
Grado en Ingeniería Informática

Seguridad en los Sistemas Informáticos

Departamento de Ingeniería Informática

# Práctica 5: Explotación de sistemas, redes y contraseñas (Parte II)

**Autor:**
Juan Boubeta Puig y
Jesús Lagares Galán

*Fecha:*
7 de noviembre de 2024

# Índice

# Índice de figuras

# 1. Objetivo

En la parte I de esta práctica hemos podido conocer y poner en práctica algunos ataques basándonos en el objetivo —hemos atacado a un sistema, una red y una contraseña—. Por ello, en esta segunda sesión descubriremos técnicas de ataque que podemos utilizar con diferentes objetivos.

Conoceremos los ataques por fuerza bruta y explotación de protocolos, así como los diferentes tipos que existen y herramientas para llevarlos a cabo como Hydra o smbmap.

# 2. Ataques por fuerza bruta

Los ataques de fuerza bruta son uno de los más sencillos de realizar por cualquier iniciado en la seguridad informática. Consisten en probar todas las opciones posibles para encontrar la contraseña o credencial correcto. En la actualidad son muchas las páginas web y aplicaciones que implementan métodos de protección contra ataques de fuerza bruta, por ejemplo el **captcha** (véase la Figura 1).

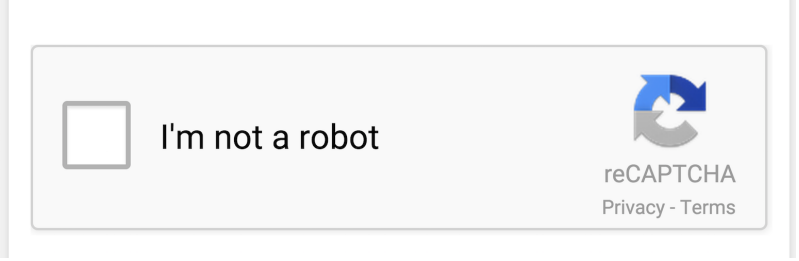


Figura 1: Tecnología captcha.

Existen diversos ataques por fuerza bruta, pero los más conocidos son los que se describen a continuación.

## 2.1. *Simple brute force attacks*

El atacante intenta adivinar los credenciales del usuario sin la ayuda de herramientas especializadas u otros medios. Esta técnica fue el origen de los ataques por fuerza bruta, y es funcional cuando no existe protección contra ellos y los credenciales son sencillos o fáciles de adivinar. Por ejemplo, si nuestra contraseña de inicio de sesión es `admin` o `contraseña123`, será muy probable que un *simple brute force attack* pueda con ella.

## 2.2. *Dictionary attacks*

El atacante escoge, o crea, un diccionario y ejecuta todas las posibles contraseñas que aparecen en él gracias a una herramienta especializada que prueba las combinaciones por él.

Un diccionario no es más que un archivo de texto (puede ser un simple `.txt`) en el que aparecen almacenadas diferentes contraseñas a probar para iniciar sesión en la cuenta del objetivo. Estos diccionarios pueden crearse gracias a *scripting* simple en cualquier lenguaje como Python, o herramientas especializadas como **John the Ripper** [6]. A su vez, también pueden descargarse diccionarios ya elaborados que recogen las contraseñas más comunes en los usuarios. Uno de los más conocidos es **rockyou.txt** [2].

## 2.3. *Reverse brute force attacks*

Tal y como su nombre indica, este ataque consiste en hallar el usuario en vez de la contraseña. Para su realización, es necesario conocer una contraseña válida para iniciar sesión en el sistema. Una vez que sepamos que la contraseña es correcta, nos dedicaremos a buscar, y probar, millones de nombres de usuario hasta encontrar los credenciales adecuados. Muchas de estas contraseñas se encuentran filtradas en bases de datos *online* a partir de ataques a empresas o fugas de seguridad. Podemos encontrar un ejemplo de este tipo de filtraciones en el extracto de tabla que Nik Cubrilovic muestra en su artículo *RockYou Hack: From Bad To Worse* [4].

Si queremos saber si nuestra contraseña se ha filtrado en algún momento, ya sea por descuido de la empresa o presa de algún ataque, podemos utilizar aplicaciones como **haveibeenpwned** [10]. Para utilizar esta aplicación web tan solo debemos:

1. Acceder a la página oficial [10].
2. Introducir nuestro correo electrónico en el área de texto y pulsar el botón `pwned?` (véase Figura 2).
3. Esperar a que la herramienta finalice el proceso y observar si nuestra contraseña ha sido filtrada o no (véase Figura 3).

Para la realización de estos ataques, aunque podemos llevarlos a cabo de forma manual con mucha paciencia, existen herramientas especializadas como **CrackMapExec** o **Hydra**.

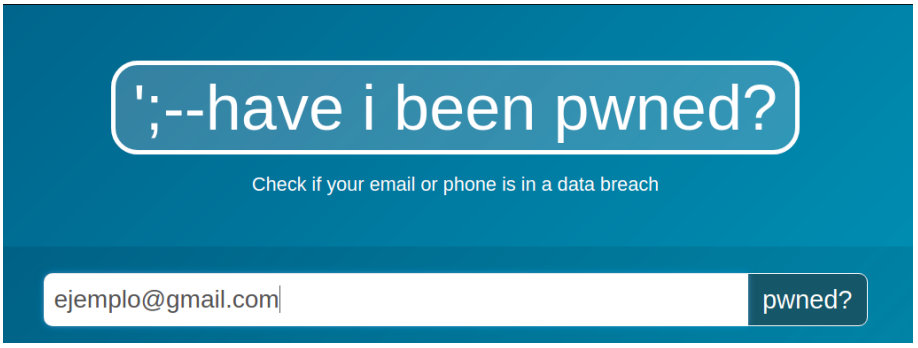


Figura 2: Introducimos el correo en la página web.

