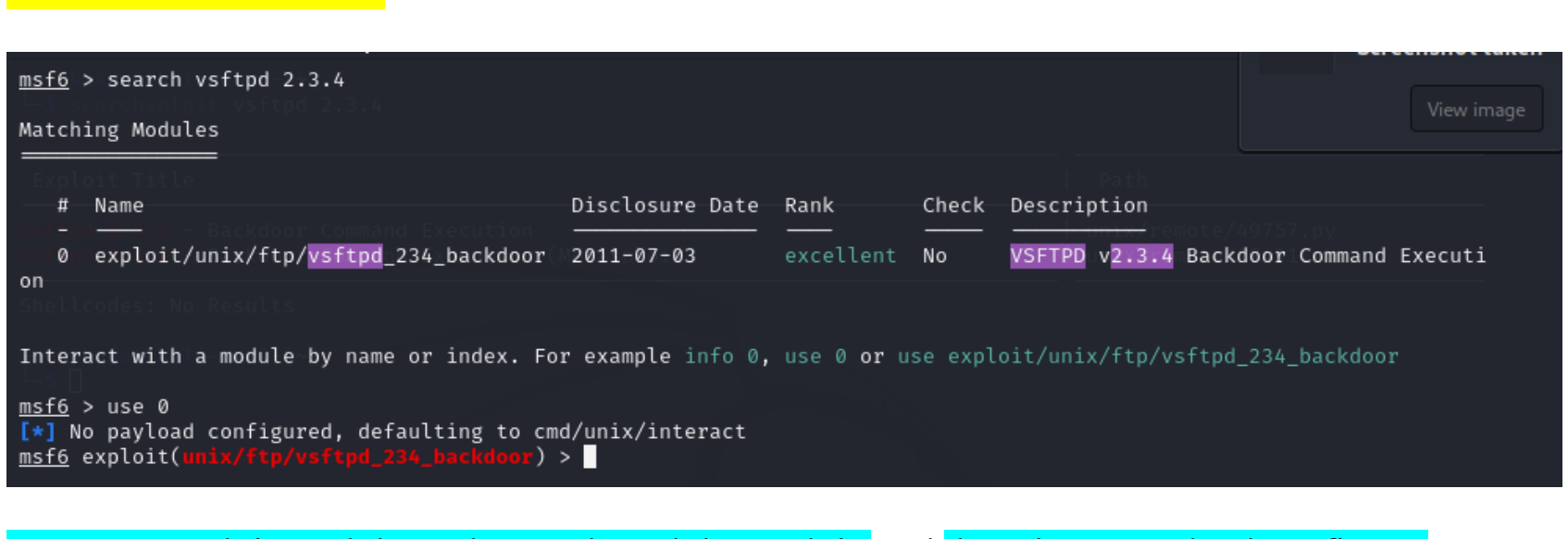


Its telling there is a backdoor exploit for the vsftp service

Ranking is important in our case it is excellent means it is highly exploitable

Selecting the exploit :

