TECNICAS DE PORT SCANNING Y USO DEL NMAP.

1- ¿Qué es el escaneo de puertos y para qué sirve?

El escaneo de puertos es una técnica usada por **hackers** y **administradores** (sin ánimo de hacer distinciones) para **auditar máquinas y redes** con el fin de saber que puertos están abiertos o cerrados, los servicios que son ofrecidos, **chequear la existencia de un firewall** así como verificaciones sobre el funcionamiento del mismo y algunas otras cosas. Ni que decir tiene que ésta es una de las técnicas más utilizadas a la hora de penetrar en un sistema y realizar un análisis preliminar del sistema... sin duda una de las mejores y más efectivas para llevar a cabo nuestras "**oscuras intenciones**".

De eso te hablaré aquí, de las diversas técnicas de escaneo, de cómo escanear una máquina o red y de cómo aprovechar los resultados obtenidos para atacar o proteger nuestro sistema, todo con un propósito puramente educativo ;-)

2- Algunos tipos de escaneo conocidos.

Antes de seguir con la explicación deberías tener algunas nociones básicas acerca del protocolo **TCP** y algunas otras cosas que ya se han explicado en la revista.

Si no estas...

Si no estás iniciado en el tema, lo que leerás a continuación quizás te amedrente un poco. Vamos a ver, este artículo servirá de plataforma para explicar en profundidad muchos temas en próximos números, todo aquello que ahora no entiendas será explicado, como hacemos siempre, pero necesitamos una referencia y este artículo cumplirá esa función.

Algunos tipos de escaneo explicados de forma breve, y que luego verás mejor con la práctica, son:

-TCP connect: esta es una técnica rápida y simple, pero tiene el inconveniente de que canta un poquito xD y se detecta fácilmente. Además por lo general si utilizas esta técnica tus conexiones serán logueadas y/o filtradas.

Se basa en intentar establecer una conexión con el puerto del host remoto mediante la llamada a **connect ()** si se establece dicha conexión, evidentemente, el puerto está abierto.

-TCP SYN: se trata de un escaneo en el que no se establece una conexión completa, enviamos **SYN** y en función de la respuesta obtenida por el host contestamos con **RST** (en caso de estar abierto) para resetear la conexión, es decir, abortar.
Puede darse el caso de que al enviar un paquete **TCP** con el **bit SYN** no se reciba respuesta lo que significa que el host está desconectado o se filtra la conexión a ese puerto.
A este tipo de escaneo se le conoce como **escaneo "medio abierto"** o **"SYN stealth"**.

-Stealth Scan (TCP FIN): se trata de enviar **FIN** y esperar la respuesta del host víctima de nuestro escaneo **FIN stealth ("sigiloso")**, si ignora los paquetes enviados entonces el puerto está abierto. Los sistemas de la compañía **Micro$oft** (entre otros) no son susceptibles a este tipo de escaneo aunque parezca mentira :P .

-Reverse Ident (TCP): realizamos un escaneo normal **TCP** pero miramos si el puerto **113** está abierto con el objetivo de saber quién es dueño de los servicios que corren en otros puertos de la máquina.

-Ping Scan: ...bastante explícito xD en todo caso se debe usar cuando tu intención sea saber que máquina(s) están despiertas, es posible bloquear los pings, pero luego (con la practica) veremos como saltarse esta "protección" en caso de encontrarnos con el inconveniente.

-Bounce Attack (vía ftp): se trata de aprovechar la conexión proxy ftp para escanear a través de un servidor **FTP**. Esto es así porque podemos utilizar el comando **PORT** indicando una dirección **IP** y un puerto, en este caso el objetivo de nuestro escaneo y solicitamos una transmisión de datos, si el puerto en cuestión está cerrado se generará un **error 425** ("Can´t get data conection" o algo muy similar). Es una buena idea deshabilitar la opción **ftp proxy** para evitar que terceros utilicen nuestro servidor para escanear y/o atacar otras redes.