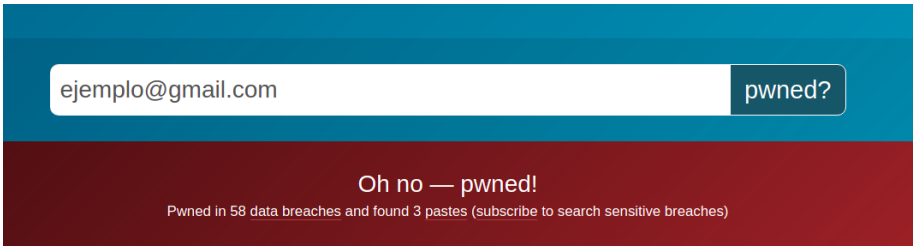


Figura 3: La herramienta nos notifica que nuestra contraseña ha sido filtrada.

## 2.4. CrackMapExec

**CrackMapExec (CME)** [3] es una herramienta que nos ayuda a automatizar la evaluación de la seguridad dentro de redes *Active Directory* (AD). Esta herramienta se aprovecha de diversas características y protocolos presentes en las redes *Active Directory* para lograr su funcionalidad, permitiendo evadir mucha de la seguridad inicial presente en estas redes.

Una red AD no es más que un conjunto de servicios y una base de datos que conectan a los usuarios con los recursos de red que necesitan para realizar su trabajo.

CrackMapExec es una herramienta con la que siempre vamos a atacar a un protocolo específico (véase Sección 3). Para atacar estos protocolos se nos presentan numerosas opciones, entre ellas la de realizar un ataque mediante fuerza bruta.

### 2.4.1. Instalación

Para instalar CrackMapExec en nuestra máquina encontramos diversas opciones dentro del repositorio oficial de la herramienta [3]. En este caso, necesitaremos tener en nuestro sistema Python 3:

1. Comprobamos si Python está instalado en nuestra máquina con el comando:

```
python3 --version
```

2. En caso negativo, lo instalamos con los siguientes comandos:

```
sudo apt-get update

sudo apt-get install python3.6
```

3. Ahora que tenemos Python instalamos **pipx** para la instalación de la herramienta:

```
sudo apt install pipx
```

4. Instalamos la herramienta con los siguientes comandos:

```
pipx ensurepath

pipx install crackmapexec
```

5. Cuando la instalación se complete, podremos ingresar el comando `crackmapexec` para comprobar que se ha completado correctamente (véase la Figura 4).

### 2.4.2. Ejemplo de uso

Para realizar un ataque de fuerza bruta con esta herramienta, tan solo debemos hacer uso de uno de los siguientes comandos:

```
crackmapexec <protocol><target(s)>-u username1 username2 -p password1
```

Donde *username1, password1*, etc. son las opciones que vamos a probar. O si queremos utilizar diccionarios podemos hacerlo mediante el siguiente comando:

```
crackmapexec <protocol><target(s)>
-u ~/file_containing_usernames
-p ~/file_containing_passwords
```

Para probar esta herramienta vamos a lanzar un sencillo ataque utilizando diccionarios contra el servicio de SSH de `Metasploitable 2`:

1. Abrimos la aplicación con el comando `crackmapexec` (véase la Figura 4).
2. Levantamos la máquina `Metasploitable 2` (visto en la sesión anterior de esta práctica).

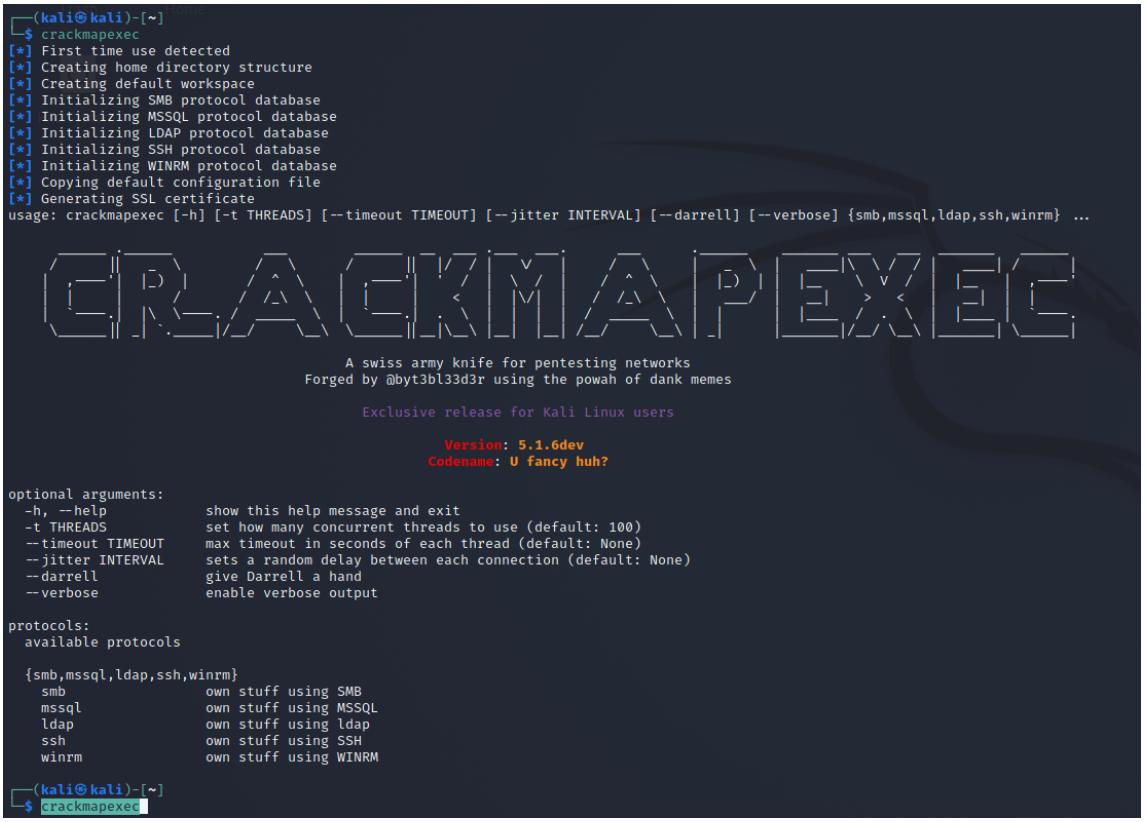


Figura 4: Instalación de crackmapexec correcta

3. Lanzamos un breve escaneo a la IP de `Metasploitable 2` utilizando la herramienta **nmap** como hemos visto en la práctica anterior para comprobar el puerto asociado al servicio SSH (véase la Figura 5).

