

Introducción A NetCat

Netcat, también conocido como "nc", es una herramienta de red versátil y ampliamente utilizada en sistemas Unix-like y sistemas operativos similares a Unix. Se le conoce como la "navaja suiza" de las utilidades de red debido a su flexibilidad y capacidad para realizar una variedad de tareas relacionadas con la comunicación y transferencia de datos a través de redes.

Netcat actúa como un socket tanto en modo cliente como en modo servidor, lo que permite la transferencia de datos a través de puertos TCP o UDP. Puede utilizarse para establecer conexiones, realizar pruebas de red, transferir archivos, escanear puertos, realizar conexiones inversas y muchas otras tareas.

Netcat es una utilidad de línea de comandos que permite la comunicación entre dos puntos finales a través de una red. Se puede utilizar para crear túneles, transferir archivos y depurar aplicaciones de red. Netcat es una utilidad poderosa que se puede utilizar para una variedad de propósitos.

Algunas de las características de Netcat incluyen:

- Soporte para TCP y UDP
- Modo cliente y servidor
- Redirección de entrada y salida
- Creación de túneles
- Transferencia de archivos
- Depuración de aplicaciones de red

Netcat está disponible para una variedad de sistemas operativos, incluyendo Windows, Linux y MacOS. Es una herramienta gratuita y de código abierto.

Aquí hay algunos ejemplos de cómo se puede utilizar Netcat:

- Para crear un túnel entre dos computadoras, puede usar Netcat para crear un túnel TCP o UDP. Esto puede ser útil para transferir archivos o depurar aplicaciones de red.
- Para transferir archivos, puede usar Netcat para crear una conexión TCP entre dos computadoras y luego copiar los archivos a través de la conexión.
- Para depurar aplicaciones de red, puede usar Netcat para crear una conexión TCP o UDP entre su computadora y la aplicación de red y luego enviar y recibir mensajes a través de la conexión.

Algunos ejemplos de su uso incluyen la creación de servidores web rápidos para transferencia de archivos, la depuración de conexiones de red, la realización de pruebas de seguridad y la construcción de herramientas personalizadas para la manipulación de datos en red. Debido a su naturaleza poderosa, Netcat también puede ser utilizado en contextos maliciosos, por lo que se recomienda usarlo con responsabilidad y ética.

A mi personalmente, me gusta NetCat, además de porque efectivamente es una herramienta con la que se pueden hacer muchas cosas con redes, una especie de "navaja suiza" o herramienta multiusos, la gran utilidad de NetCat es que es ideal

para aprender y que queden fijados conceptos de redes que son siempre muy escurridizos.

No es que sean difíciles, pero si es difícil visualizarlos, y con NetCat, se visualizan muy bien.

Instalación De NetCAT

Ahora que ya sabemos qué es NetCat, veamos como se instala en Windows y En Linux.

Windows

NetCat, ahora es una parte del software Nmap. Todos sabemos ya mas o menos qué es y cómo funciona Nmap. Es una herramienta que todo profesional de la ciberseguridad debe de conocer si o si.

Bien, pues instalar NetCat es tan fácil como instalar Nmap. De modo que primero de todo instalamos Nmap de:

<https://nmap.org/>

Para comprobar que NetCat funciona en nuestra máquina Windows:

- Abrir una ventana de comandos como administrador
- teclear
- >ncat -v

Y saldrá esto:

```
C:\WINDOWS\system32>ncat -v
```

```
Ncat: Version 7.80 ( https://nmap.org/ncat )
```

```
Ncat: You must specify a host to connect to. QUITTING.
```

Linux

En nuestra máquina de ataque Parrot Security, NetCat ya está instalado. Si no:

```
apt-get update
```

```
apt install -y netcat
```

Luego, para comprobar que funciona, abrimos un terminal y tecleamos:

```
nc -h
```

Y si está correctamente instalado saldrá la ayuda.

NetCat

Bueno, pues ya sabemos como instalar NetCat, y suponemos que hemos instalado NetCat tanto en Windows, como en Linux. Yo pensaré que lo tenemos instalado en la máquina anfitriona y en la máquina de ataque Parrot Security.

