Netcat.

Sacandole Provecho a una exelente Utilidad.

por: Kliber.

Netcat es un pequeño programa creado para uso de los administradores de redes (y por supuesto para los Hackers) :), este proggie fué creado originalmente por Hobbit y porteado a Win95 y NT por Weld Pond de L0pht , tiene mas de un año desde que fué Liberado y muy poco se ha escrito sobre este Programita; Principalmente porque la estructura de sus comandos es poco familiar para el usuario medio. Netcat tiene infinidad de funciones, aunque se deja que sea el usuario quien las averigue :P, y en el archivo de ayuda ponen algunos ejemplitos muy elementales solamente...

La especialidad de NetCat es el Protocolo tcp/ip, y le dá a la máquina de windows, cierto poder sobre este protocolo que solo tenía UNIX, trabaja con lineas de comandos desde MS-DOS (o desde el Shell de Linux), y según parece, puede hacer casi cualquier cosa sobre TCP/IP. El comando principal es nc con su respectiva variable u opción al mas puro estilio Unix.

Cabe destacar que la información sobre Netcat y sus usos especificos es bastante limitada; aunque Hobbit en su documento aclara muchas cosas, cita algunos ejemplos y dice que NetCat puede ser utilizado para mas de 1001 vainas...

Netcat puede ser encontrado en: http://www.l0pht.com/~weld/netcat

Netcat en WinX

==========

Este es el resultado de el comando de ayuda de netcat en una máquina windows

```
 c:>nc -h

connect to somewhere:   nc [-options] hostname port[s] [ports] ...
listen for inbound:     nc -l -p port [options] [hostname] [port]
options:
        -d              detach from console, stealth mode

        -e prog         inbound program to exec [dangerous!!]
        -g gateway      source-routing hop point[s], up to 8
        -G num          source-routing pointer: 4, 8, 12, ...
        -h              this cruft
        -i secs         delay interval for lines sent, ports scanned
        -l              listen mode, for inbound connects
        -L              listen harder, re-listen on socket close
        -n              numeric-only IP addresses, no DNS
        -o file         hex dump of traffic
        -p port         local port number
        -r              randomize local and remote ports
        -s addr         local source address
        -t              answer TELNET negotiation
        -u              UDP mode
        -v              verbose [use twice to be more verbose]
        -w secs         timeout for connects and final net reads
        -z              zero-I/O mode [used for scanning]
port numbers can be individual or ranges: m-n [inclusive]
```

Bien; un analisis rápido de estas variables nos da una idea del potencial de este pequeño programa y las infinitas posibilidades que nos ofrece el poder manejar conecciones de una manera tan básica y sencilla:

<------ Opciones de Netcat --------->

-d (Modo Stealth o encubierto)

Esta opción desvincula al Programa de la consola, haciendolo trabajar en el BackGround.

-e<prog> (Ejecuta un programa cuando se conecta)

Puede ser utilizado para ejecutar incluso un Shell tanto en WinX como en *NIX.

-l (Escuchando conecciones)

Deja a un puerto abierto en espera de una conección

-L (lo mismo que anteriormente pero sigue escuchando aún cuando la conección es cerrada)

Esta opción es incluida en la versión de Weld Pond de L0pth, y es muy util para seguir escuchando en el puerto, a diferencia de -l (que la conección cerrada termina con el proceso de nc) esta opción -L permite seguir escuchando en el mismo puerto (la rutina de nc -l es reiniciada).

-n (Dirección numerica especifica; no hace un DNS Lookup) 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.

-o<logfile> (obtiene un archivo log en Hex de la acción) Genera un Log de las actividades de netcat en código Hexadecimal.

-p<puerto> (Puerto para pegarse) Algunas veces debes especificarle con esta
opción el puerto a realizar una acción.

-s<ip addr> (pegarse a un IP especifico) Netcat puede utilizar IP de una red como fuente local.

-t (Funciona como un pequeño demonio telnet) Con esta opción le especificas
a netcat que debe realizar negociaciones telnet.