4. Probaremos a lanzar un *brute force* utilizando los dos diccionarios `.txt` entregados junto al enunciado de la práctica (`user.txt` y `password.txt`). Para encontrar los credenciales de acceso al servicio Telnet usaremos el comando:

   ```
   crackmapexec ssh 192.168.1.4 -u user.txt -p password.txt
   ```

5. Dejamos que la herramienta trabaje y comprobamos cómo en la última coincidencia ha encontrado el par de credenciales capaces de hacer *login* en el servicio SSH (véase la Figura 6).

6. Si intentamos iniciar sesión con estos credenciales vemos como, efectivamente, son correctos (véase la Figura 7).

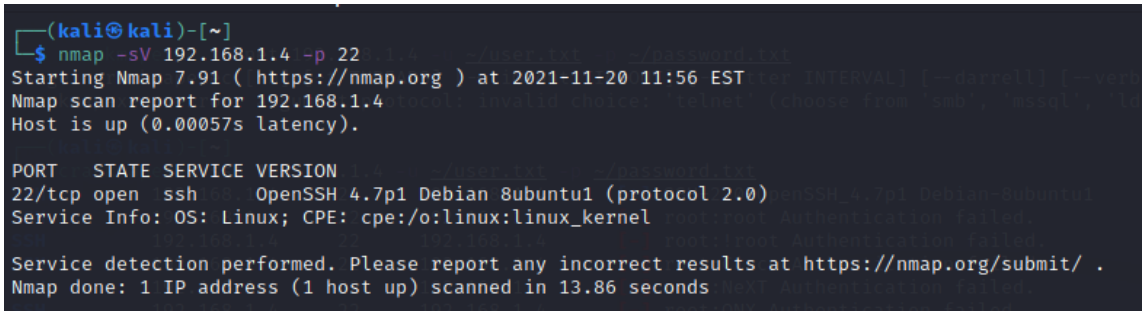


Figura 5: Verificamos el estado del puerto 22

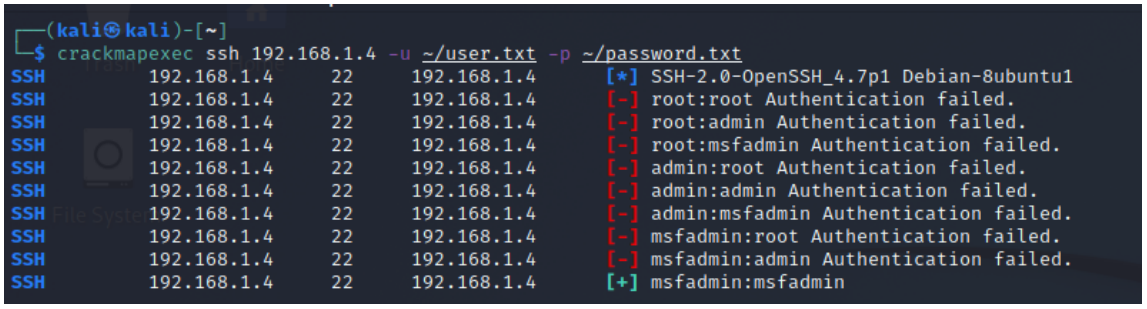


Figura 6: Se encuentran los credenciales correctos


Figura 7: Probamos los credenciales para comprobar que son correctos

## 2.5. Hydra

Hydra es una herramienta especializada en realizar ataques de fuerza bruta contra servicios de *login* o activos cliente/servidor. Gracias a esta herramienta podremos acceder de forma no autorizada a un sistema de forma remota.

### 2.5.1. Instalación

Por defecto, esta aplicación viene preinstalada en nuestra máquina Kali. En caso de no ser así deberemos:

1. Actualizar los paquetes con:

   ```
   sudo apt update
   ```

2. Instalar la aplicación:

   ```
   sudo apt-get install hydra-gtk
   ```

3. Comprobamos que se ha instalado correctamente:

   ```
   hydra -h
   ```

### 2.5.2. Ejemplo de uso

La utilización de Hydra para un ataque por fuerza bruta sigue una sintaxis muy sencilla, mediante dos simples comandos podemos lanzar ataques con usuarios o contraseñas que creamos que serán válidas o con diccionarios enteros.

Algunos ejemplos de uso son:

```
hydra <Target_IP> ssh -l <username> -p <password> -s 22 -vV
```

```
hydra -L <username_file> -P <password_file> ftp://<Target_IP>
```

Para probar esta herramienta vamos a lanzar un sencillo ataque contra `Metasploitable 2`:

1. Abrimos la aplicación con el comando `hydra` o gracias al contenedor de aplicaciones (véase la Figura 8).
2. Levantamos la máquina `Metasploitable 2` (visto en la sesión anterior de esta práctica).

3. Lanzamos un breve escaneo a la IP de `Metasploitable 2` utilizando la herramienta **nmap** como hemos visto en la práctica anterior para comprobar el puerto asociado al protocolo FTP (véase la Figura 9).

4. Probaremos a lanzar un *brute force* con solo 1 usuario y 1 contraseña (que serán *user* y *user*) contra el servicio FTP:

   ```
   hydra 192.168.1.4 ftp -l user -p user -s 21 -vV
   ```

5. En la Figura 10 podemos comprobar cómo Hydra ha automatizado el proceso de prueba y ha detectado que el usuario y contraseña probados eran correctos.

   `ftp 192.168.1.4`.