-UDP Scan: este escaneo mostrará los puertos abiertos que utilizan el protocolo **UDP** (con sus inconvenientes), es bastante lento aunque irá mejor si escaneas una máquina que utilice la plataforma **Window$** gracias a la política de **M$** de hacer las cosas "**iguales pero diferentes**" y "**viva el monopolio**".

-ACK Scan: muchas veces nos encontramos con un **bonito firewal**l que impide el "correcto **flujo de los paquetes**" xDD desde nuestra máquina al **host víctima**, por eso y otros motivos nos interesa saber qué tipo de configuración tiene el cortafuegos, es decir si el tráfico ha sido filtrado o no.

-Window Scan: muy parecido al anterior, pero nos dice también que puertos están abiertos.

-Null Scan: se trata de otro método de escaneo stealth, en el que enviamos un curioso paquete sin banderas levantadas.

-Xmas Scan: lo realizamos enviando paquetes **TCP** anormalmente configurados y todos los **flags** (banderas) **SYN, ACK, PSH, RST, URG y FIN** levantados.

-Idle scan: se trata de una técnica de **escaneo stealth** muy potente y eficaz, con la que no tenemos necesidad de enviar ni un solo paquete con nuestra IP si no que se utilizan **host's zombis**, sería interesante comentar esta técnica detalladamente en su propio espacio por lo que no profundizaremos ahora.

Finalmente,

-RCP Scan: se trata de enviar el comando **NULL (SunRPC)** a los puertos **tcp** o **udp** que están abiertos y ver si son puertos **RPC** para saber qué programa y su versión está corriendo.

Estos son los tipos de escaneo fundamentales aunque no son los únicos.

3- Nuestros enemigos: "IDS".

Antes de proceder a ver de qué manera podemos aplicar los diferentes ataques existentes que acabamos de repasar brevemente me gustaría dejar clara una cosa a la hora de escanear puertos a diestro y siniestro: **DEBES SER CUIDADOSO Y NO DAR LA NOTA** (--ino j*d*s!) ten en cuenta que aparte de haber buenos administradores (pocos pero hay :P) monitoreando el tráfico y la actividad de sistema, los cortafuegos y los logs del sistema... existen los llamados: **IDS** (Intrusion Detection System, que no traduciré porque seguro que ya sabéis lo que significa xD) mediante estos sistemas es posible detectar un escaneo, activar las alarmas y llevar a cabo las acciones oportunas, es decir, que si un "**UIA**" (Usuario del Intesné

Aburrido) se dedica a escanear ciento y pico máquinas de una red de arriba a abajo en plan "destroyer" tiene todos los números de meterse en un buen lío ...aunque os pego aquí un texto sacado de la web de los *Men In Green* ;-)

<< Es de destacar que conductas tan frecuentes en esta Sociedad de la Información, como son el Spam o el simple Escaneo de puertos, difícilmente encuentran cabida entre los delitos tipificados en nuestro Código Penal, por lo que no son perseguibles por vía penal. >>

Ejemplos de aplicaciones "**IDS**" son: **Cyber Cop, CISCO NetRanger, TripWire, Snort y L.I.D.S** siendo estos tres últimos gratuitos :)

IDS ¿Qué? ...

IDS ¿Qué? ¿?¿? Si nunca has oído hablar de esto, ya sabes, ***www.google.com*** e investiga un poco por tu cuenta.

4- Visión práctica: usando el NMap.

NMap es una herramienta **LIBRE** para la auditoria de redes, disponible en varias plataformas aunque fue desarrollada inicialmente para entornos **Unix**, y para mí una de las mejores utilidades que existen actualmente para el propósito, mediante esta potente herramienta podemos por ejemplo:

-Determinar que host's están disponibles en una red.
-Determinar los puertos abiertos que tiene el sistema y que servicios ofrecen.
-Determinar que sistema operativo corre en el host objetivo.
-Determinar la configuración y uso de cortafuegos.
-Arreglar una tarde aburrida escaneando el ordenador de nuestra vecina ;)

¿De dónde me bajo el NMap?