Entonces ahora vamos a ver las cosas que se pueden hacer con NetCat. Y creo que como es una herramienta con la que se pueden hacer una gran variedad de cosas, muy diversas, lo mejor, en vez de soltar un rollo teórico, es aprender a usar la herramienta con ejemplos concretos. Voy a poner una secuencia de ejemplos, cuantos mas mejor, los voy a explicar, y cuando hayamos visto 20 o mas y los hayamos replicado en nuestro laboratorio, seguro que habremos aprendido a usar NetCat y un montón de cosas mas sobre redes.

Lo mismo dará hacerlo en nuestra máquina Parrot o la anfitriona.

NetCat funciona en modo **cliente-servidor** y opera con sockets.

Recordamos que un **socket** es el conjunto: **IP+puerto**

Si NetCat actúa como **cliente**, entonces, NetCat crea un socket con el destino indicado.

Si NetCat actúa como **servidor**, entonces, NetCat crea un socket en el puerto indicado.

Una vez conectado, bien como cliente, bien como servidor, NetCat, envía por el socket todo lo que llegue en su entrada estándar y envía a su salida estándar todo lo que llegue por el socket.

Algo tan simple resulta ser extraordinariamente potente y flexible como vas a ver e continuación. Por simplicidad se utilizan conexiones locales aunque, por supuesto, se pueden utilizar entre máquinas diferentes.

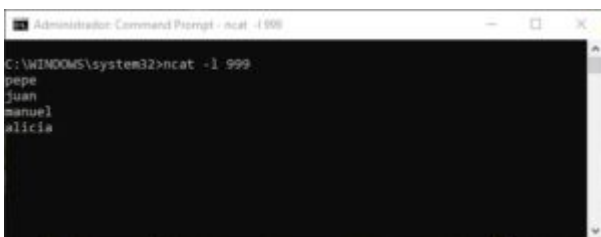
Chatear Dentro De Una Red Local

Desde nuestra máquina anfitriona Windows, abrimos dos terminales ejecutándose como administrador. En la primera de ellas tecleamos.

```
ncat -l 999
```

Esto crea un servidor que permanece a la escucha en el puerto 999. Es decir, el parámetro -l, ha creado un socket y ha hecho funcionar NetCat como servidor permaneciendo a la escucha en el puerto 999.

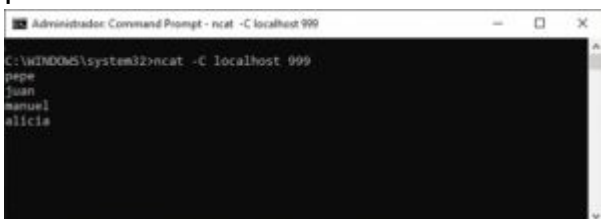
El socket utilizado es: localhost:999



En la otra ventana de línea de comandos, tecleamos:

```
ncat -C localhost 999
```

Este comando lo que hace es crear un cliente que se conecta al servidor por el puerto 999.



Como todo ocurre dentro de la misma máquina, que es la máquina anfitriona local Windows, su IP es localhost. El socket sería:

localhost:999

Bien pues ahora, ya podemos chatear entre el cliente y el servidor. Lo que tecleemos en la ventana del cliente aparecerá en la del servidor y viceversa.

Una vez que la conexión está hecha ya da igual quien hace de cliente y quien de servidor. Recordemos que la dirección IP simbolizada con localhost es 127.0.0.1 , es

decir cada vez que escribamos localhost nos estaremos refiriendo a esa IP que es la dirección reservada por la red a la propia máquina.

Para salir de ambas ventanas y liberar las conexiones, pulsar Ctrl+C

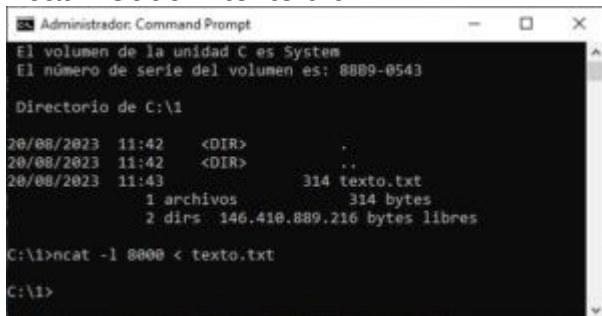
Creación De Servidor Que Sirve Un Archivo De Texto

Creamos una carpeta que contenga un archivo de texto cualquiera, llamado por ejemplo texto.txt, con un pequeño texto cualquiera dentro.