   Siendo la IP para la conexión la de la máquina `Metasploitable 2`.

   Y como observamos en la Figura 11, el *login* ha sido exitoso.

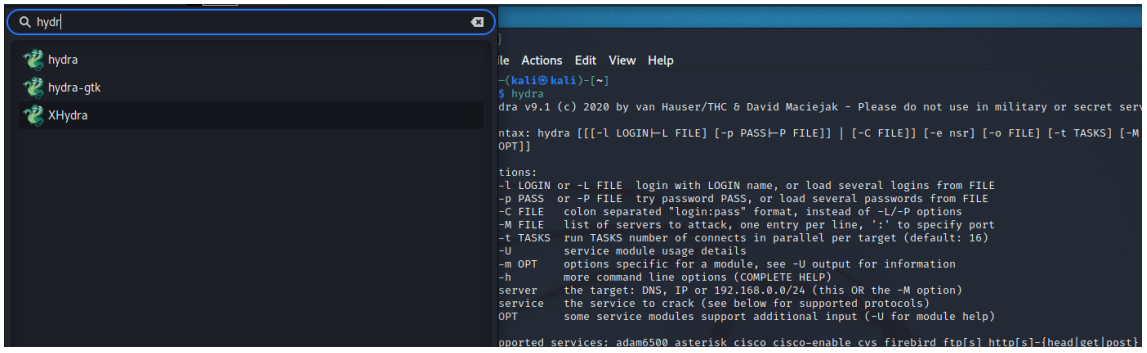


Figura 8: Abrimos la aplicación Hydra.

# 3. Explotación de protocolos

Dentro de los sistemas informáticos existen protocolos (sistemas de normas que regulan la comunicación de una determinada manera) que hacen de la transmisión de la información algo posible en sus diferentes formas. Un ejemplo de protocolo es el tan conocido HTTP *(HyperText Transfer Protocol)*, un protocolo que sirve para establecer los saltos entre una página web y otra.

Estos protocolos cambian y evolucionan con el tiempo junto a la tecnología, pero no son sustituidos en la mayoría de los casos. Por este motivo, si somos capaces de reconocer uno de estos protocolos y saber sus vulnerabilidades durante nuestra fase

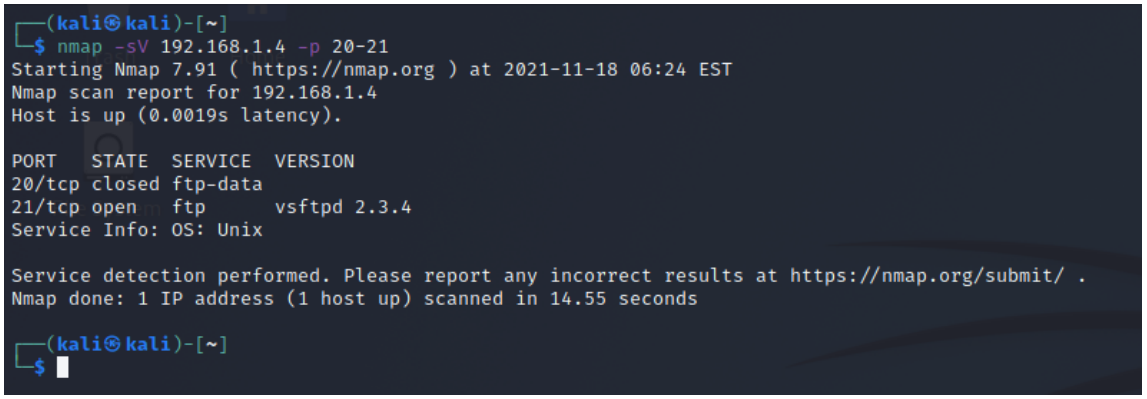


Figura 9: Compramos el estado de los puertos 20 y 21.

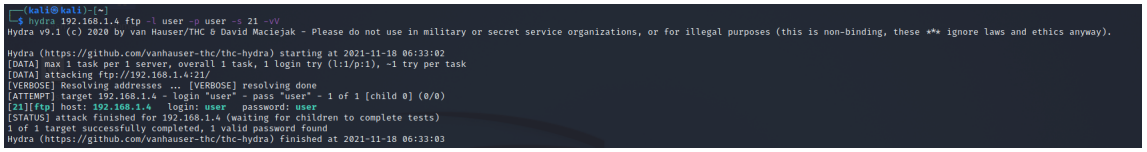


Figura 10: El ataque hacia el puerto 21 ha resultado exitoso.

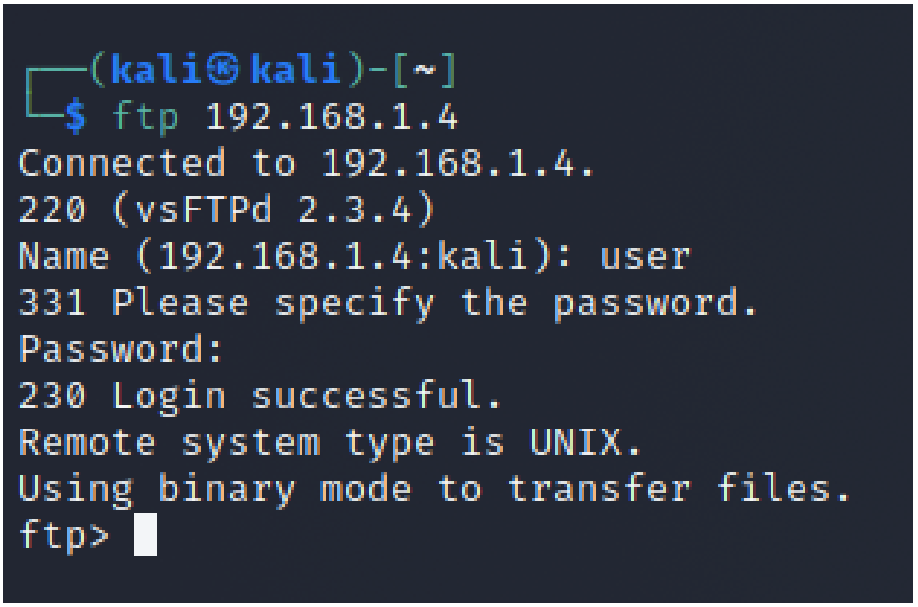


Figura 11: Comprobamos que con los credenciales podemos acceder

de escaneo podremos aprovecharnos de ello. Algunos de estos protocolos se describen a continuación.