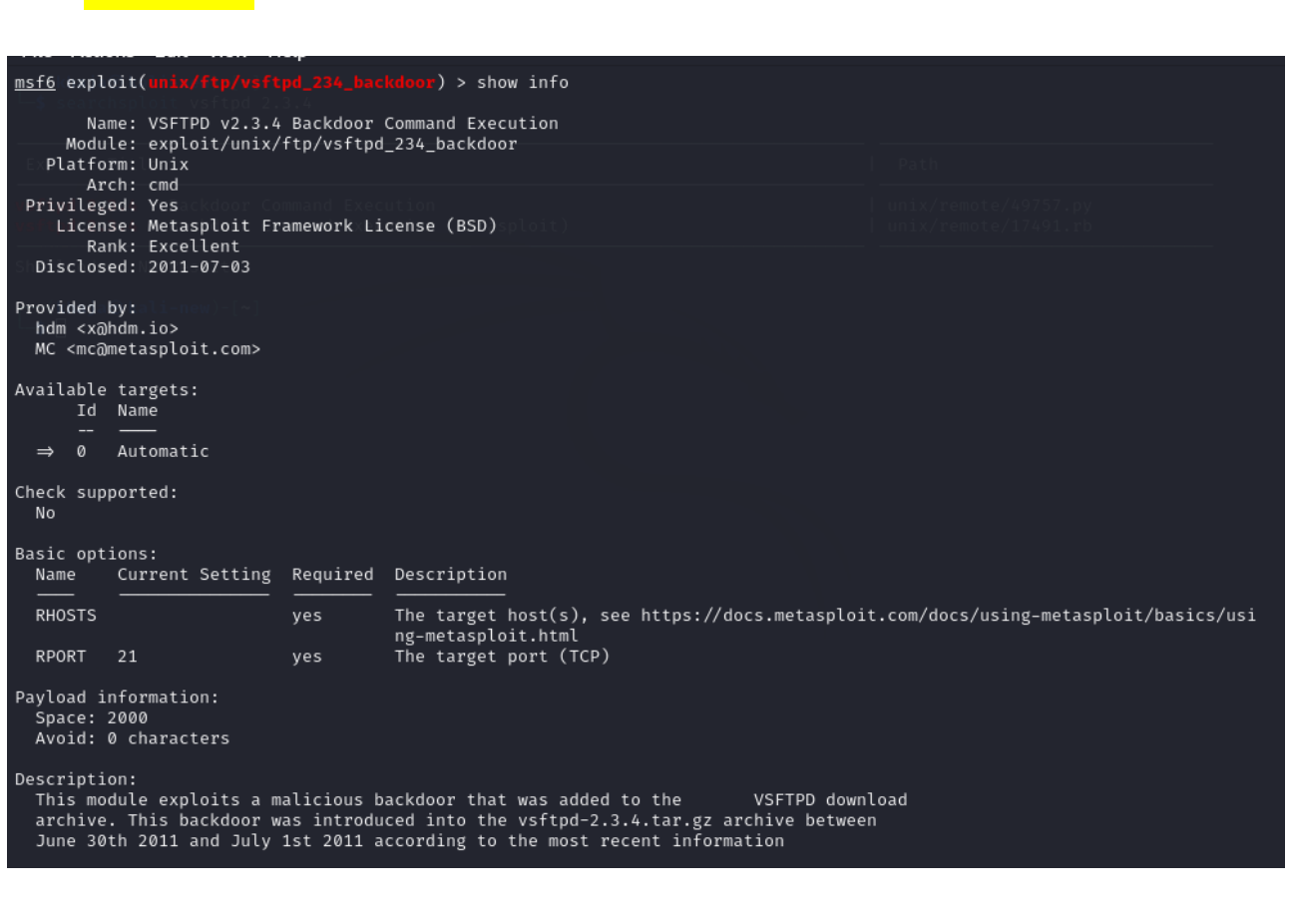
Use <name or number>



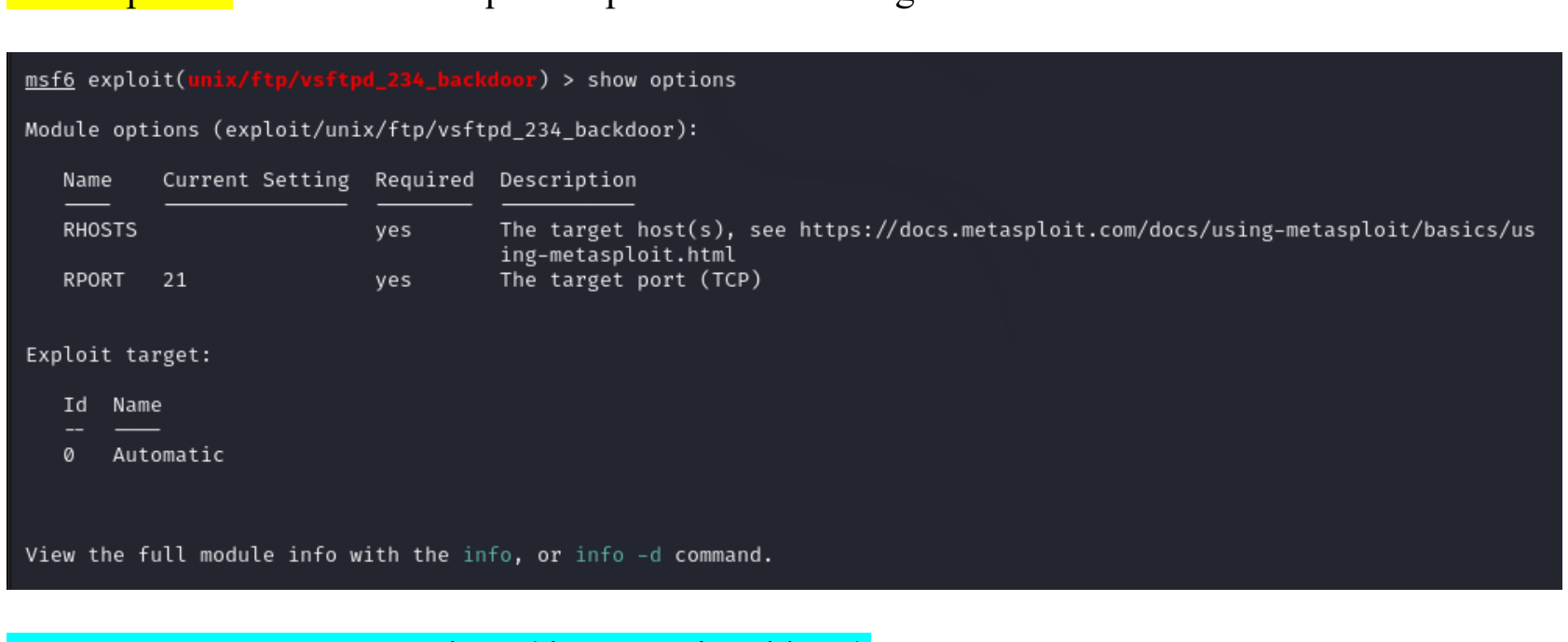
Means In exploit module we have selected the exploit and there is not payload configure

Reading the description about the exploit :

Use show info :



Show option : what are the required options we have to give



