**ARYAMAN MISHRA**

**19BCE1027**

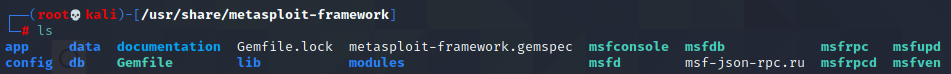
**LAB 5**

**INFORMATION GATHERING USING Metasploit**

Access Framwork folder:



View Contents of Folder:



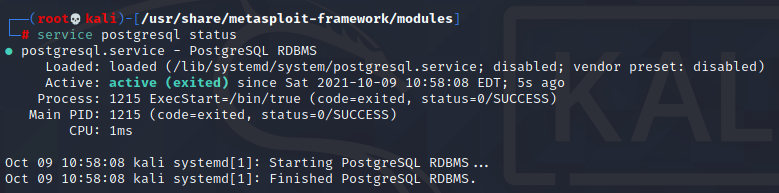
Access Modules folder:

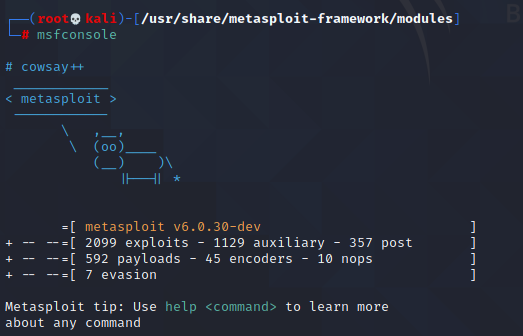


View Contents of Folder:

