A1 – Exploraciones de red con Nmap y Nessus

Joaquín García Alfaro

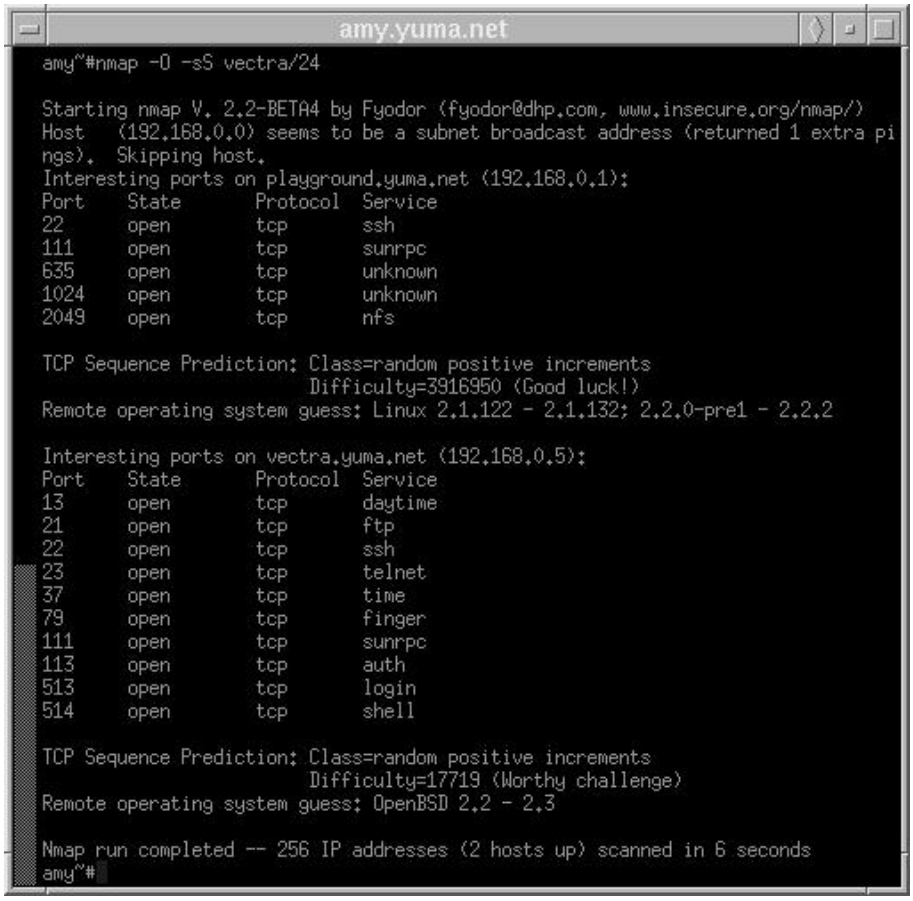
Índice

1.1. Nmap

La aplicación *Network Mapper*, más conocida como Nmap, es una de las herramientas más avanzadas para realizar exploración de puertos desde sistemas GNU/Linux. Nmap implementa la gran mayoría de técnicas conocidas para exploración de puertos y permite descubrir información de los servicios y sistemas encontrados, así como el reconocimiento de huellas identificativas de los sistemas escaneados. La siguiente imagen muestra un ejemplo de exploración de puertos mediante la herramienta *Nmap*:

A tener en cuenta

La mayor parte de herramientas de exploración de puertos pueden ser muy "ruidosas" y no son bien vistas por los administradores de red. Es altamente recomendable no utilizar estas herramientas sin el consentimiento explícito de los responsables de la red.



Aunque inicialmente Nmap fue pensada como una herramienta deshonesta para la realización de ataques, actualmente es una de las aplicaciones más utilizadas por administradores de red para la realización de comprobaciones de seguridad. Entre las muchas posibilidades que Nmap nos ofrece podríamos destacar las siguientes:

- **Auditorías de nuestra red.** Nmap permite una rápida visualización de puertos inseguros o abiertos por equivocación en los equipos de nuestra red.

- **Comprobación de la configuración de los elementos de seguridad.** Nmap puede ser de gran utilidad para comprobar la configuración de los elementos de seguridad de nuestra red como, por ejemplo, sistema cortafuegos, sistemas de detección de intrusos, etc. Por ejemplo, la realización de una exploración de puertos mediante Nmap desde el exterior de nuestra red podría asegurarnos que nuestro sistema cortafuegos está realizando el bloqueo de paquetes de forma correcta.

- **Comprobación de la configuración de los elementos de red.** Mediante Nmap podemos realizar una serie de comprobaciones de los dispositivos de conectividad y encaminamiento de nuestra red y detectar así si hay algún malfuncionamiento o, al contrario, si todo funciona con normalidad.

Desde el punto de vista de herramienta para la exploración de equipos de red, Nmap es más que un simple escáner de puertos. Algunas de las funcionalidades más interesantes que han hecho de Nmap una de de las herramientas de exploración más importantes y populares son las siguientes:

- **Alta velocidad de exploración.** Nmap ofrece una extraordinaria velocidad a la hora de realizar una comprobación de sistemas activos y servicios ofrecidos por dichos sistemas.

- **Descubrimiento de huellas identificativas.** Nmap ofrece la posibilidad de detectar la mayor parte de sistemas existentes en la actualidad con un alto grado de fiabilidad. Aunque Nmap no hace más que una predicción del sistema operativo que se esconde detrás del equipo que está explorando, dicha predicción se basa en contrastar la información recibida frente a una gran base de datos de respuestas basadas en tráfico IP, ICMP, UDP y TCP de centenares de sistemas operativos existentes en la actualidad.

- **Predicción de números de secuencia.** Todo el tráfico basado en protocolo TCP requiere un patrón aleatorio para establecer un inicio de conexión con el sistema remoto. Dicho patrón será establecido durante el protocolo de conexión de tres pasos de TCP mediante la utilización de números de secuencia aleatorios. En algunos sistemas operativos puede ser que estos números de secuencia no presenten un índice de aleatoriedad elevado, por lo que es posible realizar una predicción del número de secuencia que se utilizará en la siguiente conexión y realizar, por ejemplo, un secuestro de conexión TCP. Nmap ofrece la posibilidad de poder realizar una predicción de cómo se comporta la aleatoriedad de los números de secuencia del equipo al que está explorando.

- **Habilidad para imitar distintos aspectos de una conexión TCP.** El establecimiento de una conexión TCP requiere cierto periodo de tiempo (del orden de milisegundos). Muchos sistemas cortafuegos pueden estar configurados para descartar el primer paquete TCP de este establecimiento de sesión (con la bandera de TCP/SYN activada), y evitar que desde el exterior un atacante pueda establecer una conexión contra los equipos del sistema a proteger. La mayoría de los exploradores de puertos tradicionales utilizan este paquete de TCP/SYN para realizar este tipo de exploraciones, por lo que el sistema cortafuegos podrá bloquear dichos intentos de conexión y evitar que el explorador de puertos los detecte como activos. Nmap, sin embargo, es capaz de generar

paquetes TCP que atraviesen esta protección ofrecida por los sistemas cortafuegos y ofrecer una lista de los servicios activos en el equipo de destino.

- **Funciones para realizar suplantación.** Gracias a la capacidad de suplantación (*spoofing*) que ofrece Nmap, es posible que un atacante pueda hacerse pasar por otros equipos de confianza con el objetivo de atravesar la protección que ofrece el sistema cortafuegos de una red. Utilizado de forma correcta por parte de los administradores de dicha red, Nmap puede ayudar a modificar la configuración de estos sistemas cortafuegos disminuyendo, así, la posibilidad de recibir un ataque de suplantación.

- **Habilidad para controlar la velocidad de exploración.** La mayoría de los sistemas de detección de intrusos actuales suelen generar alertas en el momento en que detectan la presencia de una exploración secuencial de puertos contra la red que están protegiendo. De este modo, el sistema de detección podría avisar ante la presencia de diferentes exploraciones contra los equipos de dicha red. Mediante Nmap es posible modificar el comportamiento de la exploración para evitar así la detección por parte de sistemas de detección de intrusos tradicionales. De nuevo, una utilización correcta de Nmap por parte de los administradores de una red facilitará la adecuada configuración de los sistemas de detección para no dejar pasar desapercibidos este tipo de exploraciones.