Como acabo de decir se trata de una herramienta **LIBRE** y eso significa que dispones del código fuente y el programa compilado; lo tienes para diversas plataformas. Aunque **NMap** está pensado para su uso en la consola (línea de comando) dispone de un agradable **GUI** (Interfaz Gráfica de Usuario) que hace más fácil todavía su manejo, si piensas utilizar **NMap** bajo la plataforma **Window$** te recomiendo que sea un **Window$ NT/2K** aunque existe para **95/98/ME**, por supuesto las versiones para **Window$** no son tan rápidas ni estables como cabría esperar debido a que todavía se encuentran en constante desarrollo.

Te puedes bajar la versión **NMap 1.3.1** para **Window$** aquí:

ttp://download.insecure.org/nmap/dist/nmapwin_1.3.1.exe

INSTALACIÓN DEL NMAP:

Si has optado por descargarte **NMAP** junto con su **GUI ("NMapWin")** simplemente sigue estos pasos:

1-Bájate **NMapwin_1.3.1.exe** (instalador para **Window$**) mencionado justo arriba.

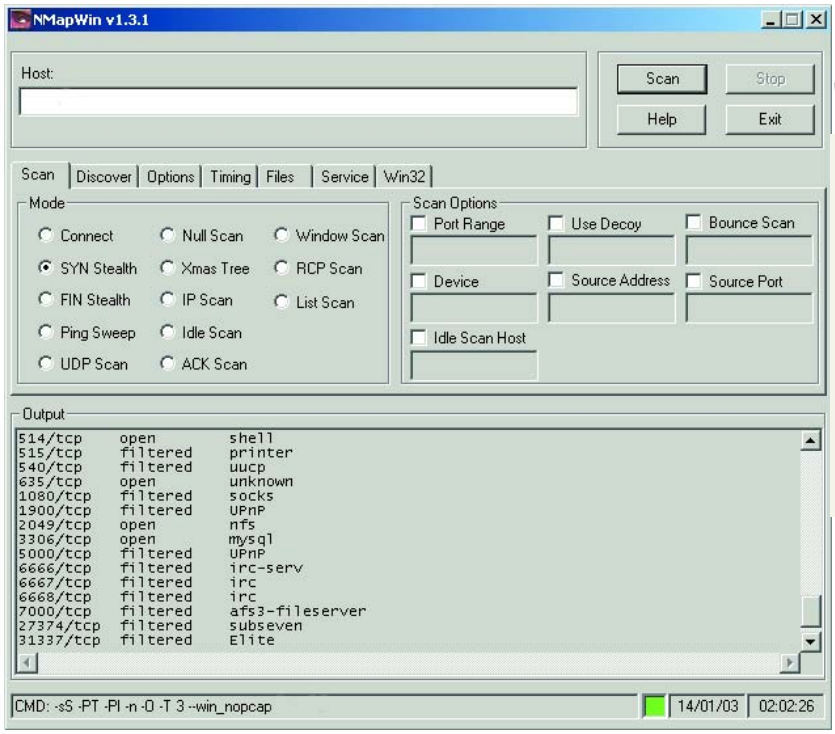
2-Ejecuta el instalador y sigue las instrucciones indicadas.

3-Una vez finalizado esto te vas al directorio que diste al programa y encontrarás nuevos ejecutables, debes instalar **WinPCap**.

4-Una vez reiniciado **Window$** (por una vez que sea decisión tuya ¿no? xD) ya dispones de **Nmap** para la línea de comandos y su interfaz gráfica plenamente funcionales :)

Ahora explicaremos cómo utilizar el **NMap** mediante su **GUI** para escanear los puertos de

un sistema mediante las diversas técnicas que he comentado.