## 3.1. SMB

SMB (*Server Message Block)* es un protocolo cliente/servidor que gobierna el acceso a archivos, directorios y recursos de una red como impresoras o interfaces compartidas. Este protocolo permite al cliente comunicarse con otros participantes de la red para acceder a los archivos que se comparten en ella. Por defecto utiliza el puerto 445.

Explotando este protocolo podemos conseguir desde el acceso a información o archivos que se comparten en la red hasta ejecutar comandos de forma remota en la gestión de los archivos compartidos. Un ejemplo lo tenemos en el **CVE-2020-0796**, una vulnerabilidad descubierta en el 2020 que permitía a los atacantes exponer la memoria del *kernel* del sistema de forma remota [1].

## 3.2. FTP

FTP (*File Transfer Protocol*) es un protocolo de transferencia de archivos. Es un protocolo que nos permite transferir archivos directamente de un dispositivo a otro. Los archivos se comparten entre ordenadores que estén conectados a Internet de forma directa y sin ningún intermediario. Aunque esta información compartida viaja sin cifrar (para ello surgió en 2001 una "actualización" de este protocolo denominada FTPS). Por defecto se utiliza el protocolo 21 para la conexión con el servidor y el puerto 20 para las transferencias de archivos.

Un ejemplo de explotación de este protocolo sería simplemente obtener una traza de FTP gracias a aplicaciones como **Wireshark** [5]. Ya que la información no se envía cifrada, con abrirla podríamos ver todo el tráfico, entre ello el usuario y la contraseña. Otro ejemplo sería acceder a un servidor FTP (conociendo la contraseña o que tuviese el acceso anónimo habilitado) y descargar los archivos que este tiene.

## 3.3. Telnet

El protocolo Telnet (*Telecomunication Network*) nos proporciona un método estándar para establecer una sesión de línea de comandos (consola) en un dispositivo de forma remota a través de una red. De forma predeterminada utiliza el puerto 23.

Un ejemplo de uso de este protocolo lo hemos encontrado durante la realización de las prácticas de la asignatura Redes de Computadores, donde nos conectábamos a los dispositivos CISCO mediante Telnet.

Explotando este protocolo podríamos conseguir el acceso al dispositivo en el que estuviese habilitado, además de obtener información de la conexión. Esto se debe a que dentro de este protocolo la información viaja sin ningún tipo de cifrado,

solamente en texto plano. En respuesta a este problema de seguridad, se popularizó el uso de otro protocolo denominado **SSH**.

## 3.4. SSH

SSH (*Secure Shell*) es un protocolo de administración remota que permite a los usuarios que lo utilicen controlar, majear y modificar servicios de forma remota a través de Internet. Todo ello gracias a un mecanismo de autenticación. Por defecto corre en el puerto 22.

Es decir, gracias a este protocolo podríamos usar SSH en nuestro servidor para conectarnos a él de forma remota desde una terminal. Si utilizamos Linux o Mac, podremos hacer uso de este protocolo con el comando:

```
ssh user@host
```

Un ejemplo de conexión real sería:

```
ssh testuser@10.0.0.55
```

Aunque si utilizamos Windows deberemos hacer uso de un cliente SSH como PuTTY [7].

Un ejemplo de explotación de este protocolo sería hallar los credenciales de conexión para un determinado servidor y, una vez los tengamos, conectarnos a él mediante SSH, haciendo todo lo que nuestros permisos nos permitan.

Para la explotación de estos protocolos podemos utilizar aplicaciones diseñada para ello. Algunos ejemplos se presentan a continuación.

## 3.5. enum4linux

**Enum4linux** [8] es una herramienta de enumeración capaz de detectar y extraer datos de los sistemas operativos Windows y Linux, incluidos aquellos que son *hosts* (SMB) en una red. Gracias a esta herramienta podemos conocer recursos compartidos en un dispositivo, el sistema operativo del objetivo, políticas de contraseña de un objetivo, e incluso listado de usuarios. Todo ello explotando el protocolo SMB.

### 3.5.1. Instalación

Esta herramienta debería estar instalada en nuestra máquina Kali. En caso de no ser así:

1. Actualizamos los paquetes:

   ```
   sudo apt update
   ```

2. Instalamos la aplicación:

   ```
   sudo apt install enum4linux
   ```

3. Comprobamos que la herramienta se ha instalado correctamente con el comando:

   ```
   enum4linux -h
   ```

### 3.5.2. Ejemplo de uso

Esta herramienta se utiliza con una sintaxis muy sencilla. Un ejemplo de uso es el siguiente:

```
enum4linux -U -o 192.168.1.200
```

Para verla en funcionamiento vamos a lanzar un sencillo ataque contra `Metasploitable 2`:

1. Abrimos la aplicación ingresando el comando `enum4linux` o desde el contenedor de aplicaciones (véase la Figura 12).
2. Levantamos la máquina `Metasploitable 2` (visto en la sesión anterior de esta práctica).
3. Lanzamos un breve escaneo a la IP de `Metasploitable 2` utilizando la herramienta **nmap** como hemos visto en la práctica anterior para comprobar el puerto asociado al protocolo SMB (véase la Figura 13).
4. Vemos que su estado es *open* y tiene un servicio corriendo, así que vamos a utilizar enum4linux para hacer una simple enumeración contra él gracias a la *flag* `-a`. Para ello:

   ```
   enum4linux -a 192.168.1.4
   ```

   Y vemos en la Figura 14 toda la información que la herramienta nos provee con un simple comando.