-u specify UDP (Utilizar Protocolo UDP) Con esta opción le dices a netcat que
trabaje con protocolo UDP en vez de TCP.

-v (modo verbose, mas información, se le puede añadir otra -v para mas info todavia) Bastante util y necesario, sobre todo para estudiar demonios en profundidad y observar todos los detalles en un Sniffing.

-w <segundos> (Especifica un tiempo para terminar) Con esta opción le especificas un
tiempo determinado para realizar conecciones .

-r (Genera un Patron Ramdom de puertos locales o remotos) Muy util para evitar
patrones lógicos de Scanning.

-g <gateway> (especificar Gateways) Una de las opciones más interesantes de netcat,
permite utilizar Routers como "puentes" de conección.

-G <numero> (Especificar puntos de Routing), Con esta opción podemos crear
una cadenaaleatoria de hosts para crear un ruta perdida para tus paquetes (Spoofing).

-i <segundos> Especifica un intervalo de segundos entre puertos Scaneados.

<---------- Fin de las opciones comentadas ------>

Netcat en Linux

==========

Netcat en una plataforma como Linux se convierte en una utilidad muy potente, pudiendo ser utilizado en conjunto con lenguajes de programación
como Perl y C , o bien desde la propia Linea de comandos del poderoso Shell de Linux mediante Shell Scripts.

Cabe destacar que distribuciones como RedHat Linux trae junto con sus paquetes de instalación una versión limitada de netcat; lo mas recomendable
es bajar de la red la versión full de netcat para Linux (Importante: La versión de netcat para linux viene a prueba de lamers, por lo cual debemos compilar a netcat con unos flags especiales para poder obtener las opciones -t y -e (Telnet y Gaping Security Hole)). Bajas el .tar de netcat y lo desempaquetas en el directorio de tu preferencia, te ubicas dentro del directorio de netcat y lo compilas con Make utilizando las siguientes Flags:

```
[root@DarkStar] #make linux DFLAGS=" -DTELNET -
DGAPING_SECURITY_HOLE"
```

Copias el binario (nc) al directorio /usr/bin , de esta manera podras usar netcat directamente llamandolo de cualquier parte del Shell, ademas de que podrás usar los scripts que hagas (o consigas en la red) sin problemas; netcat
trae unos scripts muy interesantes y bien comentados para que los estudies y comprendas mejor su implementación en scripts, los scripts están en el

directorio donde desempaquetastes netcat en /scripts , los corres como siempre: ./probe (o el script que quieras).

Utilizando Netcat.

==========

Para ilustrar mejor como trabajamos con este programa, lo mejor es observar ejemplos prácticos y analizar su estructura para poder comprender mejor como funciona y así poder crear nuestras propias aplicaciones.

Algunas de las cosas que podemos hacer con NetCat son:

Obtener un Shell rapidamente en una máquina remota usando la opción -l (Listen) conjuntamente con la opción -e (ejecutar) , cuando el proggie corre con estas variables y la conección es realizada, NetCat ejecuta el programa elegido y se conecta a stdin y stdout del programa en la conección a la red.

nc -l -p 23 xxx.xxx.xxx.xx 23 -t -e cmd.exe

Este comando dejará a NetCat escuchando el Puerto 23 (telnet) , cuando es conectado a travéz del cliente, ejecutará un Shell (cmd.exe) la opción -t le dice a NetCat que maneje cualquier negociación que el cliente pueda esperar….

Si esta conección es realizada desde una máquina NT, el shell correrá los permisos del proceso que han generado a NetCat (Hmmm...) así que hay que ser muy cuidadosos :)

La belleza de NetCat es que puede hacer lo mismo en CUALQUIER puerto :) Puedes dejar a NetCat escuchando en los puertos NETBIOS, que están probablemente corriendo en la mayoría de las máquinas NT, de esta

manera puedes lograr una conección a una máquina que esté utilizando "Filtrado de Puertos" activado en TCP/IP security Network Control Panel, NT no parece tener ninguna seguridad alrededor de cuales puertos los programas de usuarios son permitidos amarrar, esto quiere decir en pocas palabras, ejecutar comandos y programas que puedan unirse a los Puertos NETBIOS.