- **Posibilidad de guardar y leer ficheros de texto.** Nmap ofrece la posibilidad de guardar sus resultados en forma de ficheros de texto de salida, de forma que otras aplicaciones puedan leer estos resultados, así como la posibilidad de leer la información de exploraciones anteriores por parte del propio Nmap.

1.1.1. Opciones de exploración de Nmap

Una de las posibilidades más interesantes de Nmap es su versatilidad a la hora de realizar sus exploraciones. Nmap puede ser utilizado tanto para una sencilla exploración rutinaria contra los equipos de nuestra red, como para realizar una compleja predicción de números de secuencia y descubrimiento de la huella identificativa de sistemas remotos protegidos por sistemas cortafuegos. Por otro lado, puede ser utilizado para la realización de una única exploración o de forma interactiva, para poder realizar multiples exploraciones desde un mismo equipo.

En general, como la mayoría de aplicaciones ejecutadas desde consola, Nmap puede combinar toda una serie de opciones de exploración que tienen sentido en conjunto, aunque también existen otras operaciones que son especificas para ciertos modos de exploración. A la hora de lanzar Nmap con múltiples opciones a la vez, la aplicación tratará de detectar y advertirnos sobre el uso de combinaciones de opciones incompatibles o no permitidas.

A continuación, mostramos algunas de las opciones de configuración de Nmap más básicas para la realización de exploraciones:

- **-P0** Por defecto, Nmap envía un mensaje ICMP de tipo `echo` a cada equipo que va a

explorar. Activando esta opción deshabilitaremos este comportamiento. Esta opción suele ser de gran utilidad si la exploración a realizar se realiza contra sistemas que aparentemente no han de estar activos y que, por tanto, no responderán a este primer mensaje ICMP enviado por Nmap. Por contra, si utilizamos esta información, deberemos ser conscientes que la información proporcionada por Nmap puede no ser del todo precisa.

- **-PT** Indica a Nmap que utilize paquetes TCP en lugar de mensajes ICMP para la realización del *ping* inicial contra el objetivo a explorar. Para ello, Nmap enviará un paquete del tipo TCP/ACK y esperará el envío de una respuesta TCP/RST por parte del equipo remoto. Esta opción es bastante interesante, ya que nos permite comprobar la existencia de un sistema cortafuegos entre el equipo donde estamos realizando la exploración y el equipo remoto. Aunque muchos cortafuegos filtran el tráfico ICMP de tipo `echo` para evitar la realización de `pings` desde el exterior, la mayoría suele permitir la entrada y salida de paquetes TCP/ACK y TCP/RST.

- **-v** Si activamos la opción *verbose* al lanzar Nmap, la aplicación nos irá mostrando las respuestas de los equipos explorados a medida que la vaya recibiendo. Si activamos dos veces la opción (`nmap -v -v`), recibiremos información adicional, dependiendo del tipo de exploración realizada.

- **-O** Esta opción indica a Nmap que trate de hacer una predicción del sistema operativo que se está ejecutando tras el equipo o los equipos que están siendo explorados. Esta es una opción muy apreciada por un posible atacante, ya que le permitirá iniciar una búsqueda de herramientas de explotación específicas, según el sistema operativo que se encuentre tras los equipos explorados.

  Como ya hemos adelantado anteriormente, Nmap utiliza una base de datos de huellas identificativas de la mayor parte de sistemas operativos existentes para apurar al máximo la precisión de sus resultados. Además, los desarrolladores de Nmap tratan de mantener dicha base de datos lo más actualizada posible para poder estar seguros de la eficiencia de predicción de Nmap. Gran parte de la información almacenada en esta base de datos se refiere a las peculiaridades de implementación de la pila TCP/IP de los distintos sistemas operativos existentes.

Opciones `-s` de Nmap

Todas aquellas opciones de Nmap precedidas por el prefijo `-s` se consideran opciones para realizar exploraciones silenciosas (pero detectables igual que cualquier exploración de puertos).

- **-sS** Utiliza una exploración de puertos TCP silencios basada en el envío de paquetes TCP/SYN. Si bien Nmap no finaliza el protocolo de conexión de forma expresa, Nmap es capaz de recoger la información suficiente para detectar todos aquellos servicios TCP ofrecidos por el equipo remoto.

- **-sP** Al activar esta opción, Nmap utilizará únicamente mensajes ICMP para la realización de la exploración.

1.1.2. Algunos ejemplos sencillos de exploración con Nmap

Como primer ejemplo de exploración con Nmap supondremos que, como administradores de nuestra red, queremos realizar únicamente una exploración mediante el uso de mensajes ICMP de tipo `echo`. El objetivo de dicha exploración podría ser la comprobación de los equipos de nuestra red local que están activos. La red local de nuestro ejemplo corresponde a la dirección IP 10.0.1.100/30. Para ello, lanzaremos Nmap con la opción `-sP` contra la dirección `10.0.1.100/30`. Si, adicionalmente, añadimos la opción `-v` para recibir la información de la exploración tan pronto llegue, éste sería el resultado obtenido de dicha exploración:

```
root$nmap -v -sP 10.0.1.100/30

Starting nmap 3.48 ( http://www.insecure.org/nmap/ ) at 2004-02-01 13:28 CET
Host 10.0.1.100 appears to be up.
Host 10.0.1.101 appears to be down.
Host 10.0.1.102 appears to be down.
Host 10.0.1.103 appears to be down.
Nmap run completed -- 4 IP addresses (1 host up) scanned in 1.046 seconds
```

A continuación, podríamos realizar una exploración de los servicios abiertos en el equipo que acabamos de ver activo y almacenar los resultados obtenidos, con la siguiente combinación de opciones de nmap (la opción `-oN` es utilizada para almacenar la información reportada por nmap en un fichero de *log*):

```
root$nmap -P0 -oN output.txt 10.0.1.100

Starting nmap 3.48 ( http://www.insecure.org/nmap/ ) at 2004-02-01 13:43 CET
Interesting ports on 10.0.1.100:
(The 1644 ports scanned but not shown below are in state: closed)
PORT     STATE SERVICE
7/tcp    open  echo
13/tcp   open  daytime
19/tcp   open  chargen
22/tcp   open  ssh
25/tcp   open  smtp
37/tcp   open  time
80/tcp   open  http
111/tcp  open  rpcbind
723/tcp  open  omfs
5801/tcp open  vnc-http-1
5901/tcp open  vnc-1
6000/tcp open  X11
6001/tcp open  X11:1

Nmap run completed -- 1 IP address (1 host up) scanned in 1.719 seconds
```

```
root$cat output.txt
# nmap 3.48 scan initiated Sun Feb  1 13:43:00 2004 as: nmap -P0 -oN output.txt
10.0.1.100
Interesting ports on 10.0.1.100:
(The 1644 ports scanned but not shown below are in state: closed)
PORT     STATE SERVICE
7/tcp    open  echo
13/tcp   open  daytime
19/tcp   open  chargen
22/tcp   open  ssh
25/tcp   open  smtp
37/tcp   open  time
80/tcp   open  http
111/tcp  open  rpcbind
723/tcp  open  omfs
5801/tcp open  vnc-http-1
5901/tcp open  vnc-1
6000/tcp open  X11
6001/tcp open  X11:1

# Nmap run completed at Sun Feb  1 13:43:02 2004 -- 1 IP address (1 host up)
scanned in 1.719 seconds
```

Si quisieramos ahora realizar una exploración selectiva contra el equipo 10.0.1.100, tratando de descubrir únicamente si están activos los servicios `ssh` y `web` de dicho equipo, podríamos realizar la exploración con las siguientes opciones de nmap (de nuevo, utilizando la opción `-oN` para almacenar la información reportada en un fichero de *log*):

```
root$nmap -sX -p 22,80 -oN output.txt 10.0.1.100

Starting nmap 3.48 ( http://www.insecure.org/nmap/ ) at 2004-02-01 13:55 CET
Interesting ports on 10.0.1.100:
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http

Nmap run completed -- 1 IP address (1 host up) scanned in 1.181 seconds
```

Al activar la opción `-sX`, Nmap realizará una exploración silenciosa contra el equipo `10.0.1.100` mediante la técnica de exploración *Xmas Tree**. No todos los sistemas operativos contestarán correctamente a una exploración de puertos utilizando la técnica de *Xmas Tree*, dado que algunos como, por ejemplo, OpenBSD, HP/UX, IRIX, Microsoft Windows, etc. no siguen el estándar propuesto por las RFCs del IETF.