Para este sencillo ejemplo hemos utilizado la *flag* `-a`; sin embargo, enum4linux pone muchas más utilidades a nuestra disposición, como utilizar la *flag* `-U` para obtener

una lista de usuarios. Podemos obtener más información de todo lo que podemos hacer con la herramienta gracias al comando `enum4linux -h`.

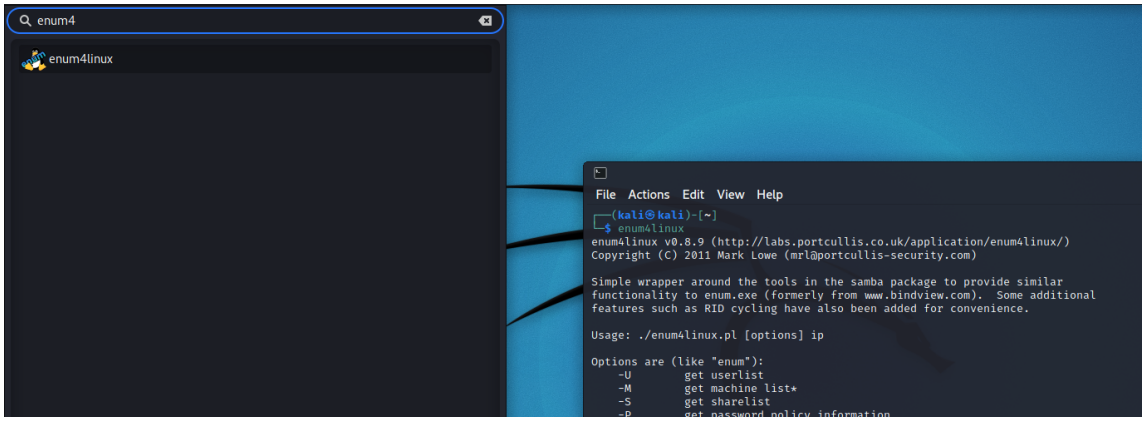


Figura 12: Abrimos la aplicación enum4linux.

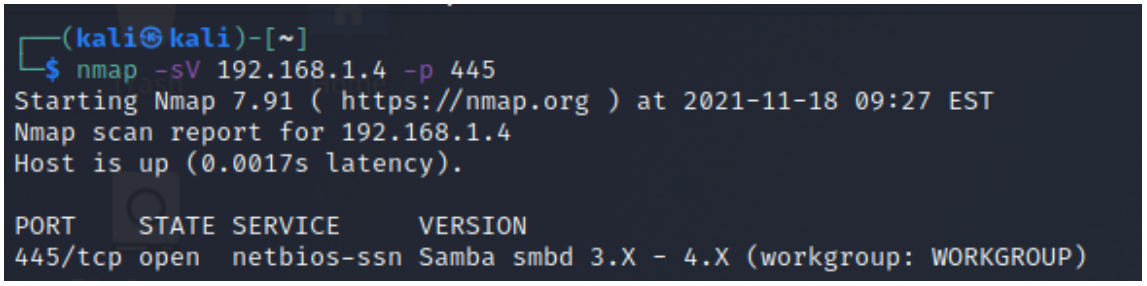


Figura 13: Comprobamos el estado del puerto 445.

## 3.6. smbmap

**Smbmap** [9] es una herramienta que permite enumerar las unidades y contenidos compartidos en todo un dominio, así como permisos de las unidades, e incluso ejecución de comandos remotos. Se diseñó para simplificar la búsqueda de datos confidenciales en grandes redes. Para ello hace uso de explotar el protocolo SMB.

### 3.6.1. Instalación

**MUY IMPORTANTE: La versión de Kali descargada incluye una versión desactualizada y no funcional de smbmap. Para seguir la práctica, será necesario actualizar la herramienta con estos pasos:**