Como anteriormente se mencionó, puedes utilizar a Netcat para estudiar diferentes puertos, con la siguiente sintaxis:

c:\>nc -v <IP> <puerto> (puedes añadir otra -v)

Uno de los puertos mas interesantes a la Hora de Analizar un Host, es el puerto 79 (Finger) , puedes obtener nombres de usuarios e información muy util a la hora de planear un "Brute-Force Attack", este comandito de Netcat te muestra la Flexibilidad del Proggie en cuestion, dandote una idea de sus posibilidades:

c:\>nc -v <host> 79 < user.txt > log.txt

El comando anterior le dice a netcat que se conecte en modo verbose al Host predeterminado en el puerto 79 (Finger) y envie el contenido del archivo user.txt (OJO: no he probado esto con una posible lista de nombre de usuarios al azahar), la respuesta del servicio será guardada en el archivo log.txt

Scanner:

========

Netcat puede ser utilizado como scanner, sus multiples opciones

le permiten realizar un gran número de combinaciones, pudiendo realizar Scannings en Puertos Random, en puertos conocidos, en modo ascendente o
descendente, con intervalos de tiempo, utilizando gateways para evitar mostrar la IP fuente del Scanning, etc.

```
C:\nc11nt>nc -v -v -z 127.0.0.1 53 25 21

DNS fwd/rev mismatch: localhost != darkstar
localhost [127.0.0.1] 53 (domain): connection refused
localhost [127.0.0.1] 25 (smtp): connection refused
localhost [127.0.0.1] 21 (ftp): connection refused
sent 0, rcvd 0: NOTSOCK
```

Pues si; aqui tienen un pequeño y primitivo scanner, se le pueden añadir puertos escogidos como en el ejemplo anterior o asignarle un rango de puertos:

```
C:\nc11nt>nc -v -v -z 127.0.0.1 1-53

DNS fwd/rev mismatch: localhost != darkstar
localhost [127.0.0.1] 53 (domain): connection refused
localhost [127.0.0.1] 52 (?): connection refused
localhost [127.0.0.1] 51 (?): connection refused
localhost [127.0.0.1] 50 (?): connection refused
localhost [127.0.0.1] 49 (?): connection refused
localhost [127.0.0.1] 48 (?): connection refused  etc...
```

Volvemos con la opción -v (verbose) y la Opción -z (zero i/o) que es usada para scanning, los puertos se lo especificamos al final del IP del host, bien sea individuales separados por un espacio; o por un rango de puertos.

Sniffer:

========

Otra de las interesante posibilidades de netcat es su capacidad para escuchar conecciones en cualquier puerto, pudiendo redireccionar todo el tráfico del mismo hacia un archivo o hacia pantalla, en este sencillo ejemplo, podemos observar las bases de un sencillo sniffer en Windows:

```
C:\nc11nt>nc -v -v -L 127.0.0.1 -p 23

DNS fwd/rev mismatch: localhost != darkstar
listening on [any] 23 ...
DNS fwd/rev mismatch: localhost != darkstar
connect to [127.0.0.1] from localhost [127.0.0.1] 1131
login: sniffado
password: jeje!!
puedo ver todo lo que escriben aqui... Muuuaaahahahahahah!!! B-]
```

Tambien podemos redireccionar toda la salida a un archivo e irnos a realizar otras actividades ,ientras netcat hace su trabajo:

```
C:\nc11nt>nc -v -v -L -p 23 127.0.0.1 -t >login.txt
DNS fwd/rev mismatch: localhost != darkstar
listening on [any] 23 ...

[Aqui viene la conneción...]

DNS fwd/rev mismatch: localhost != darkstar
connect to [127.0.0.1] from localhost [127.0.0.1] 1030

[Todo lo que escriba la connección se va al archivo login .txt]
```

sent 0, rcvd 42

[La opción -L permite que netcat escuche nuevamente al terminar la conección,
"New Victims Wanted" Hehe!]