Xmas Tree

Al igual que una exploración *TCP FIN*, la técnica de la exploración *TCP Xmas Tree* enviará un paquete `FIN`, `URG` y `PUSH` a un puerto, y esperará respuesta. Si se obtiene como resultado un paquete de reset, significa que el puerto está cerrado.

1.1.3. Utilización de Nmap en modo interactivo

Nmap ofrece un modo de trabajo interactivo que permite a los administradores de red la realización de multiples opciones de exploración desde una única sesión de consola. Para ello, tan sólo deberemos lanzar Nmap con la opción `--interactive` activada. A partir de ese momento, nos aparecerá como prefijo de consola la cadena `nmap>` desde la que podremos ir ejecutando las diferentes opciones de Nmap a nivel de comandos.

En la siguiente figura podemos ver una sesión de Nmap interactiva desde la cual hemos realizado una búsqueda de equipos activos en nuestra red local y, posteriormente, una exploración de servicios contra los equipos activos:

```
root$nmap --interactive

Starting nmap V. 3.48 ( http://www.insecure.org/nmap/ )
Welcome to Interactive Mode -- press h <enter> for help
nmap> n -v -sP 10.0.1.100/30
Host 10.0.1.100 appears to be up.
Host 10.0.1.101 appears to be down.
Host 10.0.1.102 appears to be down.
Host 10.0.1.103 appears to be down.
Nmap run completed -- 4 IP addresses (1 host up) scanned in 1.290 seconds
nmap> n -P0 10.0.1.100
Interesting ports on 10.0.1.100:
(The 1644 ports scanned but not shown below are in state: closed)
PORT     STATE SERVICE
7/tcp    open  echo
13/tcp   open  daytime
19/tcp   open  chargen
22/tcp   open  ssh
25/tcp   open  smtp
37/tcp   open  time
80/tcp   open  http
111/tcp  open  rpcbind
723/tcp  open  omfs
5801/tcp open  vnc-http-1
5901/tcp open  vnc-1
6000/tcp open  X11
6001/tcp open  X11:1

Nmap run completed -- 1 IP address (1 host up) scanned in 1.516 seconds
nmap> quit
Quitting by request.
root$
```

Si nos fijamos en el ejemplo anterior, los pasos seguidos son los siguientes:

1) Entramos en una nueva sesión interactiva de Nmap:

```
nmap --interactive
```

2) Utilizamos el comando `n` para realizar una nueva exploración de Nmap pasándole como argumentos las opciones necesarias para realizar una búsqueda de equipos activos en la red 10.0.1.100/30:

```
nmap> n -v -sP 10.0.1.100/30
```

3) Utilizamos de nuevo el comando `n` para realizar una exploración de puertos TCP abiertos en el equipo 10.0.1.100:

```
nmap> n -P0 10.0.1.100
```

4) Salimos de la sesión interactiva con Nmap mediante el comando `quit`:

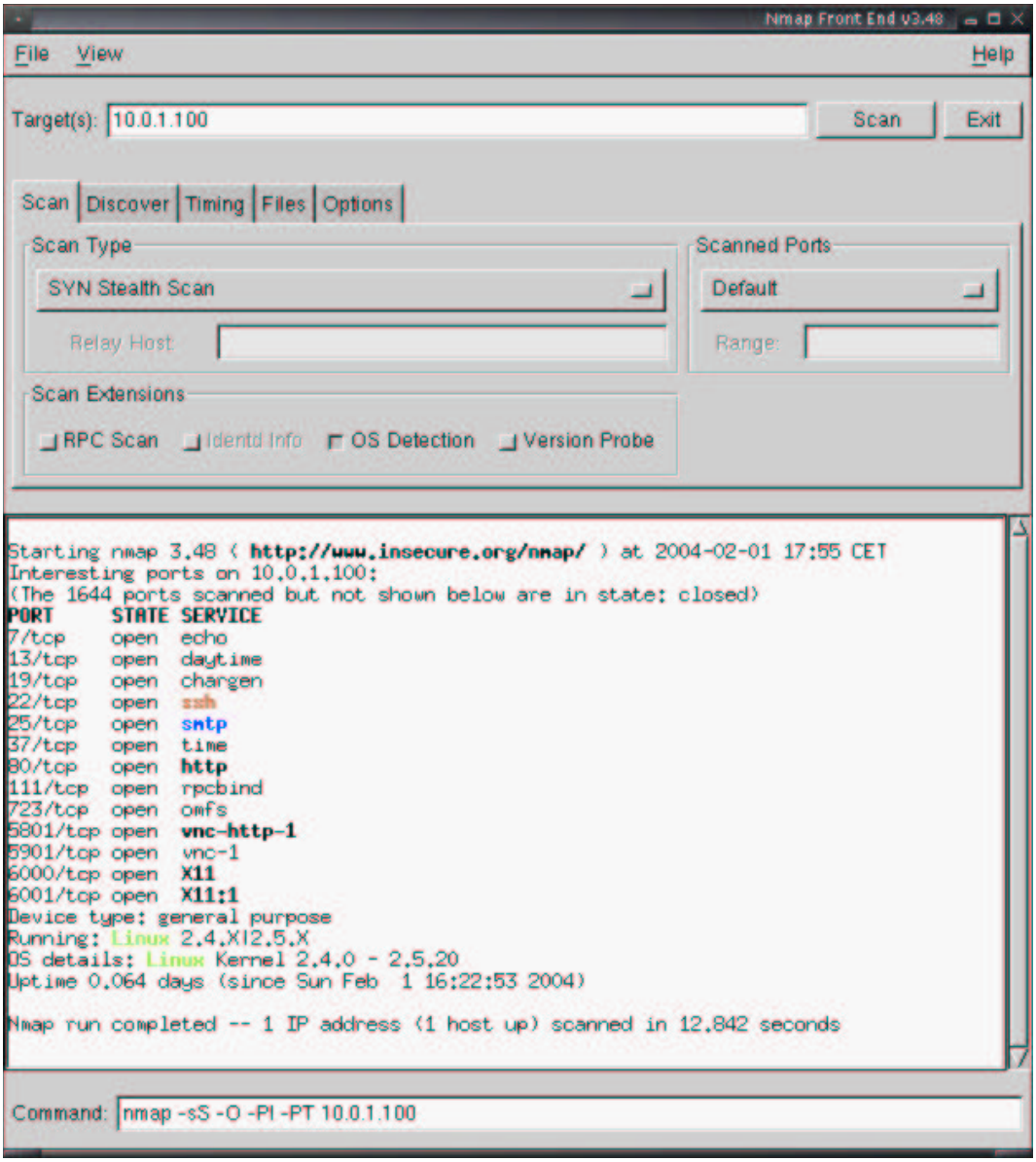
```
nmap> quit
```

Para ver los diferentes comandos que Nmap nos ofrece desde su modo interactivo, entraremos el comando `help` desde el guión de comando `nmap>`:

```
nmap> help
Nmap Interactive Commands:
n <nmap args> -- executes an nmap scan using the arguments given and
waits for nmap to finish.  Results are printed to the
screen (of course you can still use file output commands).
! <command>   -- runs shell command given in the foreground
x             -- Exit Nmap
f [--spoof <fakeargs>] [--nmap_path <path>] <nmap args>
-- Executes nmap in the background (results are NOT
printed to the screen).  You should generally specify a
file for results (with -oX, -oG, or -oN).  If you specify
fakeargs with --spoof, Nmap will try to make those
appear in ps listings.  If you wish to execute a special
version of Nmap, specify --nmap_path.
n -h          -- Obtain help with Nmap syntax
h             -- Prints this help screen.
Examples:
n -sS -O -v example.com/24
f --spoof "/usr/local/bin/pico -z hello.c" -sS -oN e.log example.com/24
```

1.1.4. Utilización de NmapFE como interfaz gráfica de Nmap

Por último, cabe destacar la posibilidad de trabajar de forma gráfica con Nmap a través del *front-end NmapFE*. Al igual que Nmap, y muchas otras herramientas relacionadas con Nmap, esta interfaz gráfica puede ser descargada del sitio web `www.insecure.org`. La siguiente figura muestra una sesión de exploración con Nmap desde esta interfaz gráfica.



1.2. Nessus

La herramienta Nmap que hemos visto en el apartado anterior es utilizada internamente por otras aplicaciones como, por ejemplo, escáners de vulnerabilidades, herramientas de detección de sistemas activos, servicios web que ofrecen exploración de puertos, etc.