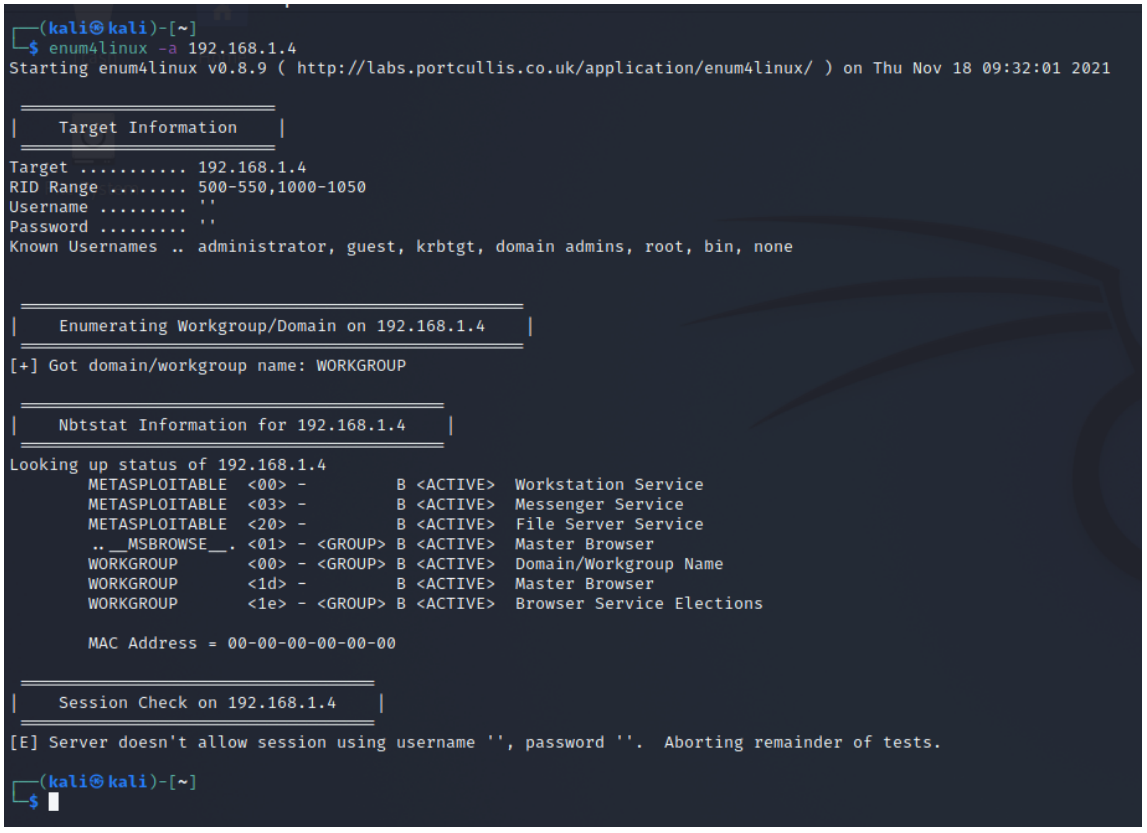


Figura 14: Utilizamos la herramienta enum4linux contra el puerto 445 de Metasploitable 2

1. Actualizar los paquetes del sistema:

   ```
   sudo apt-get update
   ```

2. Instalar la herramienta:

   ```
   sudo apt-get install smbmap
   ```

### 3.6.2. Ejemplo de uso

Para probar esta herramienta vamos a probar otro ataque contra el servicio SMB de la máquina `Metasploitable 2`:

1. Abrimos la aplicación con el comando `smbmap` o utilizando el contenedor de aplicaciones (véase la Figura 15).

2. Levantamos la máquina `Metasploitable 2` (visto en la sesión anterior de esta

práctica).

3. Lanzamos un breve escaneo a la IP de `Metasploitable 2` utilizando la herramienta **nmap** como hemos visto en la práctica anterior para comprobar el puerto asociado al protocolo SMB (véase la Figura 13).

4. Vemos que su estado es *open* y tiene un servicio corriendo, así que vamos a utilizar smbmap para ver qué podemos obtener. Si queremos obtener más información sobre la aplicación siempre podemos hacer uso del comando `smbmap -h`.

5. Lanzamos en primera instancia un escaneo simple para ver qué tipo de información nos brinda esta herramienta sin utilizar ninguna opción. Vemos el resultado en la Figura 16. El comando utilizado es simplemente indicarle la IP del *host*:

   ```
   smbmap -H 192.168.1.4
   ```

6. Si observamos la información obtenida, vemos que existe un disco `tmp` en el que podemos leer y escribir. Gracias a la *flag* `-r` podemos listar el contenido de dicho directorio (con otras *flags* podríamos borrar, descargar, e incluso subir archivos a dicho directorio). El comando a utilizar será:

   ```
   smbmap -H 192.168.1.4 -r tmp
   ```

   Como vemos en la Figura 17 se nos muestra todo el contenido de dicho directorio.

# 4. Ejercicios

Para profundizar y asentar los contenidos de esta práctica, realice los siguientes ejercicios y responda a las preguntas propuestas:

## 4.1. Ejercicio 1

En la práctica hemos visto que los ataques de fuerza bruta son comunes y fáciles de realizar por cualquier usuario. Imagina que has creado un sitio web en el que los usuarios deben *loguearse* para utilizarlo. Investiga 3 tecnologías o métodos adicionales al **captcha** con las que podrías evitar este tipo de ataques en tu sitio web y explica cómo contribuirían a anular los ataques de fuerza bruta.

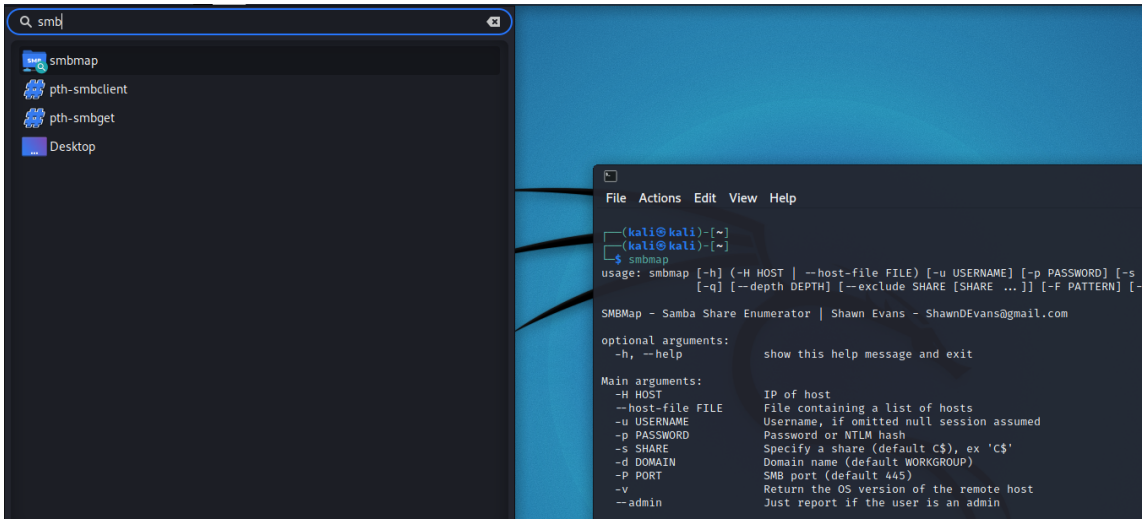


Figura 15: Abrimos la aplicación smbmap.

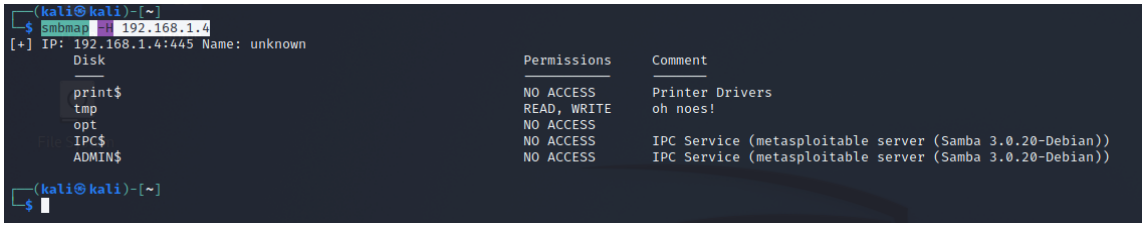


Figura 16: Indicamos a smbmap la IP del host para ver qué información nos provee.