Ahora la idea es crear un servidor de tal forma que cualquier cliente que se conecte vea el contenido de ese archivo de texto, texto.txt

Creamos el servidor en el que se introduce el archivo:

```
ncat -l 8000 < texto.txt
```



```
Administrador: Command Prompt
El volumen de la unidad C es System
El número de serie del volumen es: 88B9-0543

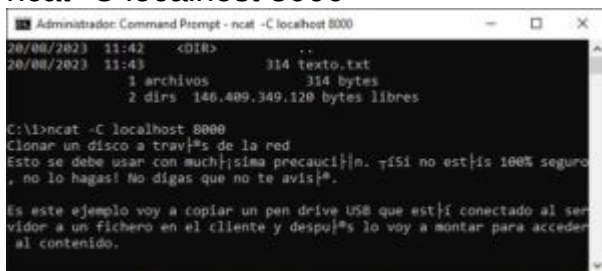
Directorio de C:\1

20/08/2023 11:42 <DIR>          ..
20/08/2023 11:42 <DIR>          .
20/08/2023 11:43              314 texto.txt
1 archivos              314 bytes
2 dirs 146.410.889.216 bytes libres

C:\1>ncat -l 8000 < texto.txt
C:\1>
```

Y ahora creamos el cliente:

```
ncat -C localhost 8000
```



```
Administrador: Command Prompt - ncat -C localhost 8000
20/08/2023 11:42 <DIR>          ..
20/08/2023 11:43              314 texto.txt
1 archivos              314 bytes
2 dirs 146.409.349.129 bytes libres

C:\1>ncat -C localhost 8000
Clonar un disco a trav s de la red
Esto se debe usar con mucha precauci n.  s  no est is 100% seguro
, no lo hag is! No dig is que no te avis  .

Es este ejemplo voy a copiar un pen drive USB que est   conectado al ser
vidor a un fichero en el cliente y despu s lo voy a montar para acceder
al contenido.
```

Y en el momento que creo el cliente, se conecta al servidor por el puerto 8000 que es el escogido e inmediatamente, el servidor le sirve texto.txt

Descarga De Archivos Del Servidor Al Cliente

Ahora la idea es la misma que el ejercicio anterior, pero en lugar de que el archivo, texto.txt se descargue en la salida est ndar (el monitor del cliente) queremos que el cliente lo almacene, copie o descargue en otro archivo de texto, llamado:

archivo_copiado.txt

Entonces creamos el servidor igual que antes:

```
ncat -l 8000 < texto.txt
```

Y ahora creamos el servidor:

```
ncat -C localhost 8000 > archivo_copiado.txt
```

En el momento que creamos el cliente, se crea el archivo archivo_copiado.txt pero con 0 bytes. Solo al terminar la conexi n con Ctrl+C se descarga el archivo.

Comprobarlo.

Con esta instrucci n lo que yo le digo al cliente es, oye, todo lo que recibas del servidor, te lo llevas al archivo, archivo_copiado.txt

Uso De NetCat Como Escaneador De Puertos

Podemos usar NetCat como escaneador de puertos. En este ejemplo escaneamos un rango de puertos entre el 20 y el 30. Desde la máquina de ataque Parrot Security, abrimos un terminal y tecleamos:

```
nc -z -v lamasiadechencho.com 20-30
```

```
DNS fwd/rev mismatch: lamasiadechencho.com != hl173.dinasever.com
```

```
lamasiadechencho.com [82.98.134.147] 25 (smtp) open
```

Fijaros que desde Linux, la instrucción para NetCat es nc y no ncat.

Uso De NetCat Para Averiguar Versiones De Servicios

Un servicio es un software que proporciona una función específica para otros programas. Un socket es un punto de conexión entre dos procesos que se comunican entre sí a través de una red. En el contexto de los servicios, un socket es el punto de conexión entre un cliente y un servidor. El cliente utiliza el socket para conectarse al servidor y solicitar un servicio. El servidor utiliza el socket para responder a la solicitud del cliente y proporcionar el servicio requerido.

Por ejemplo, un servidor web es un servicio que proporciona acceso a archivos web. Cuando un usuario introduce una dirección web en un navegador, el navegador utiliza un socket para conectarse al servidor web. El servidor web utiliza el socket para responder al navegador y proporcionar el archivo web solicitado.