Para empezar lo mejor será ver todas esas opciones que puedes utilizar jeje seguro que estás inquieto pensando en darle al botoncito de "scan" pero no te impacientes.

En primer lugar tenemos un cuadro de texto donde indicaremos el host que queremos escanear, podemos usar la IP o el nombre del host indicando además:

/32 para escanear únicamente el servidor especificado.
/16 para escanear una dirección de clase **'B'**.
/24 para escanear una dirección de clase **'C'**.

Si no entiendes ...

.Si no entiendes eso de Clase A, B, C, no te preocupes, en los próximos números y tomando de referencia este texto explicaremos las MUCHAS cosas que seguro no comprendes ahora mismo de este artículo… poco a poco y paso a paso :)

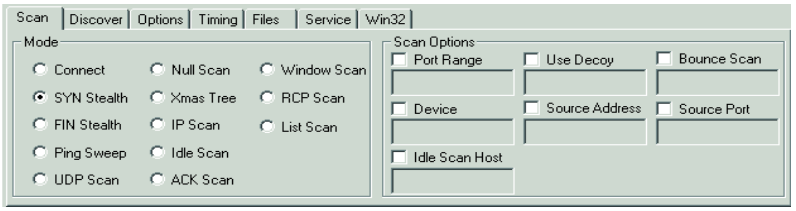
También se permite el uso del asterisco, por ejemplo: 198.154.3.* e incluso el escaneo de puertos dentro de un rango específico.

Si has leido ...

Si has leído los números anteriores de esta publicación ya sabes lo que es una IP y el formato que tiene, por lo tanto solo te puntualizo que cuando sustituyes una parte de la IP por un asterisco, lo que haces es escanear un rango de IPs.

Si tomamos como ejemplo 198.154.3., lo que hacemos es escanear desde la IP 198.154.3.0 a la IP 198.154.3.255*

Justo abajo tenemos una serie de pestañitas para indicar el tipo de escaneo a realizar así como las opciones adicionales que queremos aplicar. Los resultados obtenidos tras el escaneo aparecerán en el cuadro de salida (Output), además si te fijas en la barra de abajo aparecen los argumentos que deberíamos pasarle al **NMa**p mediante la línea de comandos.



Veamos esas opciones que es importante conocer, por ejemplo ya hemos comentado que es posible que nuestra víctima (por llamarla de alguna manera :P) esté on-line pero no deja que "pinguées" su servidor.

Si te fijas en la pestañita "**SCAN**" verás que puedes seleccionar las modalidades de escaneo que te he explicado hace un momento, y que no volveremos a comentar ahora, pero lo que si que es interesante explicar son las opciones de tu derecha ("**Scan Options**").

Como se puedes ver si quieres indicar un rango de puertos a escanear (con -P) debes marcar la opción **Port Range**, por ejemplo así: **10-2048**.
Otra opción curiosa que me gusta bastante (aunque yo no la uso :P) es **Decoy**, sirve para escanear usando uno o más señuelos para que el servidor objetivo vea que le atacan desde varias **IP**'s que debes indicar separadas por comas, si no indicas "**ME**" en una de las posiciones se te asignará una de oficio, digo, aleatoria xD imagínate que locura para un administrador ;)

Más cosas, en **Device** (-e) especificamos la interfaz desde la que enviar y recibir los paquetes, aunque normalmente **NMap** detecta esto de forma automática :) así que salvo casos puntuales no debes preocuparte por eso. Otra de las opciones de especial interés es indicar una **IP** en **Source Address** (-S) de manera que aparentemente el ataque se realiza desde esa IP, aunque evidentemente el escaneo no te servirá de nada :(especifica la interfaz en este caso (-e) y ten en cuenta que falsear la dirección de origen por otra ajena puede hacer que "esa IP" se meta en problemas, por favor, utiliza esta función con responsabilidad.