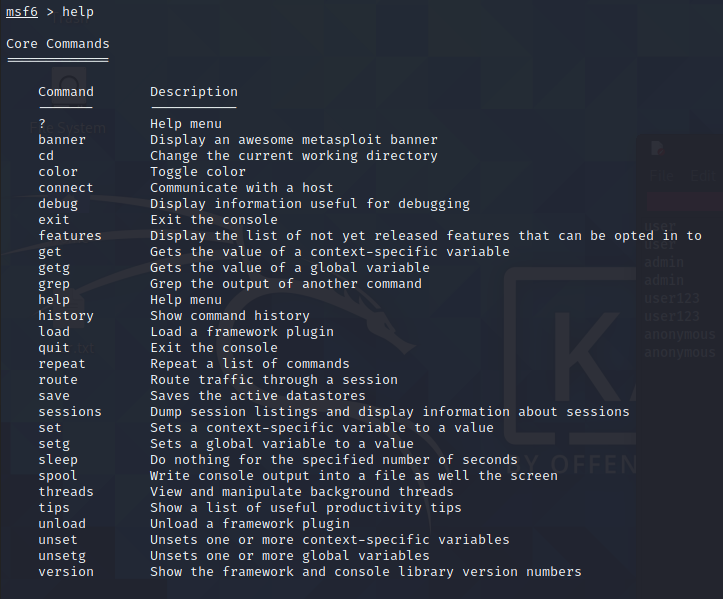


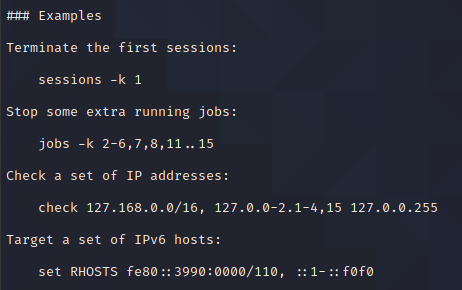
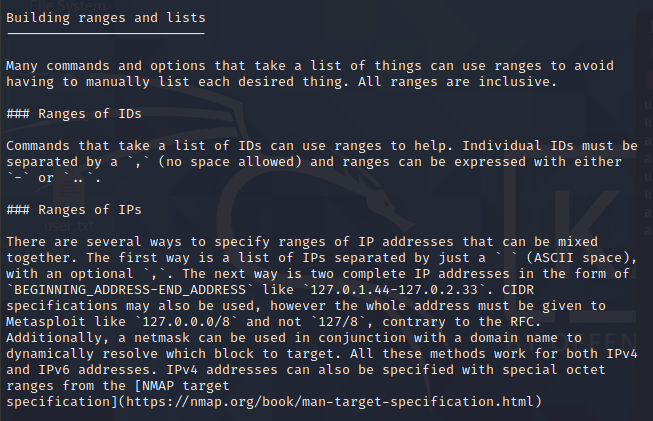
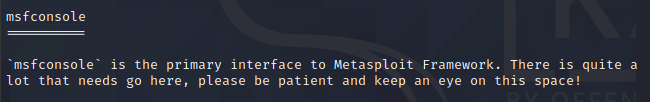
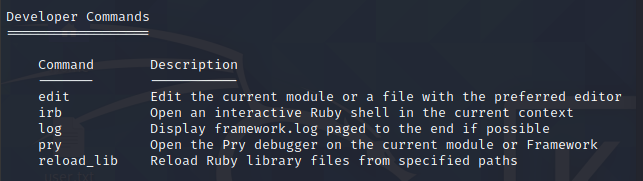
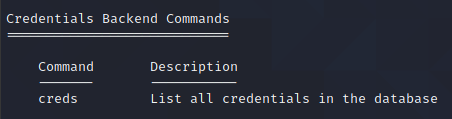
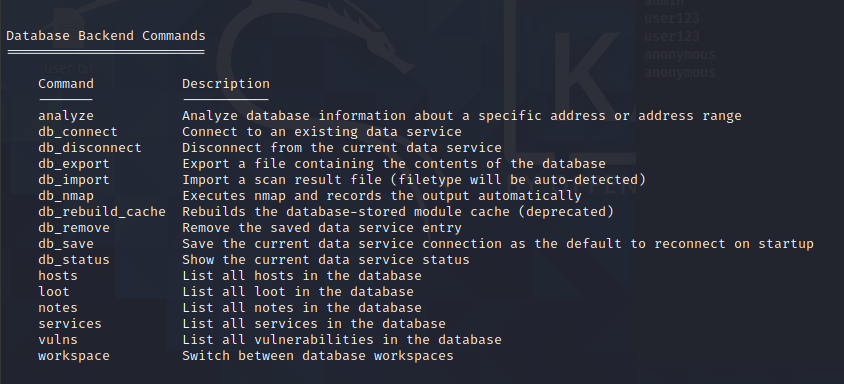
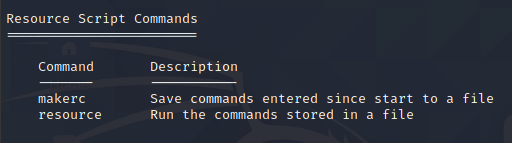
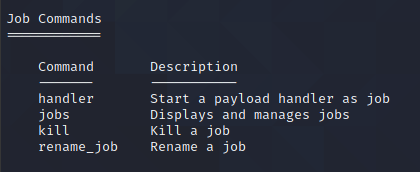
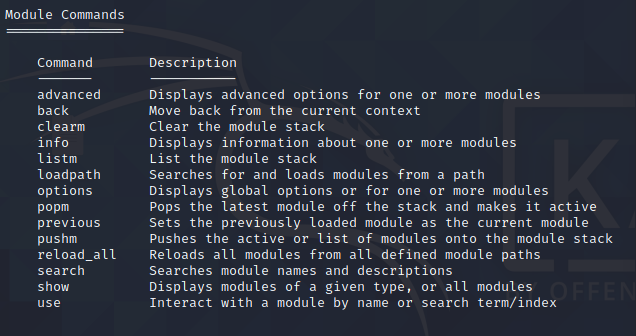
Connect to Database:

Check database status:

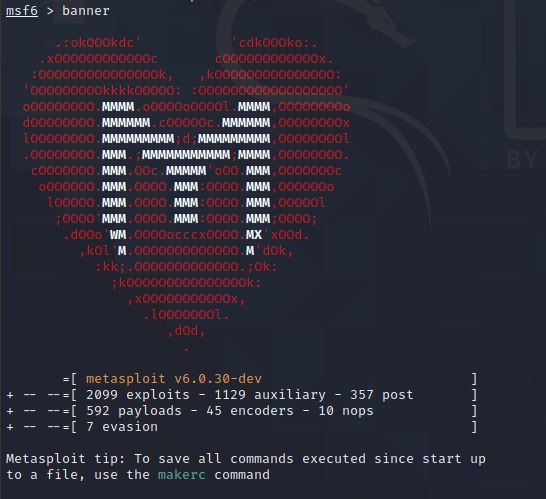
Launch Metasploit:

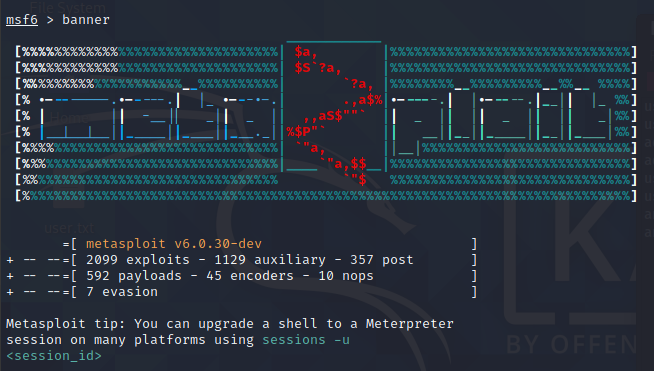
View commands:





You can change banner of Metasploit using banner command:



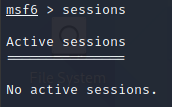


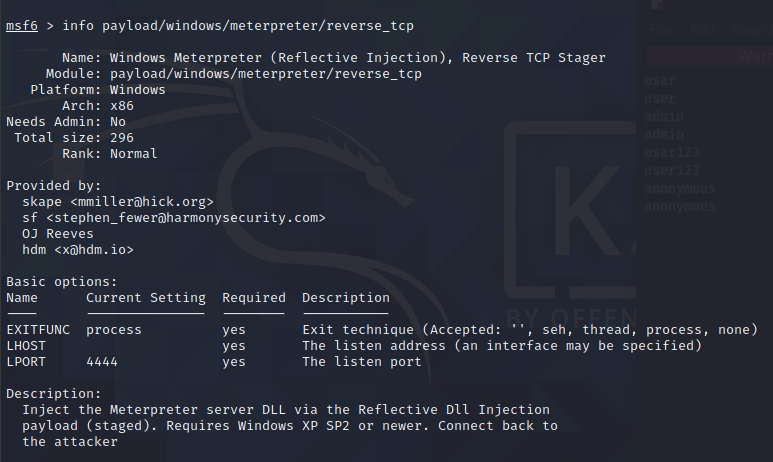
Check version:



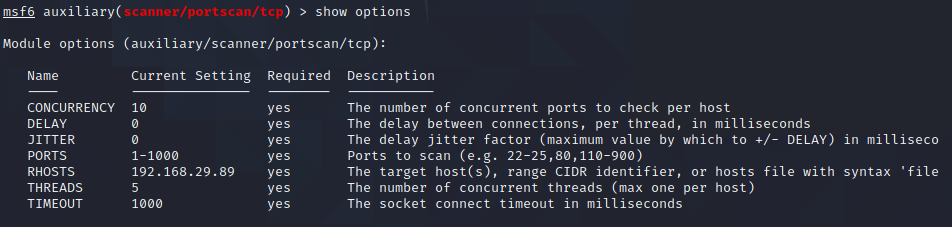
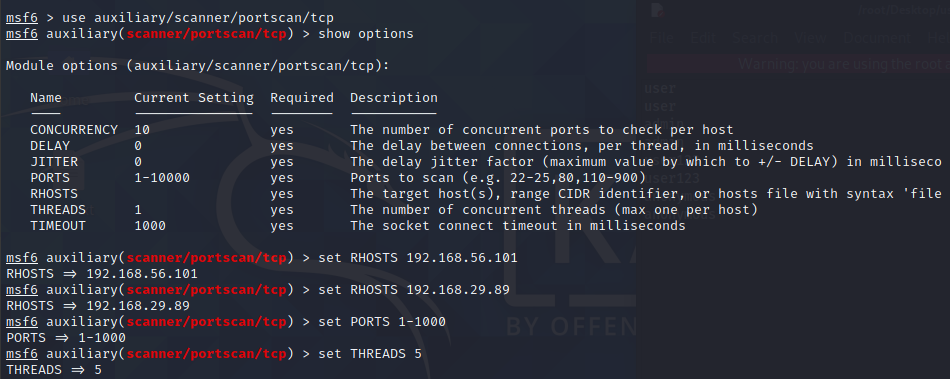
Check status of databse:

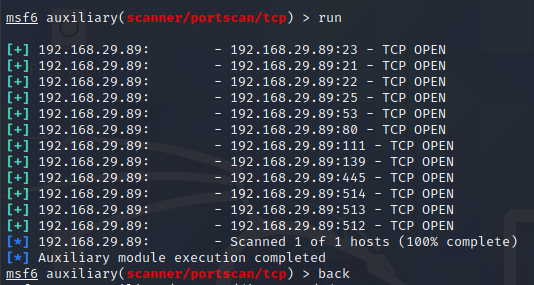
Check for active sessions:



command can be used for generating payloads to be used in many locations:

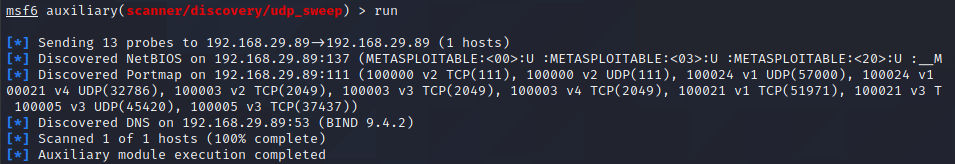
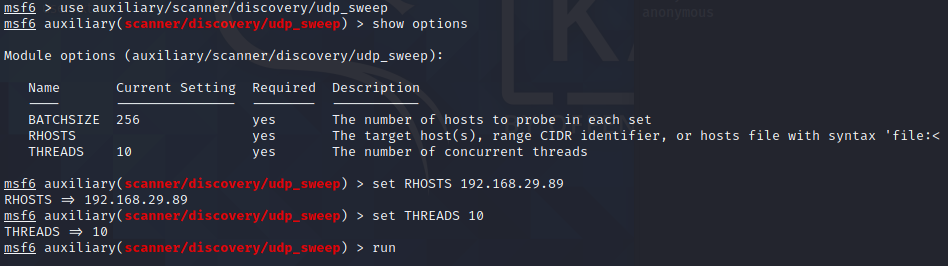
**TCP:**

View options and set Host Address,Ports and Threads:

Run with above configuration:

**UDP:**

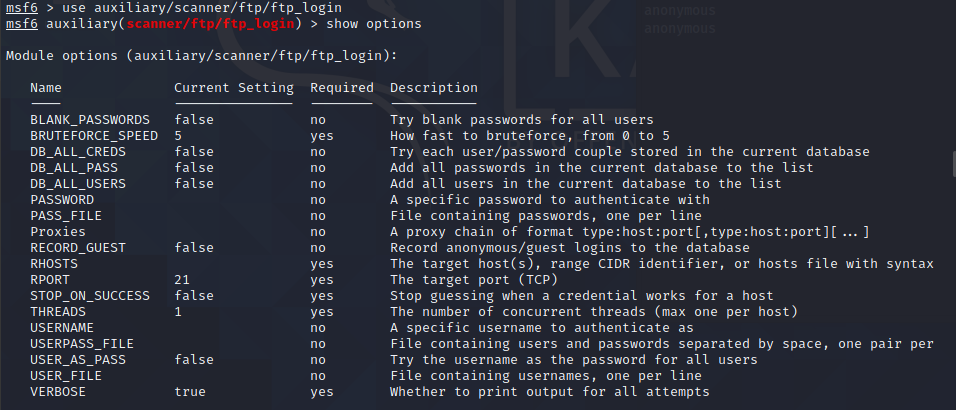
The UDP Service Sweeper auxiliary module allows us to detect interesting UDP services. Since UDP is a connectionless protocol, it is more difficult to probe than TCP. Using an auxiliary module like the UDP Service Sweeper can help you find some low-hanging fruit, in a timely manner.The **udp\_sweep** module scans across a given range of hosts to detect commonly available UDP services. To configure this module, we just need to set the RHOSTS and THREADS values and run it.

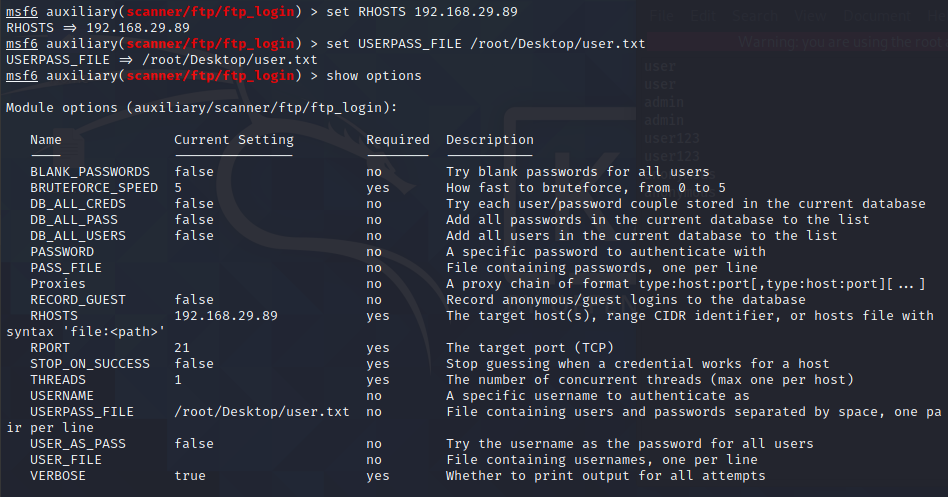


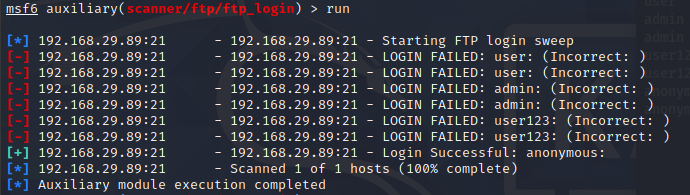


**FTP:**

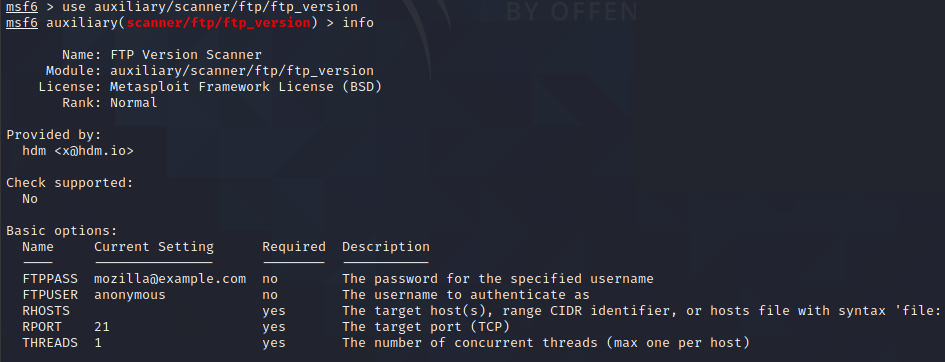
The **ftp\_login** auxiliary module will scan a range of IP addresses attempting to log in to FTP servers.