RHOST means -- > remote host (the target ip address)

RPORT means -- > remote port (we can change the port if we want)

If Required is yes it means it is mandatory to fill

Note : if is important to read about the exploit before launching it because we are responsible of our actions

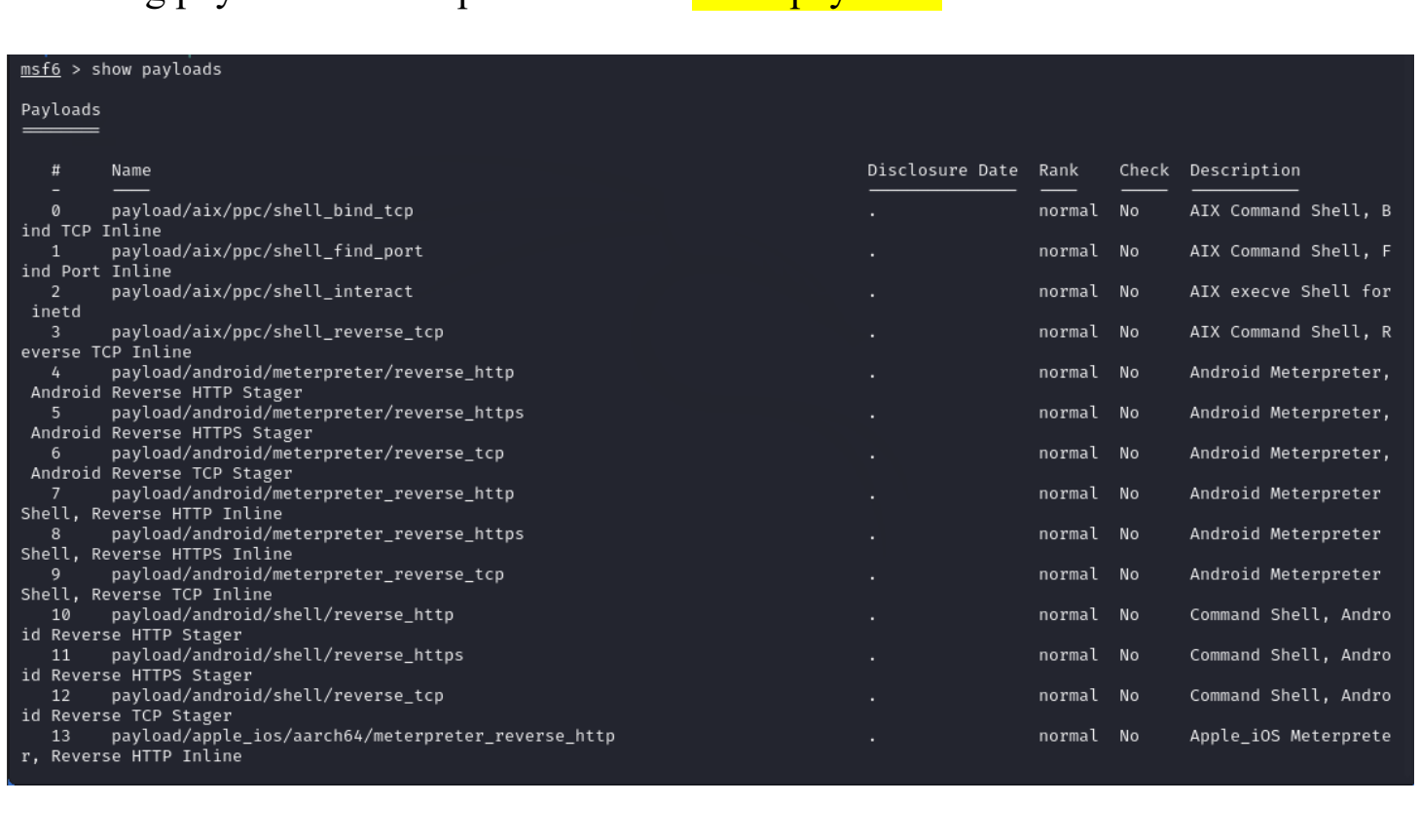
Setting the options :