Nmap, junto con **Nessus**, son dos de las herramientas más frecuentemente utilizadas tanto por administradores de redes, como por posibles atacantes, puesto que ofrecen la mayor parte de los datos necesarios para estudiar el comportamiento de un sistema o red que se quiere atacar.

Éste es el caso de la utilidad *Nessus*, una utilidad para comprobar si un sistema es vulnerable a un conjunto muy amplio de problemas de seguridad almacenados en su base de datos. Si encuentra alguna de estas debilidades en el sistema analizado, se encargará de informar sobre su existencia y sobre posibles soluciones.

Nessus es una herramienta basada en un modelo cliente-servidor que cuenta con su propio protocolo de comunicación. De forma similar a otros escáners de vulnerabilidades existentes, el trabajo correspondiente para explorar y probar ataques contra objetivos es realizado por el servidor de Nessus (`nessusd`), mientras que las tareas de control, generación de informes y presentación de los datos son gestionadas por el cliente (`nessus`).

Así pues, Nessus nos permitirá una exploración proactiva de los equipos de nuestra red en busca de aquellas deficiencias de seguridad relacionada con los servicios remotos que ofrecen dichos equipos. La mayor parte de las alertas que Nessus reportará estarán relacionadas con las siguientes deficiencias:

- Utilización de servidores (o *daemons*) no actualizados y que presentan deficiencias de seguridad conocidas como, por ejemplo, versiones antiguas de *sendmail*, *Finger*, wu-ftpd, etc.

- Deficiencias de seguridad relacionadas con una mala configuración de los servidores como, por ejemplo, permisos de escritura para usuarios anónimos por parte de un servidor de ftp.

- Deficiencias de seguridad relacionadas con la implementación de la pila TCP/IP del equipo remoto.

- Deficiencias de seguridad relacionadas con servidores de X Windows mal configurados o que presentan vulnerabilidades de seguridad.

- Utilización de aplicaciones CGI desde servidores web mal configuradas o mal programadas y que suponen una brecha de seguridad contra el sistema que las alberga.

- Instalación de puertas traseras, troyanos, demonios de DDoS u otros servicios extraños en sistemas de producción.

Como veremos más adelante, Nessus se compone de un conjunto de *plug-ins* que realizarán varias simulaciones de ataque. Alguno de estos ataques *simulados* pueden llegar a ser peligrosos contra el sistema analizado. Aunque Nessus no podrá nunca llegar a destruir información del sistema remoto, sus acciones podrían:

- Conducir a una denegación de servicio del sistema remoto. Al hacer las comprobaciones necesarias para realizar el test de análisis, ciertos *plug-ins* de Nessus pueden hacer reaccionar el sistema remoto de forma inadecuada y hacer que éste falle.

- Generar una cantidad masiva de tráfico basura en la red, pudiendo llegar a afectar al trabajo normal de los usuarios de la red.

Por estos motivos, es importante conocer correctamente las distintas posibilidades de configuración de esta herramienta y las distintas opciones que nos ofrece. Por otro lado, es conveniente realizar las pruebas de Nessus en horarios programados, con baja carga de trabajo en los equipos analizados. Es importante, por ejemplo, estar seguros de que los equipos que van a ser analizados pueden ser reiniciados, en caso de problemas, sin que esto afecte a ningún usuario legítimo de la red. También es importante tener presente que la red que va a ser analizada puede estar durante un breve periodo de tiempo saturada y que, por tanto, no estamos afectando a la producción normal de los servicios de dicha red.

1.2.1. Relación entre cliente y servidor de Nessus

Como ya hemos comentado anteriormente, para la elaboración de un escáner de vulnerabilidades, Nessus consta de dos aplicaciones. Por un lado, un servidor (`nessusd`) que será ejecutado en la máquina desde donde partirá el escaneo, y un cliente (`nessus`), que se comunicará a través de *sockets* al servidor. Generalmente, cliente y servidor se estarán ejecutando en distintas máquinas. Mientras que el cliente de nessus consta de una interfaz gráfica de usuario, el servidor de Nessus es una aplicación de consola que será ejecutada en modo *daemon*. En sistemas GNU/Linux, dependiendo de la distribución utilizada, el servidor de Nessus será ejecutado en modo *daemon* en el momento de iniciar el equipo por el guión de inicio del sistema correspondiente (generalmente situado en `/etc/rc.d/init.d/nessus`).

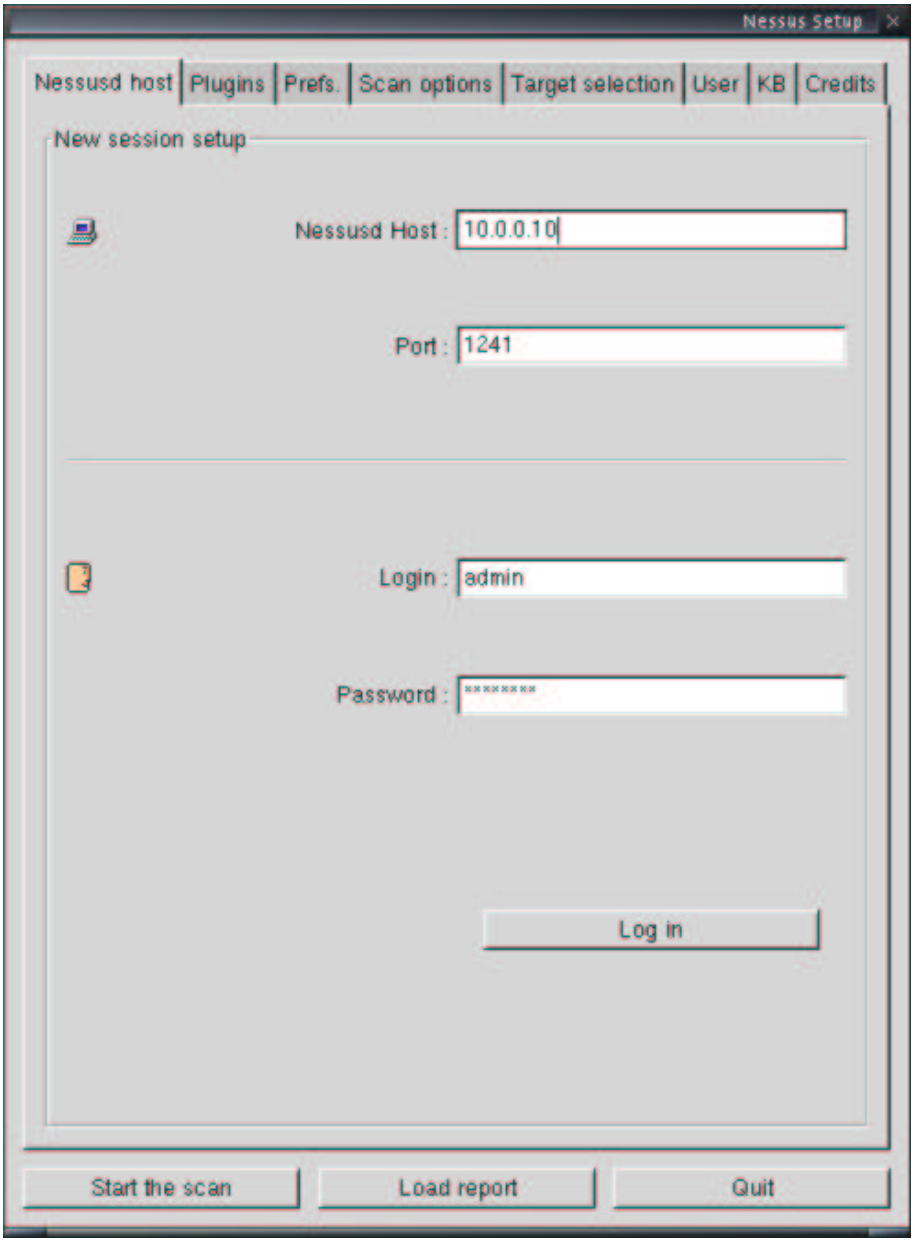
Clientes de Nessus

Existen distintos clientes de Nessus para conectarse a un servidor de Nessus, incluyendo clientes para sistemas Unix/Linux, Windows y Macintosh.