DNS fwd/rev mismatch: localhost != darkstar
listening on [127.0.0.1] 23 ...

El Exploit-Explained: nc -v -v -L 127.0.0.1 -p 23

Ejecutamos a Netcat con la opción o variable -v (verbose) (doblemente "verbose" por si acaso) ;) esto hará que el resultado de netcat, sea mostrado directamente en pantalla (a diferencia del archivo usado por Dr._X) , la opción o variable -L (Listen, and listen again) nos permitirá dejar escuchando u "oliendo" en determinado puerto aun cuando la conección sea interrumpida (listen again), con la variable -p le indicamos el puerto...

Al ejecutar a netcat con esa combinación de variables las opción -v me indica en pantalla el Host y el puerto de escucha:

DNS fwd/rev mismatch: localhost != darkstar
listening on [any] 23 ...

Realizo desde otra ventana un telnet a localhost (127.0.0.1) en el puerto 23, netcat me informa sobre lo que ocurre en el puerto 23:

DNS fwd/rev mismatch: localhost != darkstar
connect to [127.0.0.1] from localhost [127.0.0.1] 1131
login: sniffado

Voilá! un Sniffer en LocalHost! Jajaja!!!

Detector de Conneciones Sospechosas:

==========

La posibilidad de dejar a netcat escuchando en determinados puertos, nos permite crear una especie de "trampa" para un supuesto agresor que utilize
scanners, o herramientas tales como NetBus o BackOrifice en contra de nuestras estaciones. Incluso, podemos crear un archivo que haga un Flood y redireccionar su salida hacia la estación agresora en caso de una conección
no autorizada a determinado puerto. (jeje! y se me ocurren un monton de cosas
 más, Muaahahaha!) :)

Este es un ejemplo de un detector de BO, Je! y funciona! este es un ejemplo real de un dia como cualquier otro en IRC; he aquí el ejemplo:

C:\nc11nt>nc -u -v -v -L -p 31337 127.0.0.1 31337
DNS fwd/rev mismatch: localhost != darkstar
listening on [any] 31337 ...

invalid connection to [0.0.0.0] from nas1-064.ras.bqm.cantv.net
[161.196.246.65]
 31338

Back Orifice utiliza el protocolo UDP para realizar sus travesuras, realiza la conección desde un puerto aleatorio (casi siempre el 1080) aunque en este caso lo hizo desde el 31338 (posiblemente una variante de BO), por eso se utiliza la opción -u (protocolo udp) , netcat se queda esperando conecciones UDP en el puerto 31337 (default de BO) , cuando alguien hace un sweep a tu IP netcat lo detecta enviando a pantalla el IP y el DNS del agresor...

Luego un pequeño "Ping of Death" (Nuke) para el transgresor y le hacen un Scan para ver cuando desaparece B-]

nas1-064.ras.bqm.cantv.net [161.196.246.65] 48 (?): connection refused
nas1-064.ras.bqm.cantv.net [161.196.246.65] 47 (?): connection refused
nas1-064.ras.bqm.cantv.net [161.196.246.65] 46 (?): connection refused
nas1-064.ras.bqm.cantv.net [161.196.246.65] 45 (?): TIMEDOUT
nas1-064.ras.bqm.cantv.net [161.196.246.65] 44 (?): TIMEDOUT<--Chao!!!
Jeje!!

Otros usos Miscelaneos:

==========

Puedes utilizar algo de ingienería social para capturar algunos passwords con netcat, por ejemplo, si una máquina no tiene abierto el puerto de FTP o de telnet, creas un archivo de texto que solicite el ID y el Password de la víctima; algo así:

Microsoft Internet FTP Server V.5.9 [Beta]
04/16/99 myhost.com
Please introduce Username, password and press "Enter"
LogOn:

Luego redireccionas el archivo hacia la victima:

C:\nc11nt>nc -v -v -L -p 21 nombre del host -t < login.txt

Si el tonto cae... Ahí va tu password, Jeje!! B-] un poco de imaginación y maña te permitirán encontrar muchas utilidades para netcat.

