Parte IV

Valoración final

Capítulo 17

Pruebas del sistema

17.1. Nessus: Escáner de vulnerabilidades

Nessus es un potente escáner de redes de Software Libre. Consta de dos partes (cliente/servidor) que pueden estar instaladas en las misma máquina por simplicidad.

Si el ataque se hace hacia *localhost* lo que se consigue es auditar nuestra propia máquina.

Cuando finaliza el escaneo se generan unos informes que si se sabe aprovechar e interpretar explican que tipo de vulnerabilidades han sido encontradas, cómo "explotarlas" y cómo "evitarlas".

La distribución de Nessus consta de cuatro ficheros básicos: las librerías del programa, las librerías NASL (Nessus Attack Scripting Language), el núcleo de la aplicación y sus plugins.

Para instalar el programa hay que realizar el apt siguiente:
`#apt-get install nessus nessusd`

Y después descargamos de la página de Nessus (*http://www.nessus.org/*) los plugins que creamos necesarios, instalandolos en la carpeta */nessus/plugins*.

Durante el proceso de instalación se nos realizan una serie de preguntas para generar un certificado SSL para Nessus:

```
-------------------------------------------------------------------------------
                        Creation of the Nessus SSL Certificate
-------------------------------------------------------------------------------

Congratulations. Your server certificate was properly created.

/etc/nessus/nessusd.conf updated

The following files were created :

. Certification authority :
   Certificate = /var/lib/nessus/CA/cacert.pem
   Private key = /var/lib/nessus/private/CA/cakey.pem

. Nessus Server :
    Certificate = /var/lib/nessus/CA/servercert.pem
    Private key = /var/lib/nessus/private/CA/serverkey.pem
```

Donde podemos observar que se ha creado un certificado autofirmado para el servidor Nessus.

17.1.1. Configurar el programa

Vamos a seguir una serie de pasos para configurar el servidor Nessus:

- El archivo de configuración es: `/etc/nessus/nessusd.conf`

  La configuración por defecto es completa y válida, entre otras cosas escanea desde el puerto 0 al 15000. Ahí podemos modificar todas las opciones que nos parezcan oportunas.

- Creamos un usuario para lanzar el programa, para ello seguiremos las instrucciones de pantalla, una vez ejecutado el siguiente comando:

  `#nessus-adduser`

  Entre las opciones que se presentan, para validar el usuario se puede elegir entre el sistema de login/password o certificados digitales, es mucho más seguro el sistema de certificados. También podemos especificar unas reglas (rules) concretas para ese usuario.

- Se puede elegir que el usuario acceda desde una red concreta, o permitir que acceda desde todas las redes. Para esta última configuración es neceario editar el archivo de reglas */etc/nessus/nessusd.rules* e introducir la siguiente línea en la parte *address/netmask*: `default accept`

  Podemos observar como queda el archivo así:

```
#
# Nessus rules
#

# Syntax : accept|reject address/netmask

# Accept to test anything :
default accept
```

- Registrar la versión de Nessus, para ello hay que entrar en: *http://www.nessus.org/register/* poner un correo electrónico donde se nos reportará la clave. Una vez recibido el correo basta con insertarlo en la línea de comandos:

  `#nessus-fetch --register <CLAVE>`

- Una vez tengamos el servidor registrado, lo arrancamos en background:

  `#nessusd -D` o `#nessusd --background`

  Si ejecutamos: `# ps aux | grep 'nessus'`,... podremos observar si se ha cargado bien:

```
root     10938  0.0  0.9  7332 4712 ?        Ss   13:22   0:00 nessusd: waiting
root     10960  1.0  0.1  2316  772 pts/3    S+   13:23   0:00 grep nessus
```

17.1.2. Ejecución de Nessus

Para lanzar el cliente en modo gráfico, ejecutaremos el siguiente comando: `#nessus`

También se puede iniciar en modo comando, para el modo comando consultaremos el manual de ayuda: `$man nessus`

Es necesario recordar que si tenemos activo el IDS *Snort* o cualquier otro NIDS, se va a volver loco al ejecutar *Nessus*. Es necesario desactivarlo o hacer que el NIDS ignore la IP de Nessus.

Para ejecutarlo podemos seleccionar las siguientes opciones:

- En *Nessusd Host*, indicaremos los datos añadidos en la creación del usuario (login/password, etc).
- En la pestaña de *plugins* (los tipos de ataques), seleccionaremos todos si queremos un escaneo completo. Hay que tener cuidado con *Denial of service* (DoS) por razones obvias.
- En *credentials*, colocaremos la cuenta, contraseña y dominio Samba, si es que lo queremos escanear.
- En *Scan Options*, especificaremos los puertos a escanear y la herramienta a utilizar.
- En *target*, indicaremos la o las direcciónes IP de las máquinas a escanear.