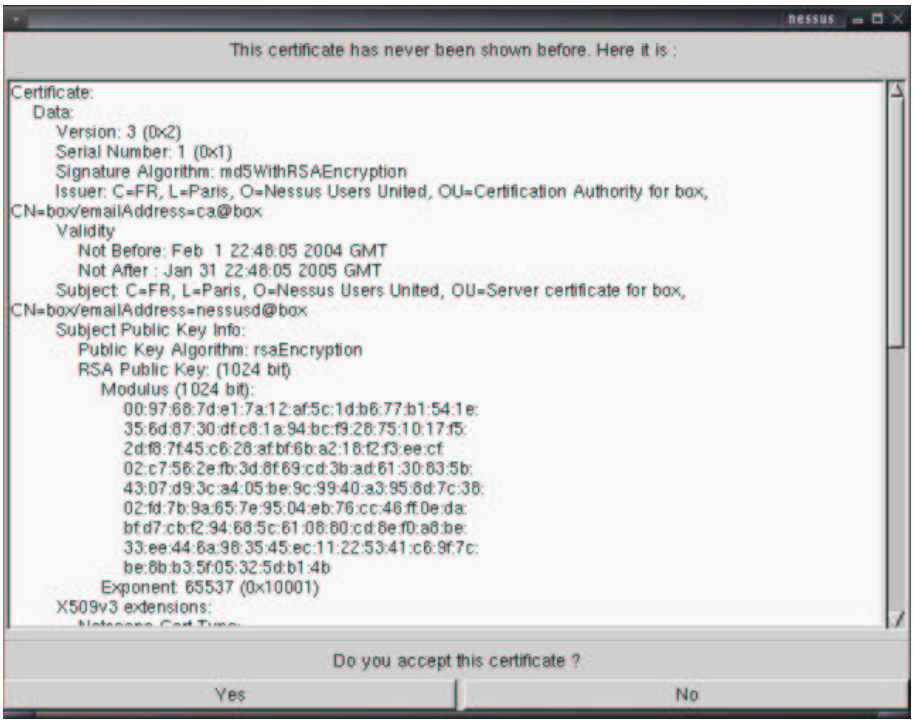
A la hora de conectarse al servidor, el cliente de Nessus realizará un proceso de autenticación. Este proceso de autenticación puede realizarse mediante el cifrado de los datos de autenticación o sin él. De las dos opciones anteriores, será preferible usar la versión cifrada, pues dificultará que un usuario ilegítimo pueda llegar a usar nuestro servidor de Nessus para realizar un análisis no autorizado.

En la siguiente figura podemos ver un cliente de Nessus ejecutado en un sistema GNU/Linux que ha utilizado el nombre de usuario `admin` para conectarse al servidor de Nessus que

está escuchando por el puerto `1241` del equipo con dirección IP `10.0.0.10`. Antes de poder realizar cualquier interacción con el servidor de Nessus que se está ejecutando en el equipo `10.0.0.10`, ha sido necesario realizar el proceso de autenticación entre ambos.



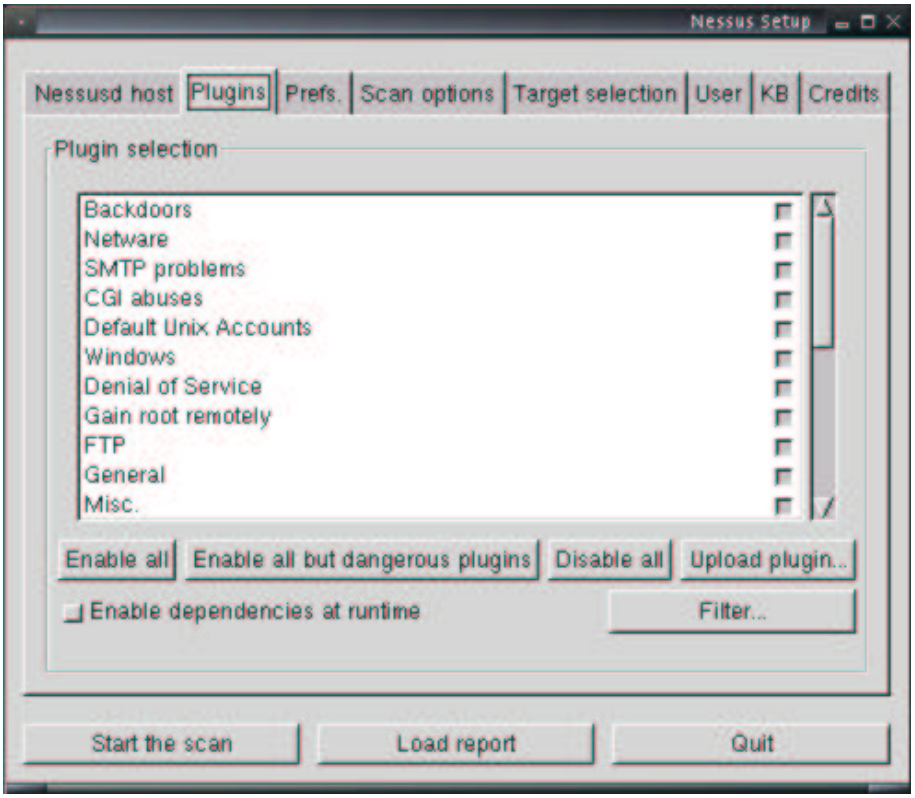
Si el mecanismo de autenticación escogido es mediante cifrado, el servidor de Nessus enviará un certificado digital con la información necesaria para poder realizar el proceso de autenticación correctamente. En la siguiente figura podemos ver un ejemplo de certificado enviado por el servidor de Nessus hacia el cliente.



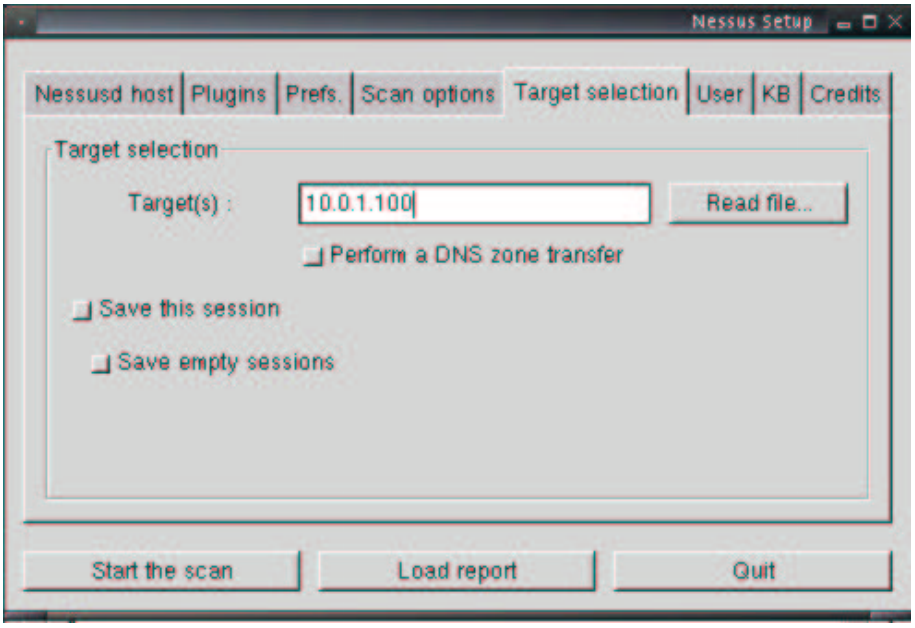
Es importante tener presente que, aparte del usuario remoto que utilizaremos para conectarnos al servidor de Nessus, existe también un usuario para poder utilizar el cliente de Nessus en la máquina local. Este usuario, utilizado para conectarse desde un equipo local al cliente de Nessus, será creado la primera vez que lancemos el cliente. Así, el cliente de Nessus solicitará un nombre de usuario y una contraseña local, para evitar que un usuario ilegítimo pueda utilizar el cliente de Nessus de forma no autorizada. Este nombre de usuario y su correspondiente contraseña no tienen nada que ver con el nombre de usuario y el mecanismo de autenticación que utilizaremos para la conexión remota contra el servidor de Nessus.

Una vez lanzado el cliente, será necesario proporcionar la dirección IP del equipo remoto donde se está ejecutando el servidor de Nessus, el puerto TCP por el que está escuchando, así como el nombre de usuario y la contraseña (o la clave correspondiente a la autenticación cifrada) asociada al servidor. Después de realizar el `Login`, la conexión con el servidor remoto de Nessus deberá iniciarse. Para ello, deberemos haber creado la correspondiente cuenta de usuario en el equipo remoto donde se encuentra el servidor. En caso contrario, el proceso de autenticación fallará.