Netcat en Vez de Telnet.

==========

Yo personalmente prefiero usar netcat para realizar connecciones remotas como alternativa al Telnet. la ventaja de realizar conecciones telnet desde netcat es que este esconde "algo" sobre tu conección, lo que lo hace mas "sigiloso" que telnet, (de ahí por que lo llamaron netcat), Realizando una conección "Limpia" en determinado puerto, obviando las negociaciones comunes de Telnet que pueden confundir al cliente en determinados casos, como por ejemplo, al utilizar ciertas Backdoors muy conocidas en Unix.

OJO: algunas máquinas interpretan al cliente de telnet y asumen el nombre del usuario que lo utiliza, de allí el porqué algunos servidores solo preguntan por password ; teoricamente netcat no envia esta información. Por eso, es recomendable acostumbrarse a utilizar netcat para hacer conecciones remotas:

c:> nc -v nombre del host 23(o el puerto de tu preferencia)

Netcat y Programación:

==========

Esta combinación desencadena todo el Poder de Netcat en su máxima expresión; Tratandose de una herramienta que funciona con lineas de comandos,
su integración con un lenguaje de programación le permite realizar gran cantidad de tareas, y posibilidades se van descubriendo dia a dia con su inclusión en nuevos Scripts y Exploits.

Muchos ScriptKiddies que no tienen idea de lo que hacen, se sienten frustrados porque muchos de los Scripts y Exploits que bajan de la Red simplemente no les funciona, porque no saben interpretar el Código y por lo tanto son incapaces de efectuar las modificaciones necesarias para incluir librerias, paths o utilidades necesarias para su funcionamiento.
(Jódanse ScriptKiddies!!! Jajaja!!)

Netcat es exelente para implementar exploits remotos, permitiendo enviar el código a cualquier puerto vulnerable con una simple orden, logrando ejecutar todos los comandos necesarios para explotar determinados servicios.

Varios exploits que circulan actualmente en la Red, usan a netcat como "motor" para manejar las conecciones, si analizamos el código de estos programas podemos observar un nc por ahí, esto significa que el Proggie en cuestión necesita una versión correctamente compilada de netcat en el directorio /usr/bin . A continuación un pequeño programa realizado por el Doctor_X de Hven utilizando a netcat:

<---------- Hven Port Scanner!! ------>

```
# !/bin/bash
# Scanner de Puertos
# By DoctorX 17/04/99  email: d0ct0r_x@bactery.8m.com
# Zona de Bacterias  http://bactery.8m.com
# Hackers de Venezuela http://www.hackhour.com.br/hven
```

```
# Este es un shell script hecho por mi para la verificacion de
# conexiones a un host utilizando netcat.

# Declaracion de Variables

export NETCAT=" nc -v -v -w 8 -z "
export RANGO=$2
LOCALHOST=$(uname -n)
export PUERTOS="21 23 25 79 80 110 111 113 139 143 513 514 515 6000
31337"
export MEM1="Scanner de Puertos "
export MEM2="by "
export MEM22="para Hackers Venezuela"
export MEM3="Victima : "
export MEM4="Falta el GateWay para el Source Routing !!!!!!"
export MEM5="Te van a pillar !!!!!!! $USER jejejejeje "
export MEM6="Local Host : $LOCALHOST "
export MEM7="UDP Scan "
export MEM8="http://www.hackhour.com.br/hven"
export re=" [5m"
export cl=" [0m"
export rojo=" [31m"
export email="email:d0ct0r_x@bactery.8m.com"

# Declaracion de Funciones

# Mensaje cuando no se le dan Parametros

function mem() {
local uso="uso  :$0  [opcion]  <host> <gateway>"
local DRX="DoctorX"
echo $MEM1
echo $MEM2   ${rojo}$DRX${cl}  $MEM22
```

```
echo $MEM8
echo ${rojo}$uso${cl}
echo "<host>             :IP/HOSTNAME de La Victima jejejeje "
echo "<gateway>          :source-routing , es opcional "
echo "opciones : "
echo "u                  :esta opcion de utiliza para hacer scan udp"
echo "so                 :Determinacion de SO de servidores Web"
echo "r rango_de_puertos :Cambia el rango de puertos por defecto :plow-phi"
&&
exit ; }

# Mensaje Inicial

function mem2() {
VICTIMA=$1
echo $MEM1
echo $MEM2 ${rojo}DoctorX${cl} $MEM22
echo $MEM3  $VICTIMA
echo $MEM6 ; }

# Mensaje 2

function mem_web() {
mem_web1="Hackers Venezuela"
mem_web2="By"
mem_web3="Victima : "
VICTIMA=$1
mem_web4="Determinacion de SO en Web Servers"
echo $mem_web1
echo $mem_web4
echo $mem_web2 ${rojo}DoctorX${cl} $email
echo $mem_web3 $VICTIMA ;}
```

```
# Scan Tcp

function tcp() {
 HOST=$1
 $NETCAT  $HOST $PUERTOS ; }

# Scan Tcp con Rango

function tcp_rango() {
 HOST=$2
 RANGO=$1
 $NETCAT  $HOST $RANGO ; }

# Scan UDP

function udp() {
VICTIMA=$1
echo $MEM7
$NETCAT  -u $VICTIMA $PUERTOS ; }

# Scan UDP con gateway

function udp_gateway() {
echo $MEM7
VICTIMA=$2
GATE=$1
NETCAT_GATE="nc -v -v -z -u $VICTIMA -g $GATE "
$NETCAT_GATE $PUERTOS ; }

# Scan con Source Routing

function tcp_gateway() {
```

```
GATE=$1
HOST=$2
RANGO=$PUERTOS
echo "Gate : $GATE "
$NETCAT -g $GATE  $HOST $RANGO ; }

# Advertencia

function  adv() {
local MEM4="Falta el GateWay para el Source Routing !!!!!!"
local MEM5="Te van a pillar !!!!!!! $USER jejejejeje "
echo ${rojo}$MEM4${cl}
echo ${re}${rojo}$MEM5${cl}  ; }

# Determinacion de SO

function web_so() {
NC="nc -w "
HTTPPORT="80"
GET="GET /"
ECHO="/bin/echo"
HEAD="HEAD / HTPP/1.0"
HTTPVARIABLE="Server:"
WEB_SERVER="$1"
LOG="salida.txt"
#CHECKHTTP=( echo $GET  ; sleep 5)| $NETCAT $VICTIMA  80
( echo $HEAD  ; echo ; echo ) | $NC  8 $WEB_SERVER $HTTPPORT | grep
$HTTPVARIABLE  | cut -d: -f2  1> $LOG
cat  $LOG
rm -f $LOG  ;}


# Seleccion de Opcion
```

```
case $# in 0 ) mem ;;
          1 ) mem2 $1 ;
              adv  ;
              tcp $1 ;;
          3 ) if [ "$1" != "r" ]; then
                mem2 $2
                udp_gateway $3 $2
              else {  mem2 $3
                   adv
                   tcp_rango $2 $3 ;}
              fi ;;
          2 ) if [ "$1" != "u" ] ; then
                    if [ "$1" != "so" ] ; then
                         {  mem2 $1
                         # adv
                         tcp_gateway $2 $1 ;}
                    else { mem_web $2
                          web_so $2 ;}
                    fi
               else  { export HOST=$2
                       mem2 $HOST
                       udp $2
                       exit 0 ;}
               fi ;
esac
```

<------- Fin del Hven-Scanner, cortar aquí --------->

kliber@hven.com.ve