set RHOST <Target ip >

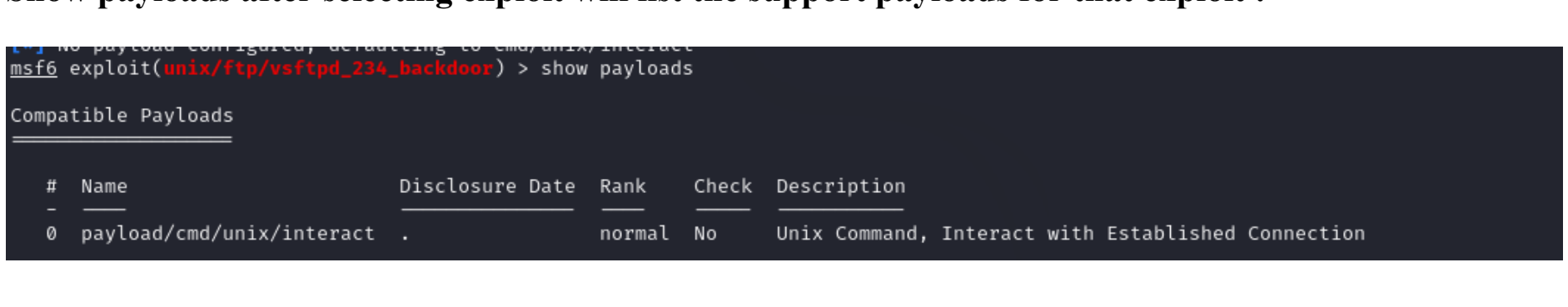
Then show options to verify.