Otra opción es **Bounce Attack** (-b) necesita que le pases usuario:password@servidor:puerto y ya sabes lo que hace ;) pero recuerda que va muy lento.
También podemos especificar el **Source Port** (puerto de origen) con -g o marcando su casilla en la interfaz gráfica con el fin de establecer ese puerto como fuente de nuestro escaneo. Esto tiene sentido especialmente si un **firewall** hace excepciones en función del puerto de origen para filtrar o no los paquetes.

En la pestañita "**DISCOVER**" encontrarás nuevas posibilidades:

-No realizar un ping previo. (-P0)
-Realizar un "ping" TCP usando ACK tal que así: -PT80 (puerto 80 por no ser normalmente filtrado) si el objetivo no permite pings puedes estas dos últimas opciones... aunque es una sugerencia xD.
-Realizar un "ping" TCP usando SYN (-PS).
-Realizar un ping mediante ICMP.
Por defecto NMap realiza un ping ICMP + ACK en paralelo, es decir, -PI y -PT.

Si ahora te vas a "OPTIONS" podrás ver unas opciones muy interesantes, por ejemplo:

Fragmentation (-f) con esta opción se usan paquetes fragmentados que dificultan la detección del escaneo, en un futuro os hablaremos de ataques de ataques de fragmentación y relacionados ampliamente.

OS detection: pues eso, si quieres determinar que tipo de SO utiliza el objetivo marca esta casilla, recomendado.

Get Ident Info: usa la técnica del escaneo inverso (TCP reverse ident scanning) siempre que sea posible, úsalo al realizar un escaneo tipo -sT (connect), útil cuando quieras saber si un servicio corre con privilegios de administrador.

Fast Scan (-F): escanea los puertos especificados de /etc/services (nmap-services en nuestro caso xD).

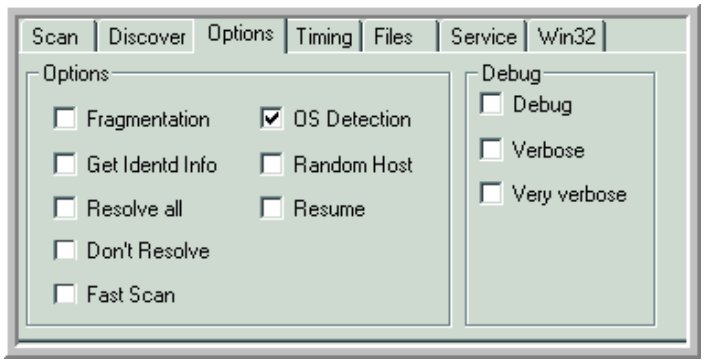
Random Host's: objetivos aleatorios (-iR) y bueno, realmente no nos interesa.

Con don´t resolve (-n) y resolve all (-R) indicamos si queremos resolución DNS, es decir, resolución de nombres de host por su IP.

Mediant Resume (--resume fichero) podemos retomar un escaneo detenido usando un fichero de salida.

Verbose y very verbose: nos da información más amplia de lo que sucede (-v) puedes usarlo dos veces (very verbose) para mostrar todavía

más información.



Si ahora pasas a la pestañita "timing" encontrarás opciones referentes al tiempo que, por lo general, no es necesario que modifiques aunque si por ejemplo te encuentras en una red con tráfico excesivo u otros casos en los que sea necesario reajustar estos valores debes hacerlo aquí, veamos lo que podemos hacer:

Throttle: sirve para configurar el tiempo de envío y respuesta de paquetes de forma general, siendo Paranoid el más lento e Insane el más rápido. (-T Paranoid|Sneaky|Polite|Normal...).

Otras alternativas disponibles (ino debes usarlas si modificas algo usando Throttle!)

Host Timeout: indica el tiempo máximo en milisegundos que se dedica al escaneo de un único sistema, por defecto no tiene impuesto un límite tal y como se puede ver.