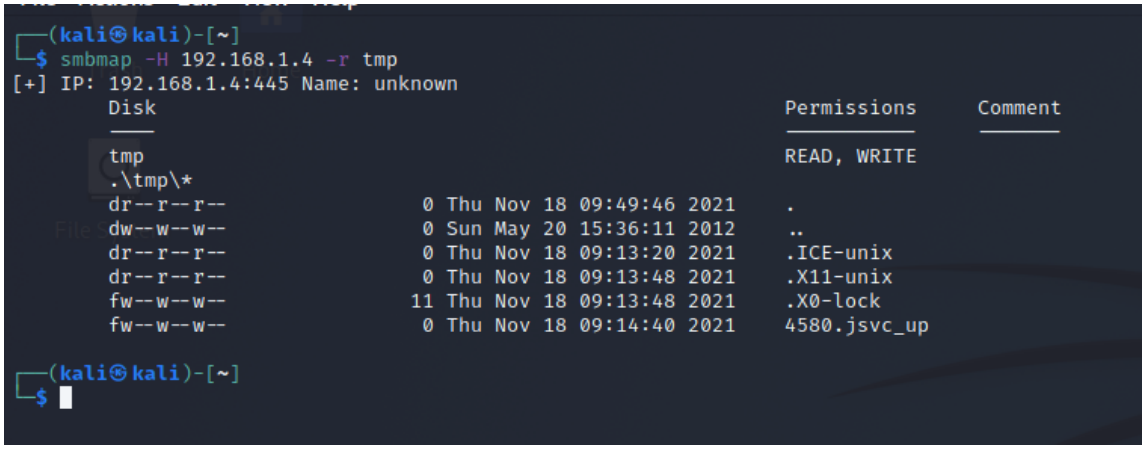


Figura 17: Listamos el contenido del directorio tmp.

## 4.2. Ejercicio 2

Utilizando la herramienta **Hydra**, realiza un ataque de fuerza bruta utilizando diccionarios (`user.txt` y `password.txt`) al servicio Telnet de la máquina Metasploitable 2 y responde a las siguientes preguntas:

- ¿Qué comando has utilizado para realizar el ataque?
- Ahora que sabes los credenciales, ¿cómo te conectarías por Telnet a Metasploitable 2?

## 4.3. Ejercicio 3

Utilizando la herramienta **enum4linux** para la explotación del protocolo SMB en la máquina de Metasploitable 2, responde a las siguientes preguntas:

- ¿Qué comando deberíamos utilizar si queremos obtener la lista de usuarios?
- Dentro de la información provista por la herramienta con el comando utilizado en el apartado anterior vemos una lista de usuarios conocidos, ¿cuál es?
- Si lanzamos un escaneo simple hacia la IP de la máquina de Metasploitable 2 podemos ver que nos da información sobre *`Nbtstat Information`*. ¿Qué es Nbstat? ¿Para qué sirve dentro de la seguridad informática?

## 4.4. Ejercicio 4

Gracias a **smbmap** y el conocimiento que hemos conseguido durante la práctica sobre los permisos en el directorio `tmp` de Metasploitable 2 responda a las siguientes preguntas:

- ¿Qué comando tenemos que utilizar si queremos listar el contenido del directorio `tmp`?
- ¿Qué comando utilizarías si quisieras autenticarte como *msfadmin msfadmin*?
- Autenticándote como *msfadmin*, sube un archivo `.txt` dentro del directorio `tmp`. ¿Qué comando has utilizado? A la hora de subir un archivo, en el directorio de destino tendremos que decirle cómo queremos que se llame ese archivo en el destino, por lo que si queremos subir un archivo al directorio `tmp`, el directorio destino será `tmp/ejemplo.txt`.
- Ante esta funcionalidad de subir archivos de forma remota se nos presentan una gran cantidad de posibles ataques. Si en caso de acceder al directorio `tmp` hubiéramos tenido acceso al directorio `htdocs`, ¿qué ataque malicioso podríamos haber llevado a cabo? Pista: quizás sea útil investigar el concepto de *webshell*.

# Referencias

[1] Pablo López Bonilla: *Vulnerabilidad crítica en SMB (CVE-2020-1206): SMBleed.* https://unaaldia.hispasec.com/2020/06/vulnerabilidad-critica-en-smb-cve-2020-1206-smbleed.html, visitado el 07/11/2024.

[2] brannondorsey: *Rockyou github.* https://github.com/brannondorsey/naive-hashcat/releases/download/data/rockyou.txt, visitado el 07/11/2024.

[3] byt3bl33d3r: *CrackMapExec GitHub oficial.* https://github.com/byt3bl33d3r/CrackMapExec, visitado el 07/11/2024.

[4] Nik Cubrilovic: *RockYou Hack: From Bad To Worse.* https://tcrn.ch/3njftwU, visitado el 07/11/2024.

[5] DigitalOcean: *Wireshark web oficial.* https://www.wireshark.org/, visitado el 07/11/2024.

[6] Openwall: *John the Ripper password cracker.* https://www.openwall.com/john/, visitado el 07/11/2024.

[7] Puttty.org: *Puttty web oficial.* https://www.putty.org/, visitado el 07/11/2024.

[8] OffSec Services: *Enum4linux Kali Tools.* https://www.kali.org/tools/enum4linux/, visitado el 07/11/2024.

[9] ShawnDEvans: *Smbmap Repositorio Oficial.* https://github.com/ShawnDEvans/smbmap, visitado el 07/11/2024.

[10] troyhunt.com: *Haveibeenpwned web oficial.* https://haveibeenpwned.com/, visitado el 07/11/2024.