Step 6 : selecting the payload

Checking payload in metasploit use -- > show payloads



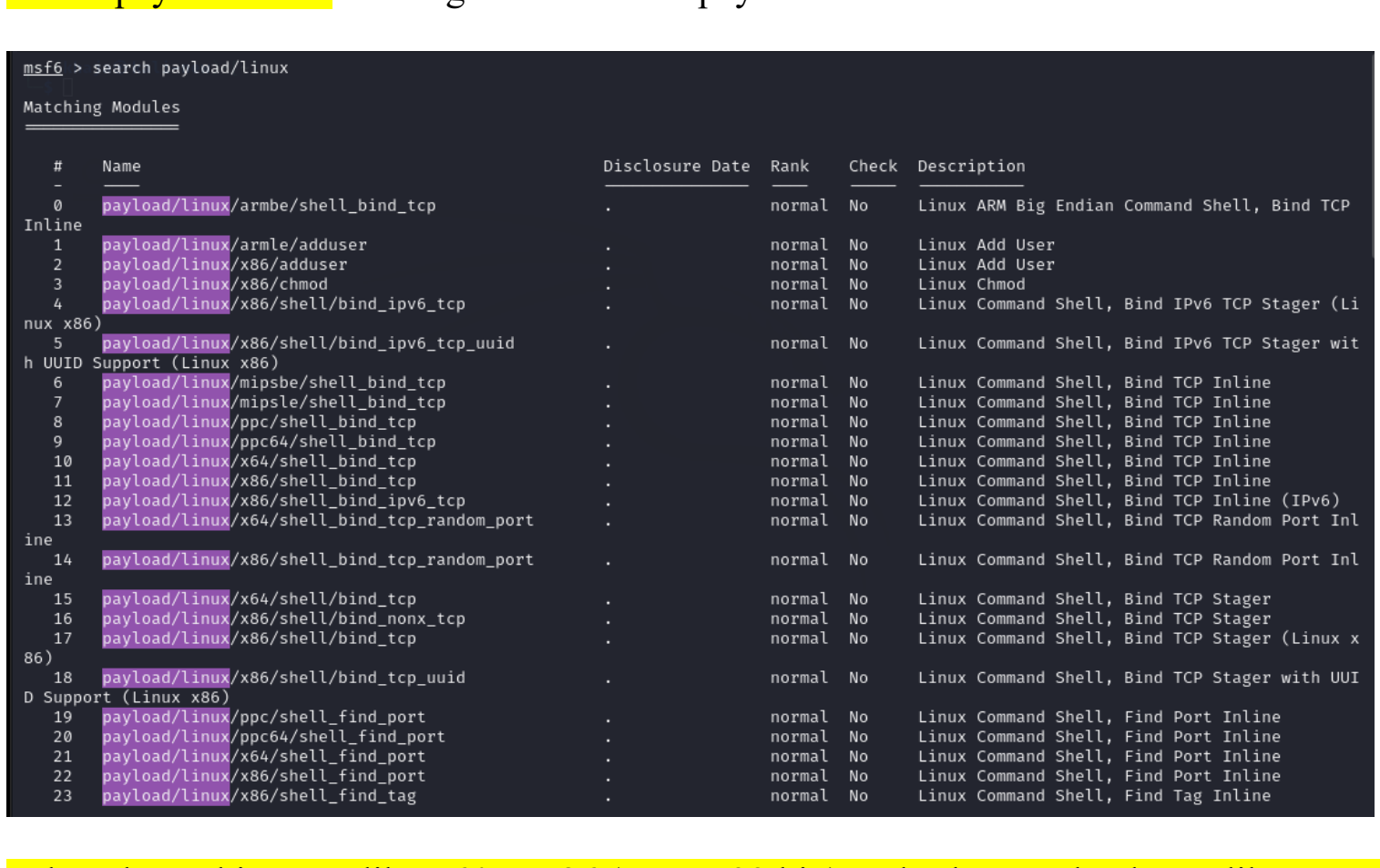
Show payloads after selecting exploit will list the support payloads for that exploit :



Note : we don't need to select if one single payload is there

We want reverse shell payload for linux for that use :

search payload/linux -- > to get all the linux payloads



Select the architecture like x64 or x86 (means 32 bit) and select payload type like reverse shell , bind shell or etc

The following is the reverse shell payload but we have staged and unstaged payload meaning :

- **Staged**:
  exploit → small loader (stage 1) → full payload (stage 2)

  Like :Send small loader → it downloads the full payload
  1 stage will delivery and it will call the shell and rest payload will get delivered
  If the payload name has a **slash** /, it's staged.