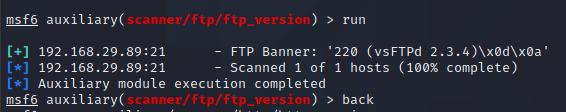
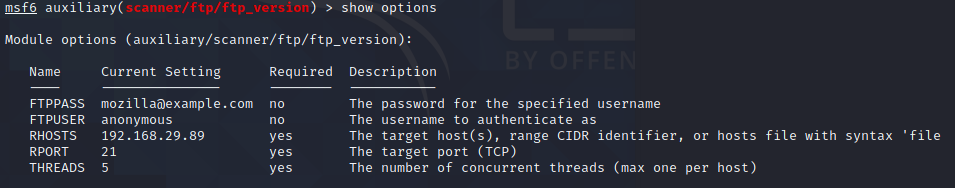
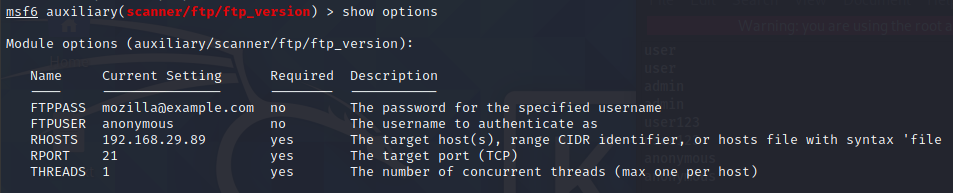




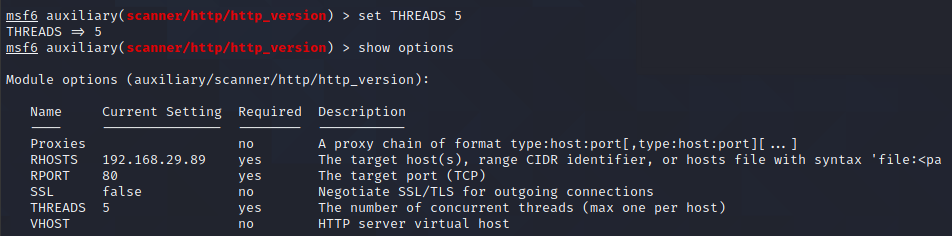
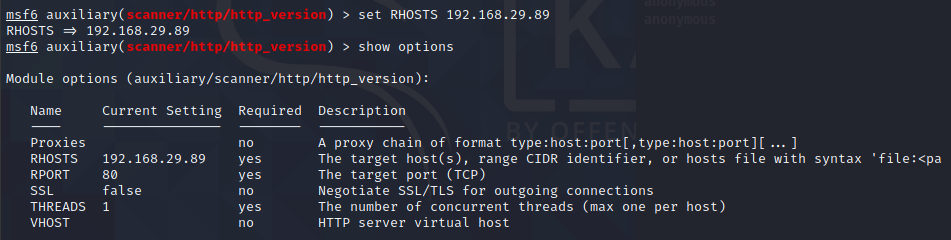
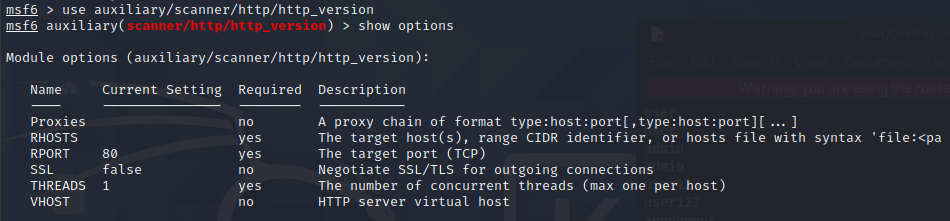
This module can take both wordlists and user-specified credentials in order to attempt to login.

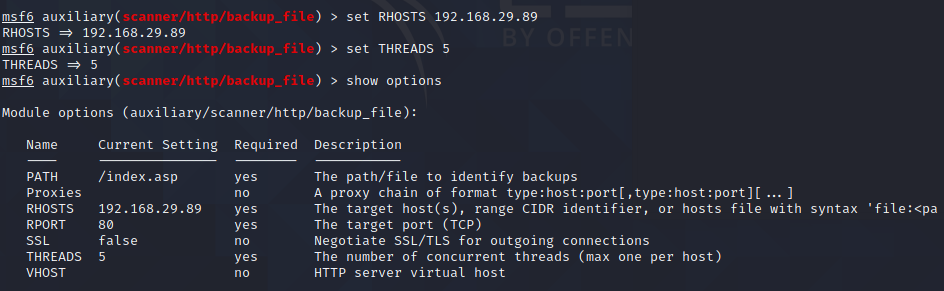
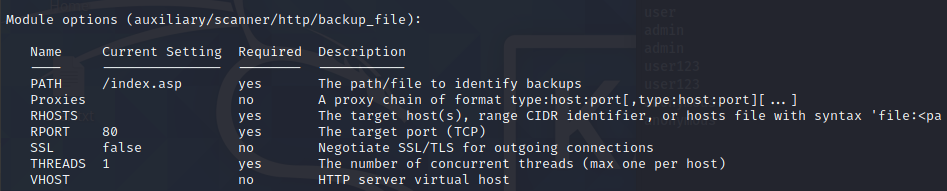
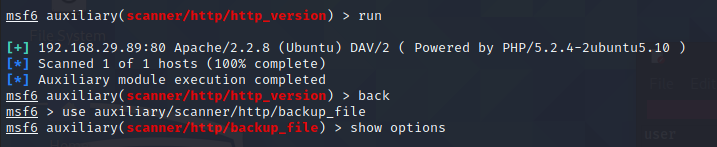
The **ftp\_version** module simply scans a range of IP addresses and determines the version of any FTP servers that are running. To setup the module, we just set our RHOSTS and THREADS values and let it run.