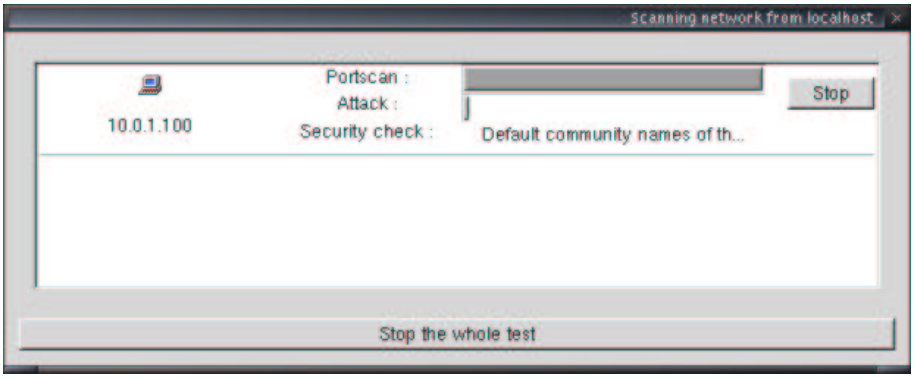
Si el proceso de autenticación es correcto y el cliente ha podido conectarse al servidor de Nessus, deberá mostrarse en la pantalla del cliente la lista de *plug-ins*. Desde esta pantalla podremos activar y desactivar los distintos *plug-ins* que el servidor utilizará durante el análisis. Si el proceso de autenticación no se ha realizado con éxito, el menu de *plug-ins* estará vacío. En la siguiente figura podemos ver la pantalla de *plug-ins* tras realizar un proceso de autenticación correcto.



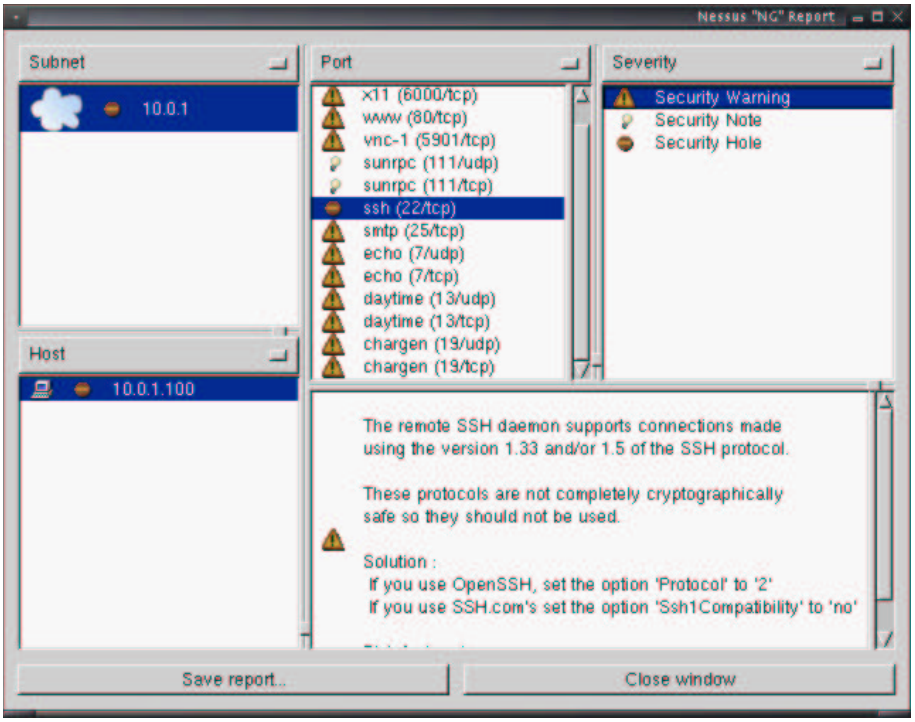
Una vez realizada la conexión con el servidor de Nessus, y habiendo seleccionado los *plug-ins* correspondientes a los análisis que deseamos realizar, podemos escoger el equipo (o una lista de equipos) a analizar. En la siguiente figura vemos, por ejemplo, cómo seleccionar desde el cliente de Nessus el equipo `10.0.1.100`. Hay que recordar que dicho equipo recibirá las diferentes pruebas desde la máquina donde se encuentra instalado el servidor de Nessus, no el cliente.



Tras seleccionar el objetivo, podemos iniciar el análisis mediante la opción correspondiente en la siguiente pantalla del cliente de Nessus. A medida que el análisis se vaya realizando, iremos viendo por pantalla las distintas operaciones que el servidor de Nessus va realizando contra el equipo o equipos seleccionados. La siguiente figura muestra el análisis de vulnerabilidades contra el equipo `10.0.1.100` seleccionado anteriormente.

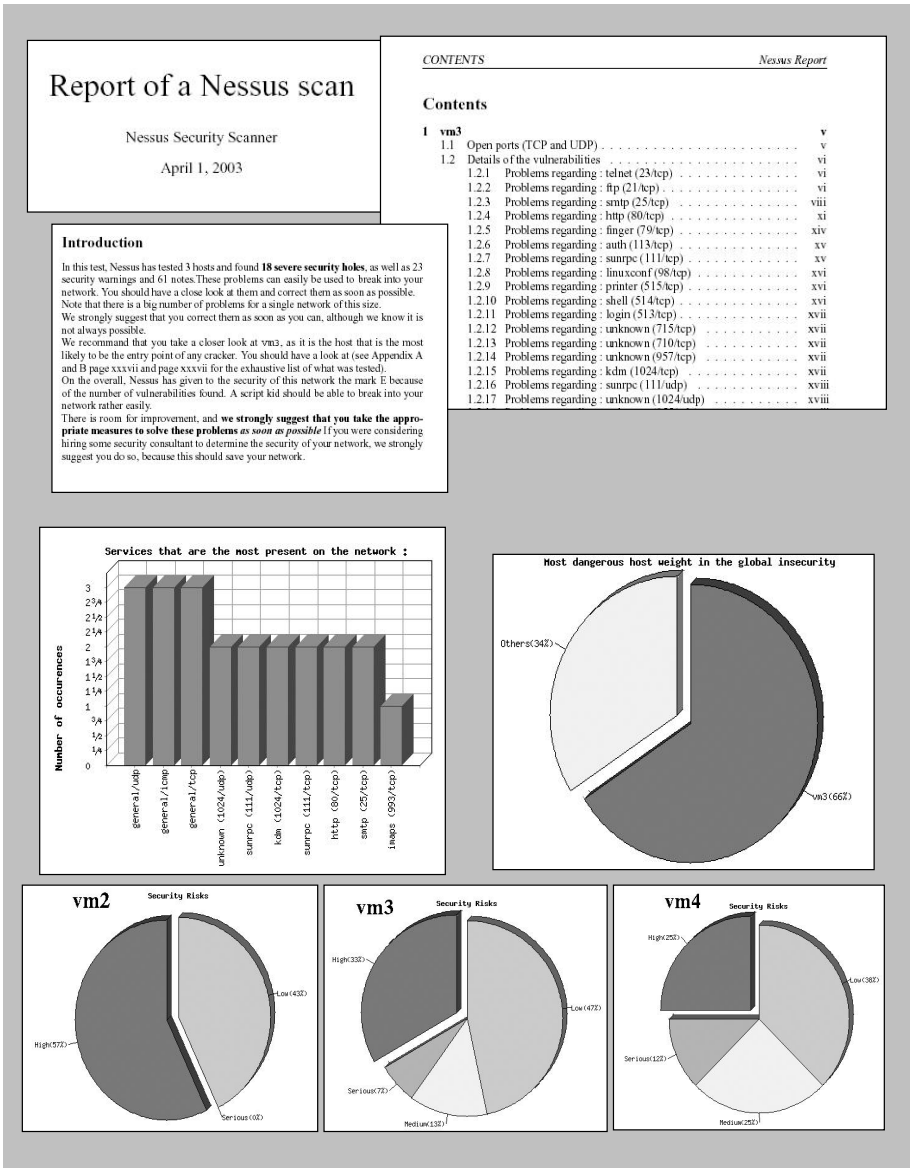


Una vez finalizada la exploración de vulnerabilidades por parte del servidor de Nessus, el cliente nos mostrará por pantalla los resultados de dicha exploración. Estos resultados dependerán de los *plug-ins* que hayamos activado, de las diferentes opciones de configuración del servidor, etc. Como vemos en la siguiente figura, la interfaz ofrecida por el cliente para mostrar los resultados permitirá su visualización por equipos, por redes, según la severidad de las vulnerabilidades detectadas, etc.