  Staged payload are used for bypassing defensive solutions

- **Unstaged**:
  exploit → full payload in one shot

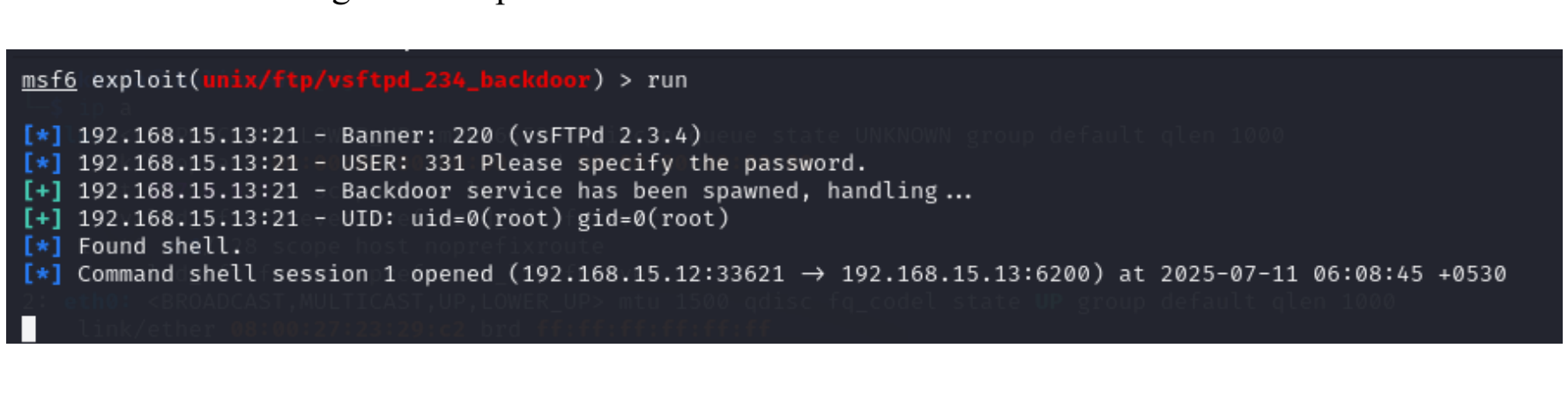
  Like: everything is delivered **together immediately**

  If it has an **underscore** _, it's unstaged.

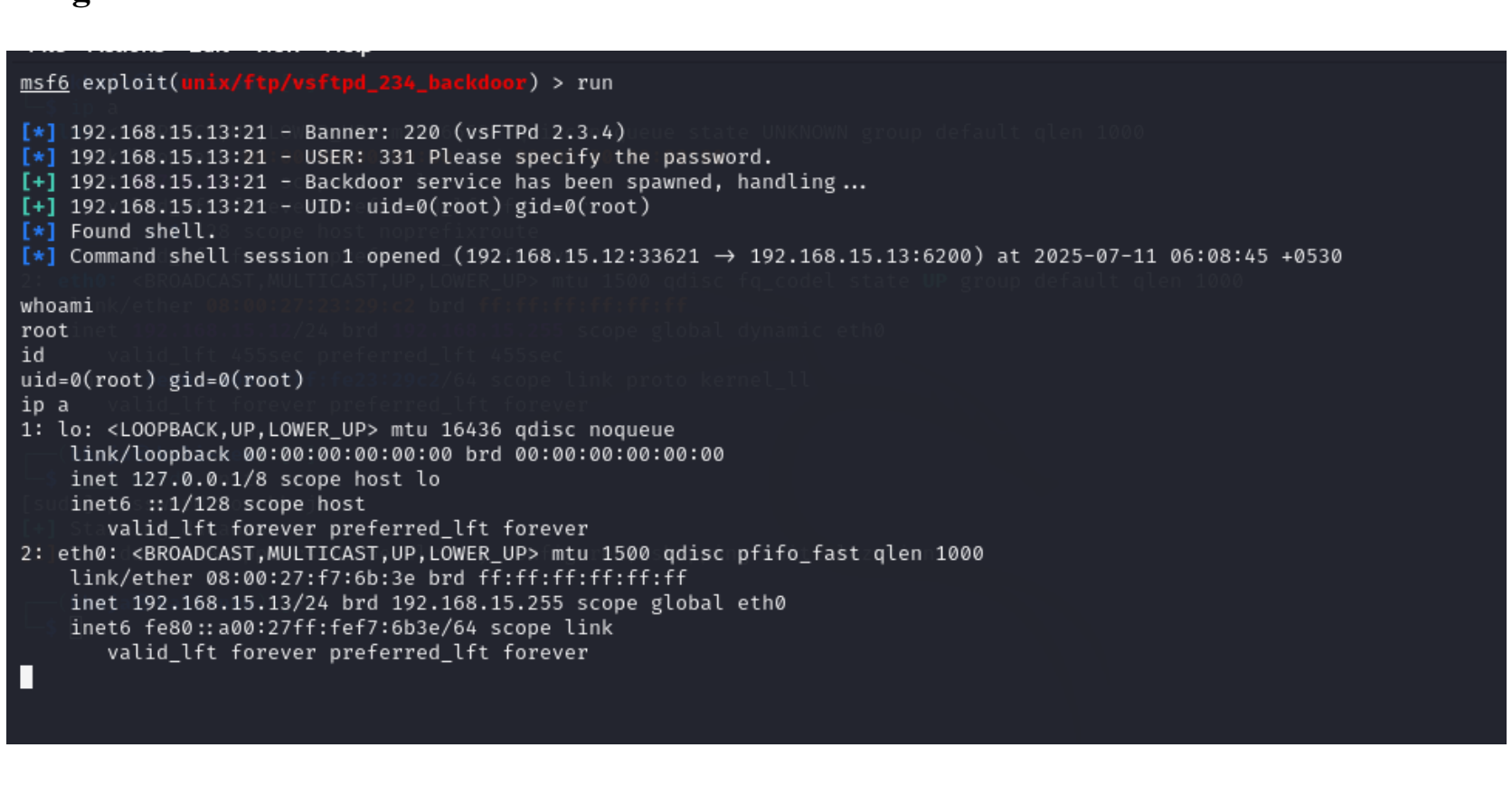
Note : certain exploit will support only selected payloads