Max RTT: tiempo máximo de espera en milisegundos que NMap esperará a recibir respuesta del sistema escaneado.

Min RTT (ms): para evitar casos de demora en la comunicación NMap nos asegura un tiempo mínimo de espera.

Initial RTT (ms): al especificar la opción -P0 para probar un cortafuegos podemos especificar un tiempo de espera en las pruebas iniciales.

Parallelism: (--max_parallelism num) indica el número máximo de escaneos que se permitirán ejecutar en paralelo, no debes utilizar un valor muy alto para no agotar los recursos de tu máquina ;-)

Scan Delay: tiempo mínimo de espera entre las diferentes pruebas en milisegundos, lo puedes usar para evitar (¿?) que algunos IDS detecten el escaneo.

Pasemos a la siguiente pestañita como ves hay MUCHAS opciones xD para "curiosear" redes; en Files encontrarás Input File y Output File (-iL y -oN respectivamente):

Input File: escanear los objetivos de un fichero dado con objetivos separados por retornos de carro (intro), espacios y tabulaciones.

Output File: guarda los resultados obtenidos en un fichero especificado, podemos especificar el formato del salida pero lo suyo es dejarlo normal (-oN) puedes marcar Append para añadir la nueva información al final del fichero.

A continuación se puede ver la pestaña "Service" que simplemente se usa para configurar el servicio pero para lo que estamos tratando ahora no es importante así que pasemos a otras opciones que aunque tampoco son especialmente necesarias veremos algunas, son las de la pestañita "Win32" y se usan para lo siguiente:

No Pcap (--win_nopcap): deshabilita el soporte Wincap.

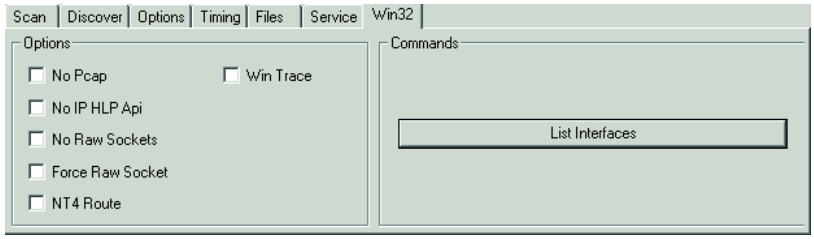
No Raw Sockets (--win_norawsock): deshabilita el soporte para raw sockets (algún día hablaremos de los sockets y los sockets en crudo puesto que proporcionan ventajas interesantes aunque por ahora puede ser demasiado avanzado).

List Interfaces (--win_list_interfaces): lista todas las interfaces de red.

C:\>nmap --win_list_interfaces
Available interfaces:

Name Raw send Raw recieve IP
loopback0 SOCK_RAW SOCK_RAW 127.0.0.1
eth0 SOCK_RAW winpcap 0.0.0.0
ppp0 SOCK_RAW SOCK_RAW 211.98.199.104

Force Raw Socket(--Win_forcerawsock): probar los raw sockets incluso en un sistema que no sea Window$ 2000.



Hemos visto lo podemos hacer y supongo que te das cuenta de la potente herramienta que tienes en tus manos ¿verdad? jeje pues espera a usarla ;).

4.1- Algunos ejemplos de uso.

Ahora veremos algunos ejemplos de escaneo utilizando el NMap, yo como soy un maniático usaré la línea de comandos, cada uno que use lo que más le guste :)

Antes de ponernos a "curiosear" permíteme recomendarte no escanear desde tu propio PC =:) búscate el cyber cutre de turno o asegúrate de ocultar bien tu IP, etc.
Realizamos un primer chequeo:

Hoy es un día aburrido y no hay nada mejor que hacer que buscar WinGates (servicio ofrecido normalmente en el puerto 1080), abrimos una ventana de comandos y escribimos: (**sustituye "objetivo" por la IP de la víctima**)

nmap -sP -vv -p 1080 **objetivo**

y obtendremos como respuesta algo parecido a esto:

Starting nmap V. 3.00 (www.insecure.org/nmap)
*Host (***.***.**.*) appears to be up ... good.*
*Initiating SYN Stealth Scan against (***.***.**.*)*
The SYN Stealth Scan took 0 seconds to scan 1 ports.
*Interesting ports on (***.***.**.*):*
Port State Service
1080/tcp filtered socks
[...]