Otro ejemplo es un servidor de correo electrónico. Cuando un usuario envía un correo electrónico, el cliente de correo electrónico utiliza un socket para conectarse al servidor de correo electrónico. El servidor de correo electrónico utiliza el socket para recibir el correo electrónico y entregarlo al destinatario.

Los sockets son una parte esencial de la comunicación en red. Permiten a los programas de computadora conectarse entre sí y compartir información.

Entonces, ahora que sabemos qué es un servicio, vamos a usar NetCat para averiguar la versión del servicio que se está proporcionando al socket por parte del servidor.

Sabemos que un socket es: IP+puerto.

Abrimos la máquina Ubuntu de nuestro laboratorio. La contraseña de ésta máquina es toor:

Desde la máquina de ataque Parrot Security, abrimos un terminal y tecleamos:

```
sudo su
```

```
nmap -sV -v --script nbstat.nse 10.10.1.9
```

Con esta instrucción sabremos los puertos que están abiertos y la versión de los servicios que se ejecutan en ellos.

Esto mismo también lo podemos hacer con NetCat así:

```
echo "EXIT" | nc 10.10.1.9 22
```

```
echo "EXIT" | nc 10.10.1.9 80
```

En ambos casos devolverá el software que se está ejecutando en el servicio. He puesto el puerto 22 y el 80 porque son los que me ha dicho Nmap que están abiertos.

Recordemos que 10.10.1.9 es la IP de la máquina Ubuntu.

Recuperación De Una Página Web

Es posible recuperar el contenido de una página web, lo que es el código html, conectándose al servidor mediante NetCat en el puerto 80, la instrucción sería desde la máquina de ataque Parrot:

```
printf "GET /nc.1 HTTP/1.1\r\nHost: man.openbsd.org\r\n\r\n" | nc man.openbsd.org 80
```

El resultado es la página html de la url:

man.openbsd.org

Le hemos hecho pasar una instrucción propia del protocolo HTTP. Esa instrucción es GET. Todos los navegadores entiendes esas instrucciones para poder navegar por internet.

Instalación De Socat

Socat es como NetCat, pero mucho mas avanzado. Aquí solamente vamos a ver la instalación en Linux y Windows. Solo nos quedaremos con la instalación, que no es fácil, y con la idea de que Socat sirve para hacer tunelizaciones entre procesos, lo cual es muy importante para hacer desplazamientos laterales en escalada de privilegios dentro del proceso de hacking.

Linux

En la máquina atacante Parrot Security ya viene instalado. En cualquier máquina Ubuntu, hacemos:

```
sudo apt-get update  
sudo apt-get install socat
```

Windows

Aquí es bastante mas complicado, porque no hay una aplicación nativa, que yo sepa, de socat en Windows. Entonces hay que como ejecutarla en una especie de capa de Linux por encima de Windows.

Para ello, lo primero que haremos será instalar cygwin. Desde nuestra máquina anfitriona, creamos la carpeta socat en la parte del disco que queramos , en el Desktop, en otro disco etc. Yo recomiendo hacer la carpeta en el disco C, para evitar líos después. En mi caso:

C:\Users\Juan\Desktop\CIBERSEGURIDAD\Socat

Pero vosotros tendréis una localización diferente. Quedaros con el PATH de esa localización.

Y descargamos el último archivo de instalación de socat en:

C:\Users\Juan\Desktop\CIBERSEGURIDAD\Socat

que en mi caso fue:

<http://www.dest-unreach.org/socat/download/socat-1.7.4.4.tar.gz>

Pero que en vuestro caso puede ser otro mas reciente.