Step 7 : launching

We can launch it using run or exploit

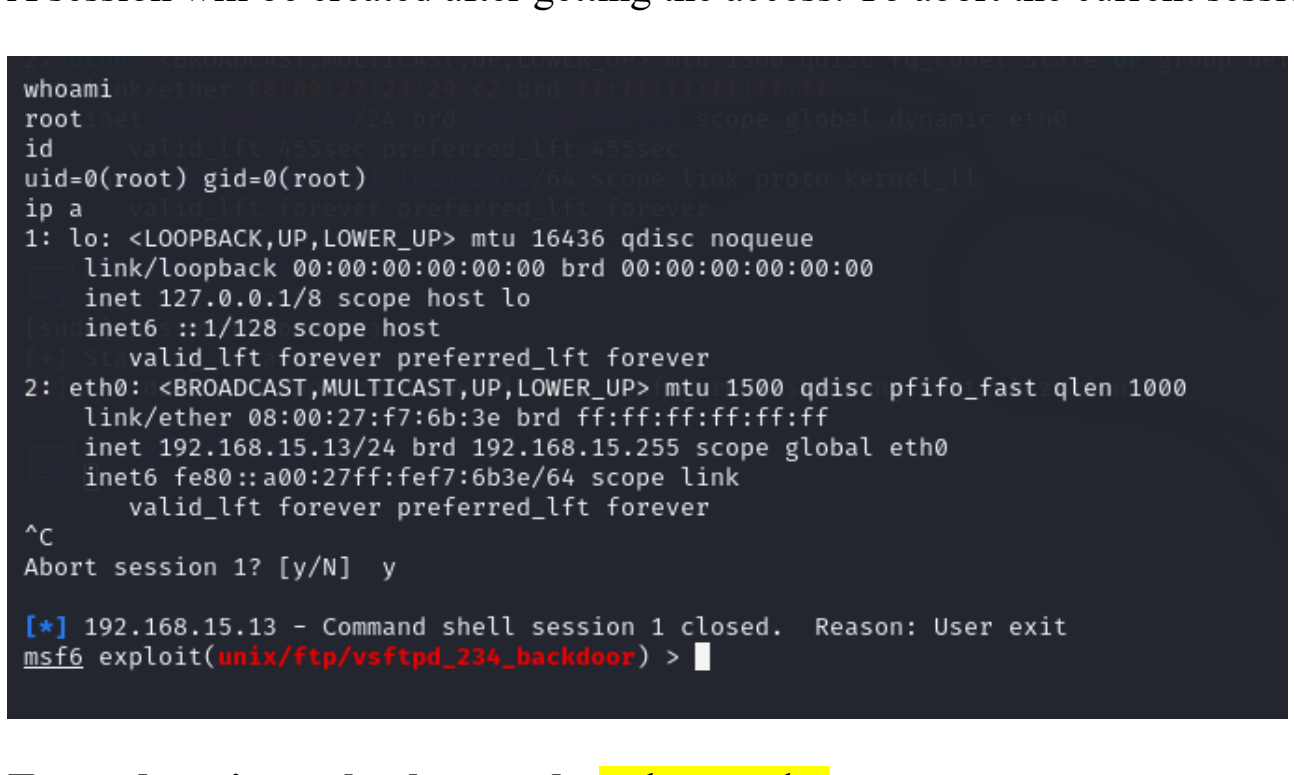


We got the root lvl access :