Aunque lo mismo interesa buscar algunos servidores con NT:
nmap -sS -O -p 80 **objetivo***/24.*

Más ejemplos,

nmap -sS -PI -v ***

Starting nmap V. 3.00 (www.insecure.org/nmap)
*Host (***.***.***.***) appears to be down, skipping it.*
Note: Host seems down. If it is really up, but blocking our ping probes, try -P0

...mmm parece ser que con un escaneo (usando ping) normalillo no basta :P

nmap -sS -PT -v ***

Starting nmap V. 3.00 (www.insecure.org/nmap)
*Host (***.***.***.***) appears to be up ... good.*
*Initiating SYN Stealth Scan against (***.***.***.***)*
Adding open port 80/tcp
Adding open port 53/tcp
Adding open port 22/tcp
Adding open port 443/tcp
Adding open port 25/tcp
The SYN Stealth Scan took 31 seconds to scan 1601 ports.
*Interesting ports on (***.***.***.***):*
(The 1584 ports scanned but not shown below are in state: closed)
Port State Service
22/tcp open ssh

25/tcp open smtp
53/tcp open domain
79/tcp filtered finger
80/tcp open http
137/tcp filtered netbios-ns
138/tcp filtered netbios-dgm
139/tcp filtered netbios-ssn
143/tcp filtered imap2
389/tcp filtered ldap
443/tcp open https
901/tcp filtered samba-swat
3306/tcp filtered mysql
5432/tcp filtered postgres
6346/tcp filtered gnutella
6699/tcp filtered napster
10000/tcp filtered snet-sensor-mgmt

vaya... pues si que estaba despierto mira tu que cosas :P y resulta que tiene algunos puertos filtrados, ok. En cambio tiene otros como el 25 (servidor SMTP), el 53 (name-domain server) y 22/ssh (secure shell) abiertos.

Vamos a investigar eso un poco ;-) por simple curiosidad:

*C:\>telnet ***.*** 25*

*220 correo.***.*** ESMTP*
MAIL FROM yo mismamente!
250 ok
#en fin, no ha hecho falta el saludo xD. Seguimos.
[...]

y si nos conectamos al puerto 22 sacamos esto **SSH-2.0-OpenSSH_2.5.2p2 xD.**

Molaría saber el sistema operaivo: **nmap -sS -PT -PI -O -T 3** (usamos -O para detectar el SO)

Starting nmap V. 3.00 (www.insecure.org/nmap)
[...]
Remote operating system guess: Linux 2.2.12 - 2.2.19
[...]

nmap -sS -O -vv **NOMBRE/IP_víctima/24**

Con este escaneo barremos una red de clase C mediante un escaneo oculto (stealth SYN) y con la intención de averiguar el SO, además hemos especificado el modo very verbose para obtener más detalles acerca de lo que está ocurriendo.

Estos son algunos ejemplos simples de escaneo, ahora ya estás listo para salir "ahí fuera" a explorar de forma efectiva y segura (?) ya iremos viendo como explotar algunos agujeros de seguridad más a fondo en la revista para los cuales es buena idea realizar un análisis previo del sistema del que queremos comprometer la seguridad, no te pierdas los próximos artículos ;).

Para tus primeras...

Para tus primeras pruebas, te recomendamos utilizar como víctima para tus escaneos el servidor de Hack x Crack que está en la IP 80.36.230.235, que para eso está ;)