Después navegamos hacia:

<https://www.cygwin.com/>

Y descargamos el instalador de cygwin y seguimos las instrucciones de instalación.
Hay que asegurarse de que se instalan los paquetes:

```
gcc-g++  
gcc-core  
cygwin32-gcc-g++  
cygwin32-gcc-core  
make
```

En la lista de paquetes a instalar hay que buscarlos y asegurarse de que se instalan.
Una vez instalado cygwin, lo ejecutamos, y lo que sale es un terminal tipo Linux pero sobre nuestra máquina anfitriona. Dentro de ese terminal tenemos que desplazarnos a la carpeta:

C:\Users\Juan\Desktop\CIBERSEGURIDAD\Socat

y, dentro del terminal, ejecutar las instrucciones:

```
tar xzvf socat-1.7.4.4.tar.gz  
cd socat-1.7.4.4  
./configure  
make  
make install
```



```
cygwinPC-F13D-4305 ~  
$ cd /  
cygwinPC-F13D-4305 /  
$ ls  
Cygwin-Terminal.ico  Cygwin.ico  cygdrive  etc  lib /sbin  usr  
Cygwin.bat          bin         dev      home  proc  tmp  var  
cygwinPC-F13D-4305 /  
$ cd cygdrive  
cygwinPC-F13D-4305 /cygdrive  
$ ls  
c  d  e  f  g  h  i  j  }
```

cygwinPC-F13D-4305 /cygdrive/c		
Recycle.Bin	ProgramData	cygwin64
DefaultAgent	QT	hamlib
3	Recovery	liberf1.sys
ADVANCED IP SCANNER	SAMD	testsub
Archivos de programa	SQL2019	paperfile.sys
Documents and Settings	System Volume Information	sqltreadwin
DumpStack.log	TCPview	swapfile.sys
DumpStack.log.tmp	UTILIDADES	tools
NETADMIN MANAGER	Users	usr
Program Files	Windows	
Program Files (x86)	android	

```
cygwinPC-F13D-4305 /cygdrive/c  
$ cd /Users/Juan/Desktop/CIBERSEGURIDAD/Socat  
-bash: cd: /Users/Juan/Desktop/CIBERSEGURIDAD/Socat: No such file or directory  
cygwinPC-F13D-4305 /cygdrive/c  
$ cd /Users/Juan/Desktop/CIBERSEGURIDAD/Socat  
cygwinPC-F13D-4305 /cygdrive/c/Users/Juan/Desktop/CIBERSEGURIDAD/Socat  
$
```

Y en teoría, si todo va bien ya estaría instalado socat. Hay que ejecutarlo siempre abriendo el terminal cygwin.

tecleamos:

socat -h

Y debería de salir la ayuda si va todo bien.

```
/cygdrive/c/Users/Juan/Desktop/CIBERSEGURIDAD/Socat
$ socat -h
socat by Gerhard Rieger and contributors - see www.dest-unreach.org
usage:
socat [options] <h1-address> <h1-address>
options:
-h          print version and feature information to stdout, and exit
-h1-f       print a help text describing command line options and addresses
-hh         like -h, plus a list of all common address option names
-hh1        like -hh, plus a list of all available address option names
-d[ddd]     increase verbosity (use up to 4 times; 1 are recommended)
-o          analyze file descriptors before loop
-l[facility] log to syslog, using facility (default is daemon)
-lf[logfile] log to file
-ls         log to stderr (default if no other log)
-lt[facility] mixed log mode (stderr during initialization, then syslog)
-lp[program] set the program name used for logging
-ls         use microseconds for logging timestamps
-ls         add hostname to log messages
-v          verbose text dump of data traffic
-x          verbose hexadecimal dump of data traffic
-r <files>   raw dump of data flowing from left to right
-R <files>   raw dump of data flowing from right to left
-b[size,t]  set data buffer size (BKB)
-s         sloppy (continue on error)
-t[timeouts] wait seconds before closing second channel
-t[timeouts] total inactivity timeout in seconds
-u          unidirectional mode (left to right)
-U          unidirectional mode (right to left)
-g          do not check option groups
-L clockfile try to obtain lock, or fail
-W clockfile try to obtain lock, or wait
-A         prefer IPv4 if version is not explicitly specified
-k         prefer IPv6 if version is not explicitly specified
h1-address:
  pipe[,<opts>]    groups=FD, FFD
  <single-address>|<single-address>
  <single-address>
single-address:
  <address-head>[,<opts>]
address-head:
  <create>[:<filename>] groups=FD, REG, NAMED
  exec[:<command>] lines groups=FS, FIFO, SOCKET, EXEC, FORK, TURNIDS, PTY, PARENT, UNIX
  fd:<name> groups=FD, FIFO, CWD, BLK, REG, SOCKET, TURNIDS, UNIX, IPA, IP4, IP6, TCP, UDP, SCTP
```