La aplicación también permitirá almacenar los resultados obtenidos durante la realización de la exploración en forma de informes. Las actuales más recientes del cliente de Nessus

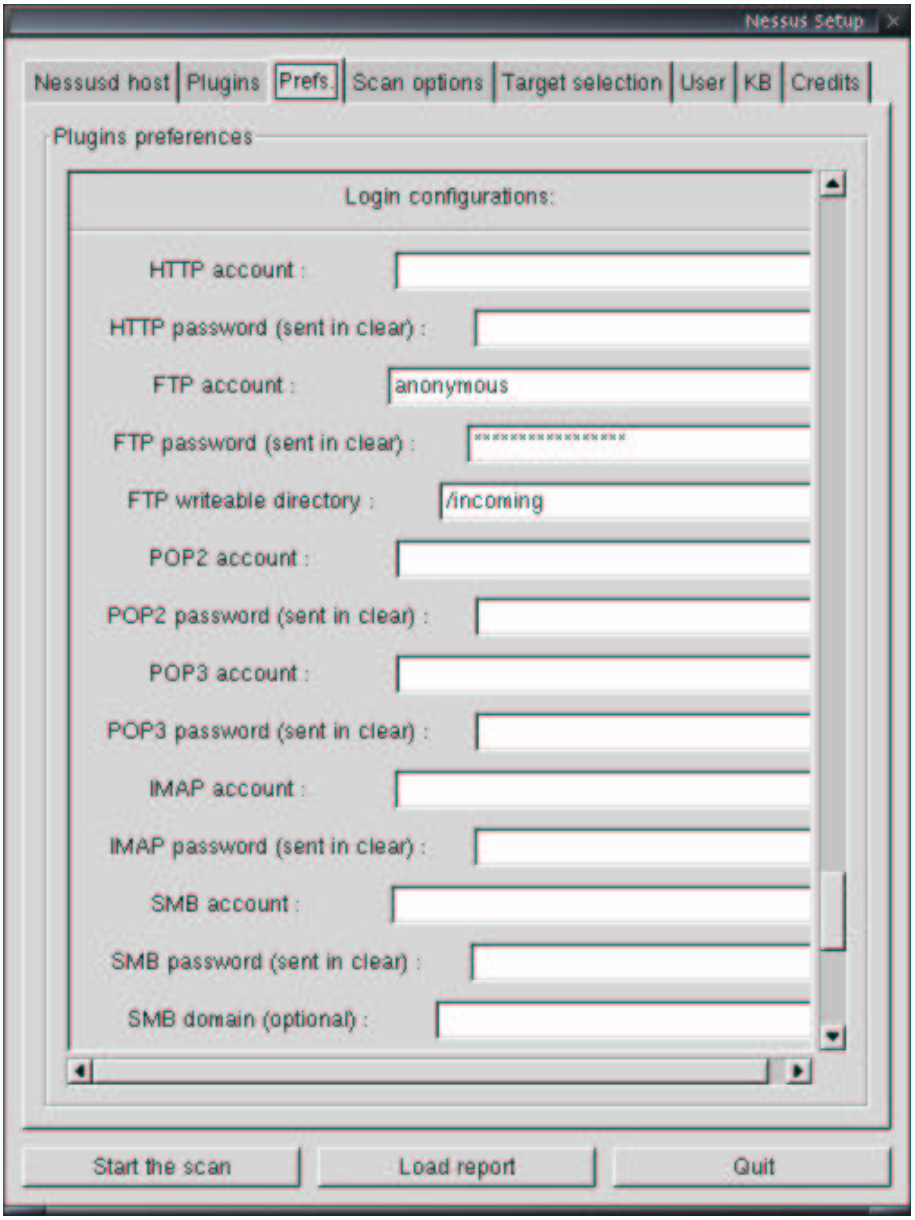
ofrecen un amplio rango de formatos para poder almacenar tales informes. Desde guardar el informe en un formato propio de la aplicación (para poder ser visualizado de nuevo desde otro cliente de Nessus) hasta la realización de completos informes en HTML, XML, LaTeX, etc. En la siguiente figura, podemos ver un ejemplo de informe generado mediante el cliente de Nessus en HTML y PDF.



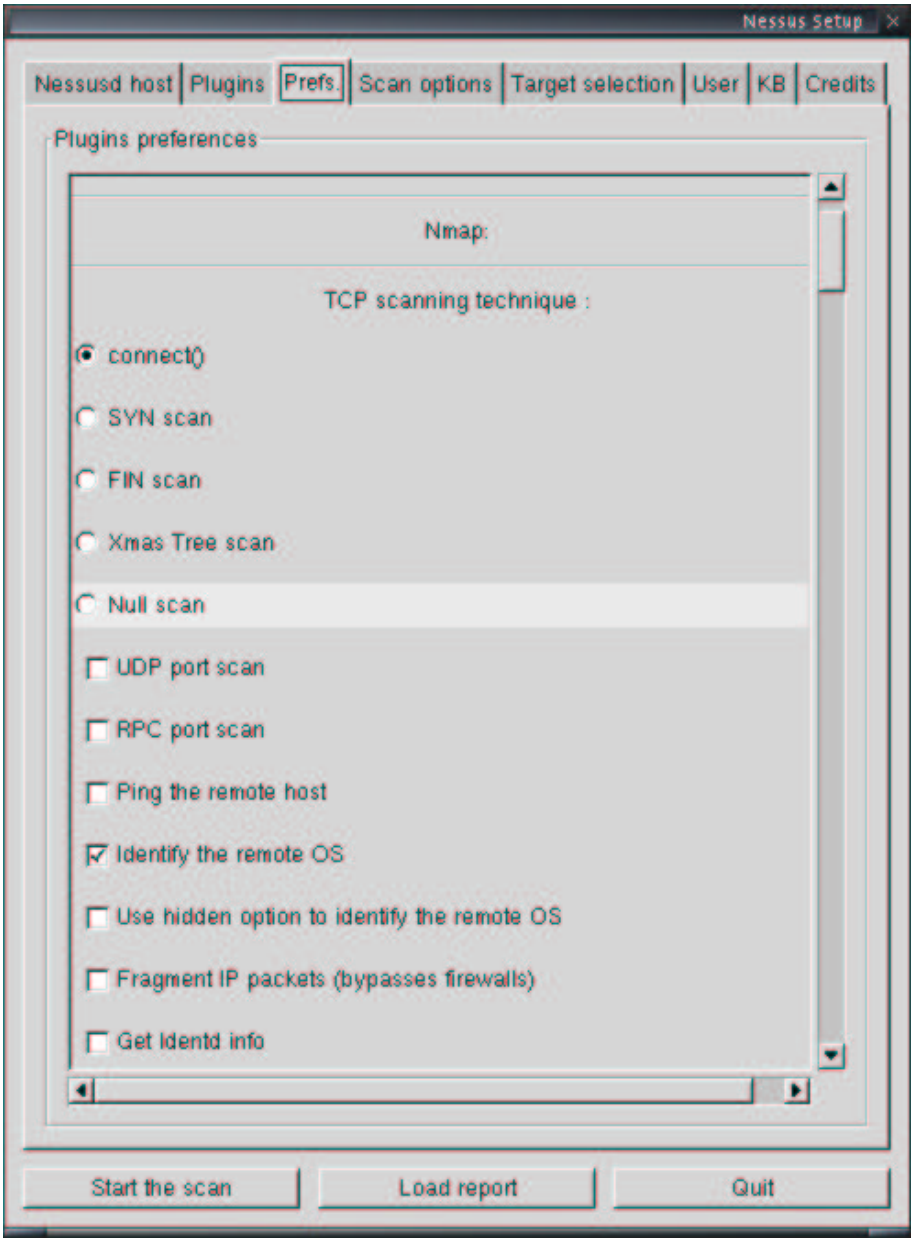
1.2.1.1. Configuración de *plug-ins*

Aunque la mayoría de los *plug-ins* de Nessus pueden ser utilizados sin necesidad de ajustes, otros *plug-ins* requerirán la inserción de información adicional para funcionar correctamente. Por ejemplo, los análisis relacionados con *plug-ins* de `ftp`, `smtp` y otros servicios similares con autenticación de usuarios requerirán la inserción de información relacionada con autenticación de usuarios. También es posible, por ejemplo, configurar el *plug-in* de `ftp` para que trate de almacenar información en el caso de encontrar una mala configuración en los permisos de escritura de los recursos asociados. Otra posibilidad es la

modificación de los parámetros de Nmap (relacionado con la exploración de puertos que el servidor de Nessus realizará). En la siguiente figura podemos ver la interfaz ofrecida por el cliente de Nessus para realizar algunas de las modificaciones comentadas.