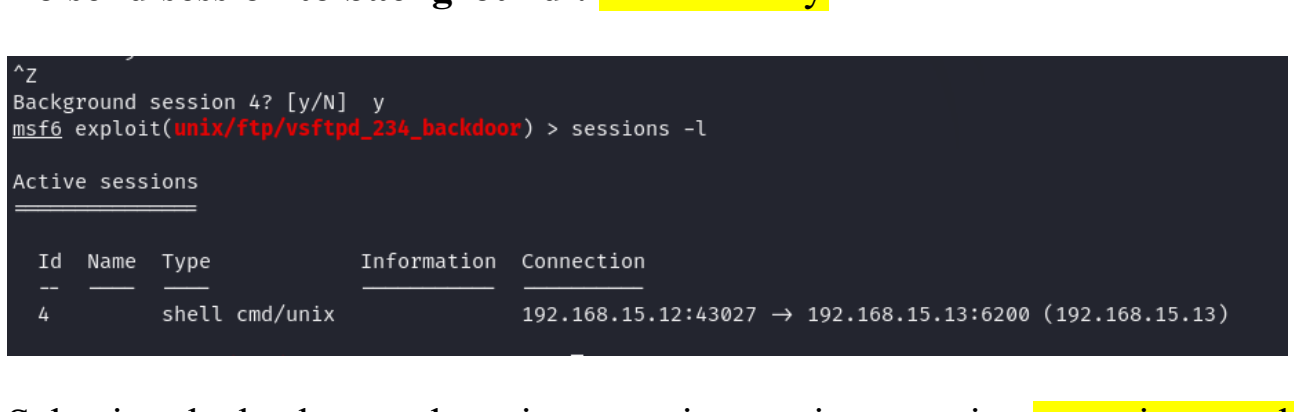


Sessions :

A session will be created after getting the access. To abort the current session ctrl +c and y

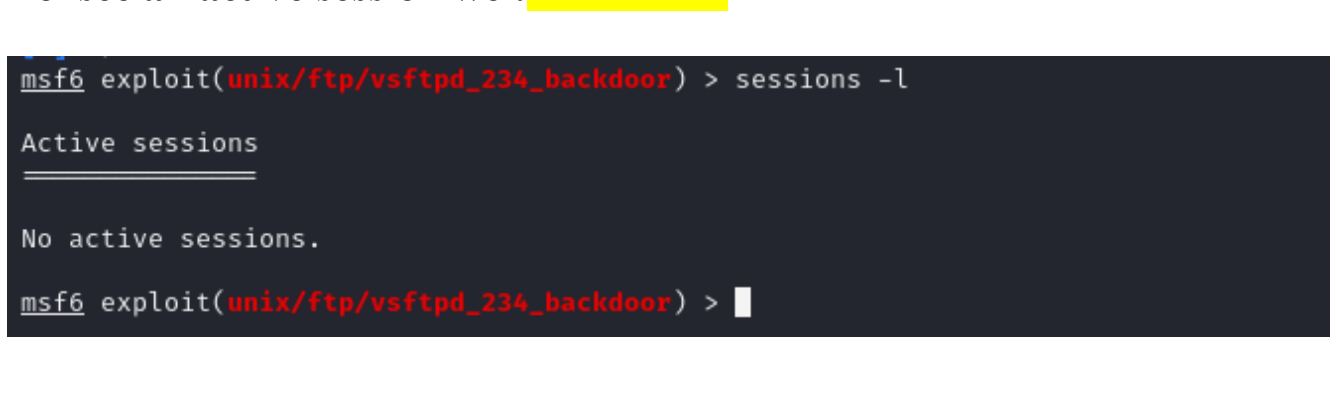


To send session to background : ctrl + z and y



Selecting the background session or active session : session <session number>

To see all active session we : sessions -l



Poc (proof of concept) :

Whether the exploit worked or not

Creating a directory named khaja