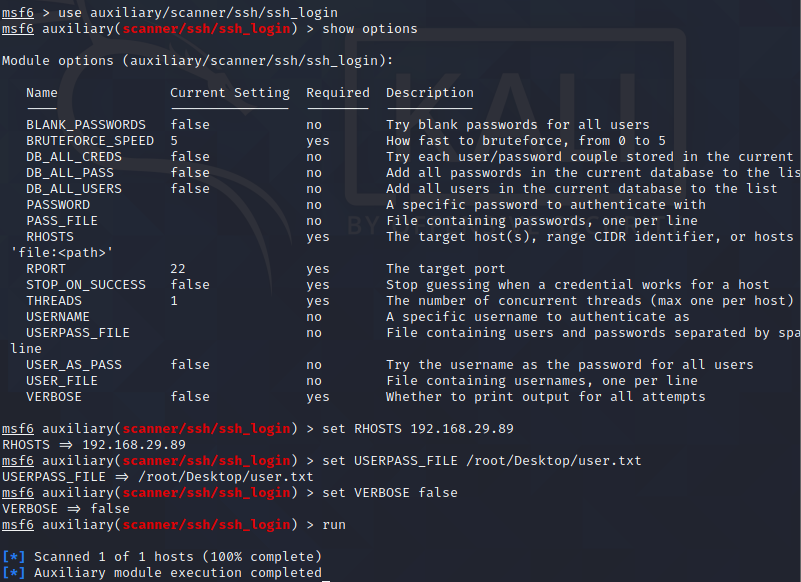
**HTTP:**

The **http\_version** scanner will scan a range of hosts and determine the web server version that is running on them.

To run the scan, we set the RHOSTS and THREADS values and let it run.

**SSH(SECURE SHELL):**

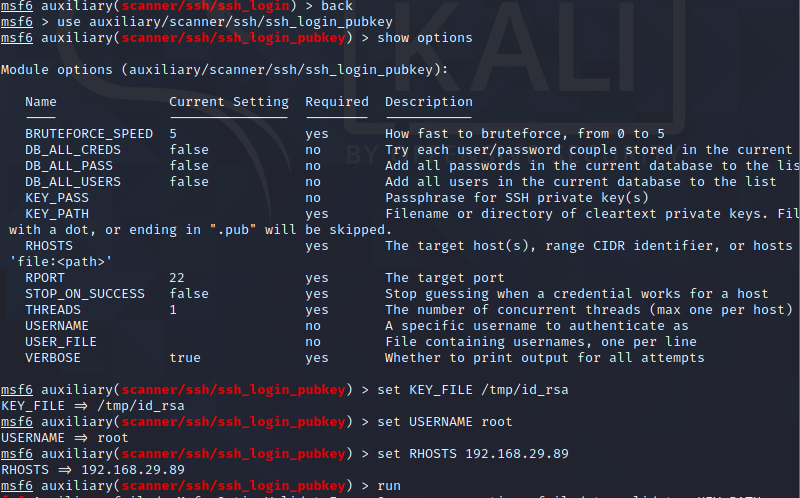
The **ssh\_login** module is quite versatile in that it can not only test a set of credentials across a range of IP addresses, but it can also perform brute force login attempts. We will pass a file to the module containing usernames and passwords separated by a space as shown below. Next, we load up the scanner module in Metasploit and set USERPASS\_FILE to point to our list of credentials to attempt.



With everything ready to go, we run the module.

SSH\_LOGIN\_PUBKEY

Using public key authentication for SSH is highly regarded as being far more secure than using usernames and passwords to authenticate. The caveat to this is that if the private key portion of the key pair is not kept secure, the security of the configuration is thrown right out the window. If, during an engagement, you get access to a private SSH key, you can use the **ssh\_login\_pubkey** module to attempt to login across a range of devices.



[\*] 192.168.89.29:22 - SSH - Testing Cleartext Keys

[\*] 192.168.89.29:22 - SSH - Trying 1 cleartext key per user.