La mayor parte de estas opciones estarán disponibles desde el cliente de Nessus en el momento de realizar la conexión con el correspondiente servidor. Aun así, una vez realizadas las modificaciones, éstas pueden ser almacenadas localmente para tratar de aplicarlas a servidores de Nessus adicionales. Así, si modificamos, por ejemplo, las preferencias de la exploración de puertos para que Nmap utilize UDP como protocolo, en lugar de TCP, podemos tratar de almacenar estas preferencias para que se apliquen de nuevo en futuras conexiones al mismo o a distintos servidores de Nessus. En la siguiente figura podemos ver la interfaz que ofrece el cliente de Nessus en cuanto a las opciones de exploración relacionadas con Nmap.



Cabe destacar que la actualización de *plug-ins* puede ser automatizada a través de guiones de sistema como, por ejemplo, los guiones del servicio `cron` de sistemas Unix. A medida que van apareciendo nuevas vulnerabilidades o problemas de seguridad en general, la comunidad de desarrollo de Nessus, así como administradores de red u otros profesionales en seguridad de redes, suelen traducir tales vulnerabilidades en forma de nuevos *plug-ins* de Nessus para ser descargados por parte de los usuarios de la aplicación. Aunque existen distintas formas de realizar el proceso de actualización, la más intuitiva consiste en una simple descarga desde el sitio web de Nessus con todo el paquete de *plug-ins*, y reemplazar los anteriores.

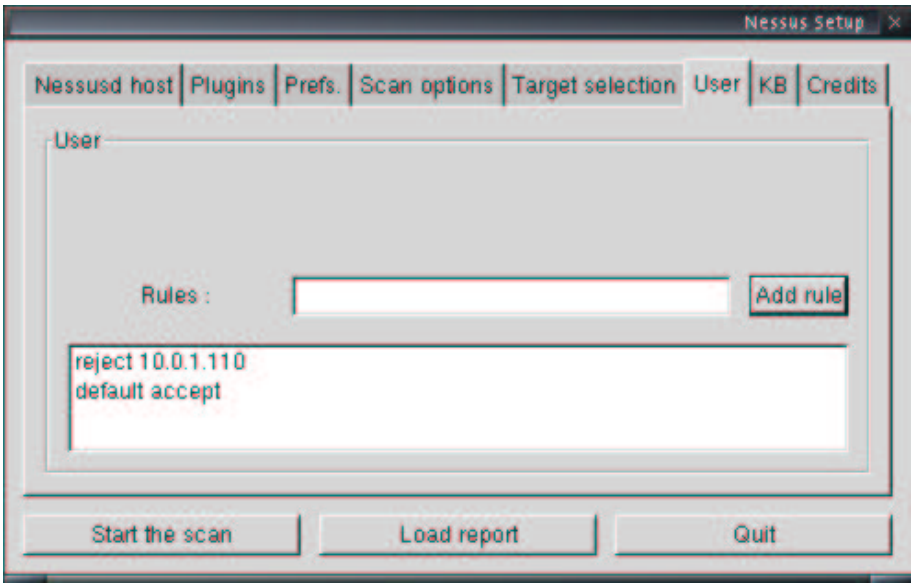
1.2.2. Creación de usuarios

Como ya hemos comentado anteriormente, para poder contactar con el servidor de Nessus desde el cliente, es necesario la utilización de usuarios. Mediante la utilización de la aplicación `nessus-adduser` será posible la definición de nuevos usarios desde sistemas GNU/Linux. Esta utilidad nos permitirá crear tanto usuarios locales, que tan sólo podrán acceder a Nessus desde el equipo local, como usuarios remotos, que podrán acceder a Nessus desde máquinas remotas.

Por otro lado, Nessus también ofrece la posibilidad de configurar distintos parámetros asociados con dichos usuarios: listado de máquinas desde las que se podrán conectar los usarios remotos, listado de *plug-ins* que se les permitirá ejecutar, listado de máquinas que el usuario podrá analizar, etc. La posibilidad de poder modificar estas conductas de exploración responden a la posibilidad de tener diferentes perfiles de administradores, responsables de la realización de distintos tipos de exploración según sus privilegios en el sistema.

Aunque para la creación tanto de usuarios locales como de usuarios remotos será posible la utilización del comando `nessus-adduser`, la configuración de las conductas de exploración u otras características más específicas deberán ser especificadas de distintas maneraras. Algunas, como la lista de máquinas desde las que se podrá conectar el usuario, deberán ser especificadas de forma manual en los ficheros de configuración correspondientes a cada usuario (generalmente, en `/etc/nessus/`).

A través del cliente de Nessus también será posible realizar un control personal para su utilización como, por ejemplo, limitar el número de máquinas que podrán ser analizadas por el usuario. La siguiente figura muestra cómo limitar el cliente para que pueda hacer una exploración a cualquier máquina excepto al equipo `10.0.1.110`.



Resumen

En este capítulo hemos visto cómo utilizar dos poderosas herramientas basadas en código abierto para la realización de exploración de puertos y exploración de vulnerabilidades en red. Mediante la primera de estas dos herramientas, Nmap, es posible descubrir qué puertos TCP o UDP están ofreciendo servicios en equipos explorados. Pero Nmap no es tan sólo una herramienta para descubrir servicios abiertos, Nmap ofrece muchas otras características como, por ejemplo, descubrir el sistema operativo albergado por dichos equipos, las características de implementación de la pila TCP/IP de tales sistemas operativos, la posibilidad de realizar predicción de números de secuencia TCP, realizar comprobaciones contra *routers* o *firewalls* intermedios, etc. Nmap es una herramienta muy importante a conocer, ya que es utilizada por la mayor parte de la comunidad de administradores de *software* libre. Además, es utilizada por terceras aplicaciones como, por ejemplo, Nessus.

La segunda herramienta estudiada, Nessus, es un potente escáner de vulnerabilidades basado en código libre, que proporcionará al administrador la posibilidad de realizar complejos análisis de red para detectar vulnerabilidades de seguridad en equipos remotos. Basado en una arquitectura cliente-servidor, Nessus ofrecerá la posibilidad de realizar exploraciones proactivas en busca de servidores antiguos o mal configurados, deficiencias de seguridad en la implementación TCP/IP de equipos en producción, instalación de servicios ilegítimos o sospechosos en los equipos de nuestra red, etc.

Por un lado, el cliente de Nessus hará de interfaz intermedia entre el administrador de la red y la aplicación que realizará los distintos chequeos de seguridad (servidor de Nessus). De este modo, podremos ejecutar las exploraciones de vulnerabilidades desde el exterior de la red, utilizando distintos sistemas operativos donde recoger los resultados y construir los informes con la información reportada por el servidor de Nessus. Por otro lado, será posible la instalación de diferentes instancias del servidor de Nessus en distintas localizaciones de la red, para realizar las exploraciones de seguridad con distintas vistas al sistema.

Nessus ofrece también un conjunto de posibilidades de autenticación de usuarios, para garantizar que usuarios no autorizados no utilicen los recursos de la red para realizar exploraciones de puertos o de vulnerabilidades de forma ilegítima. A través del uso de contraseñas de usuario, o bien mediante el uso de técnicas criptográficas, el cliente Nessus realizará un proceso de autenticación contrá el servidor de Nessus para garantizar que el usuario que se conecta a dicho servicio es un usuario legítimo.

Finalmente, Nessus ofrece la posibilidad de almacenar los resultados de la exploración realizada. Un amplio rango de formatos es ofrecido por el cliente de Nessus para almacenar los informes entregados por el sevidor de Nessus. Desde un formato propio de aplicación, hasta la utilización de formatos como ASCII, HTML, XML, LaTeX, etc.

Bibliografía

[1] Eyler, P. (2001). *Networking Linux: A Practical Guide to TCP/IP.* New Riders Publishing.

[2] Stanger, J.; Lane, P. T.; Danielyan E. (2001). *Hack Proofing Linux: A Guide to Open Source Security.* Syngress Publishing, Inc.

[3] Toxen, B. (2000). *Real World Linux Security: Intrusion Prevention, Detection, and Recovery.* Prentice Hall PTR.