[\*]Command shell session 1 opened (?? -> ??) at 2021-09-10 17:17:56 -0600

[+] 192.168.1.154:22 - SSH - Success: 'root':'57:c3:11:5d:77:c5:63:90:33:2d:c5:c4:99:78:62:7a' 'uid=0(root) gid=0(root) groups=0(root) Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux '

[\*] Scanned 1 of 1 hosts (100% complete)

[\*] Auxiliary module execution completed

msf auxiliary(ssh\_login\_pubkey) > sessions -i 1

[\*] Starting interaction with 1...

ls

reset\_logs.sh

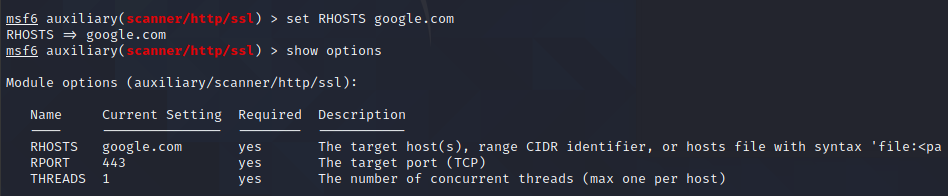
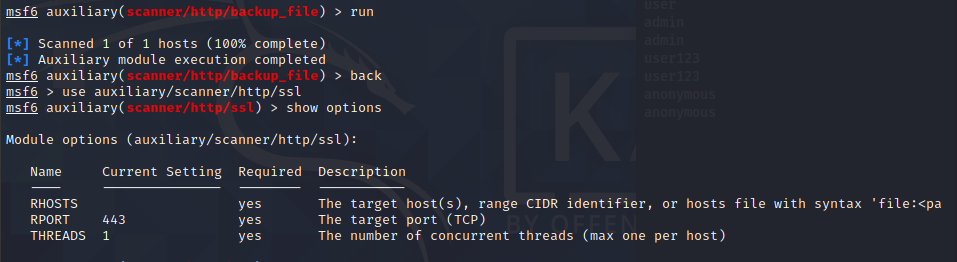
id

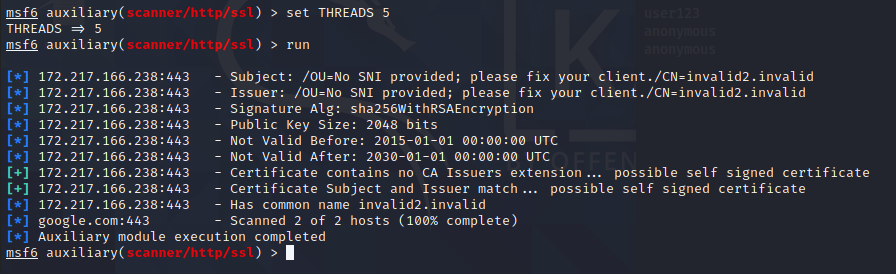
uid=0(root) gid=0(root) groups=0(root)

exit

[\*] Command shell session 1 closed.

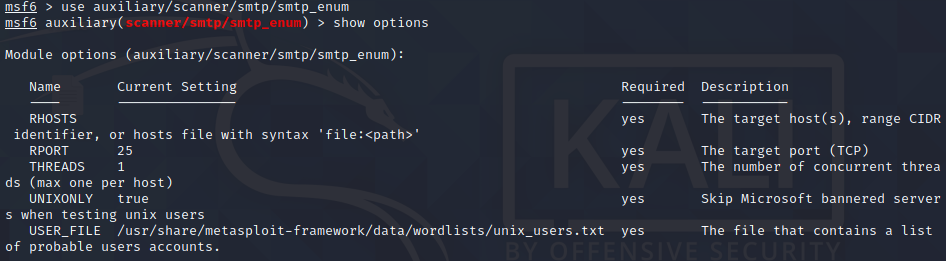
**SSL:**

The **ssl** module queries a host or range of hosts and pull the SSL certificate information if present.

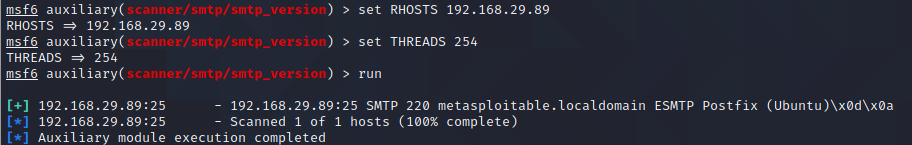
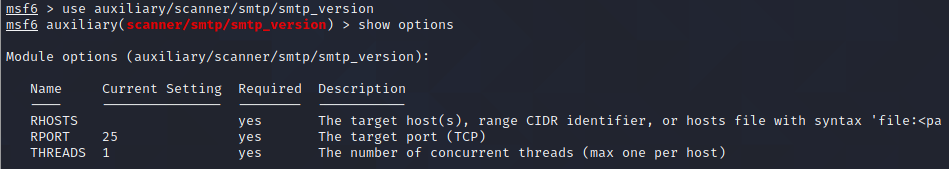
To configure the module, we set our RHOSTS and THREADS values and let it run.

**SMTP:**

The SMTP Enumeration module will connect to a given mail server and use a wordlist to enumerate users that are present on the remote system.

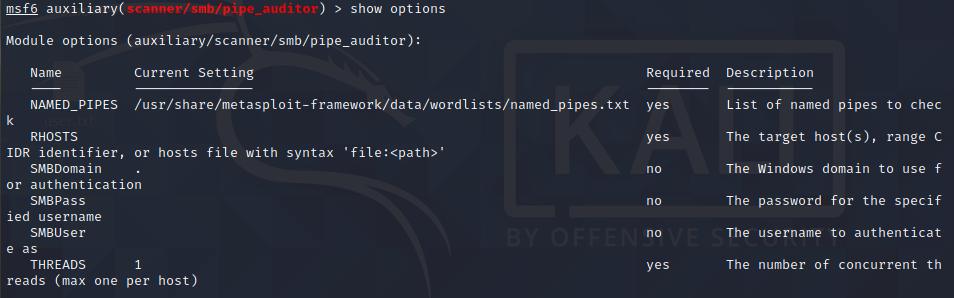


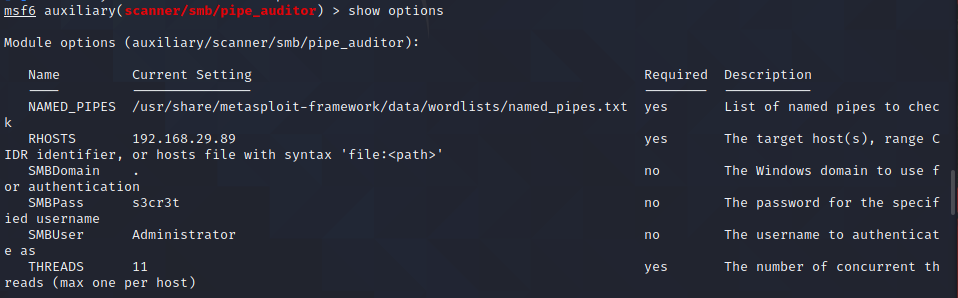
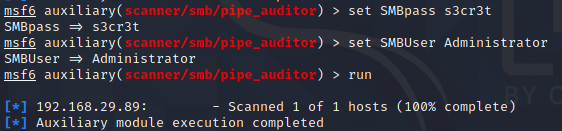
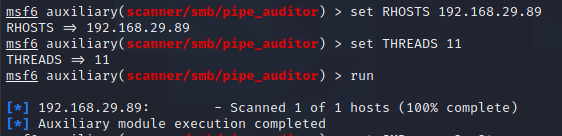
 Since the email username and system username are frequently the same, you can now use any enumerated users for further logon attempts against other network services.

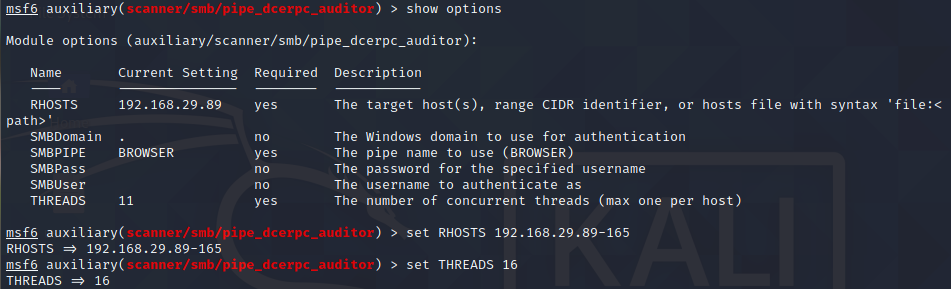
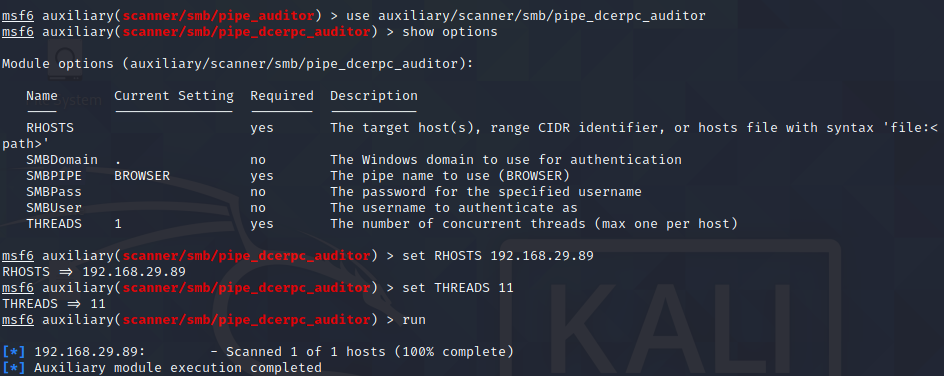
Poorly configured or vulnerable mail servers can often provide an initial foothold into a network but prior to launching an attack, we want to fingerprint the server to make our targeting as precise as possible. The **smtp\_version** module, as its name implies, will scan a range of IP addresses and determine the version of any mail servers it encounters.

**SMB:**

The **pipe\_auditor** scanner will determine what named pipes are available over SMB. In your information gathering stage, this can provide you with some insight as to some of the services that are running on the remote system.





The **pipe\_dcerpc\_auditor** scanner will return the DCERPC services that can be accessed via a SMB pipe.

The **smb\_enumshares** module, as would be expected, enumerates any SMB shares that are available on a remote system.