En las siguientes pantallas podemos observar cada una de estas secciones:

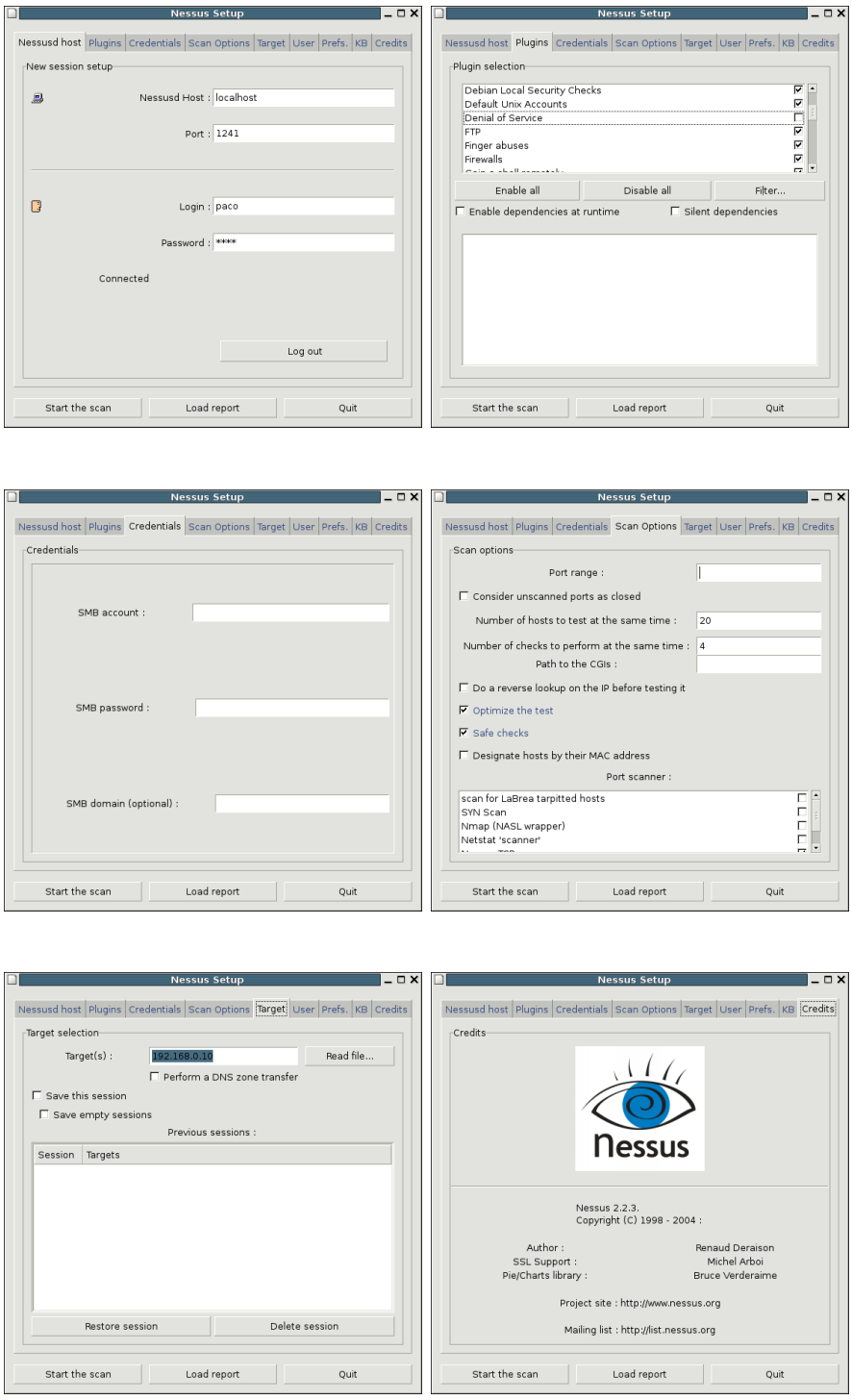

Figura 17.1: Configuración de la aplicación Nessus

Una vez realizada la comprobación del servidor podemos observar las siguientes vulnerabilidades encontradas:

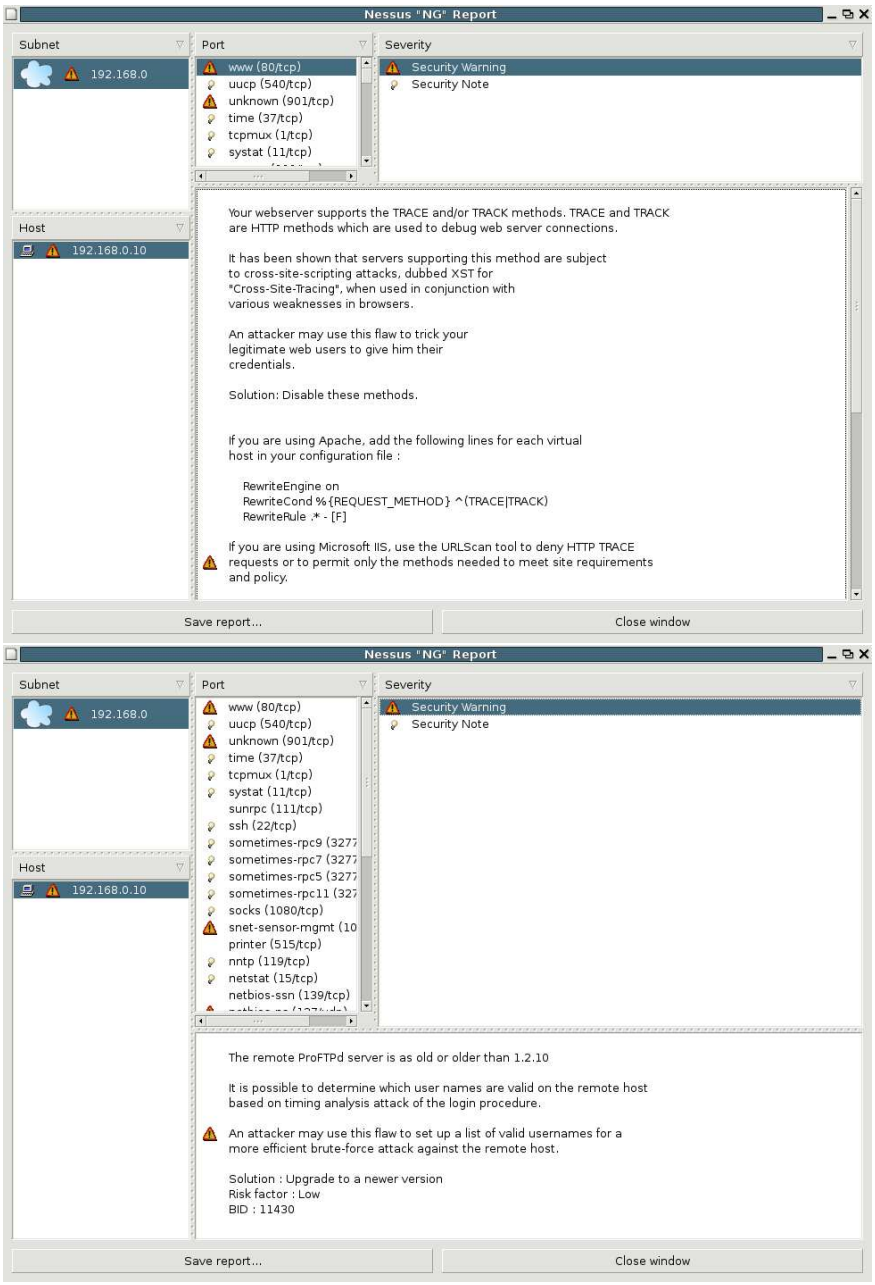

Figura 17.2: Vulnerabilidades encontradas en el sistema

Básicamente podemos decir que la seguridad del servidor esta controlada. A esto únicamente habría que añadir alguna actualización de versiones para parchear posibles *exploits* descubiertos recientemente.

17.1.3. Otros interfaces de configuración

También podemos ejecutar otras interfaces gráficas para Nessus:

- NPI: Interfaz PHP. Se puede descargar en: *http://enterprise.bidmc.harvard.edu/pub/nessus-php/*
- NCC: Nessus Command Center. Se puede descargar en: *http://www.netsecuritysvcs.com/ncc/*

17.2. Nmap: Escáner de red y puertos

La herramienta de exploración de red y escáner de seguridad, Nmap es posiblemente el mejor escaner de puertos existente, permitiendo determinar, de una forma rápida y sencilla, que servidores están activos y qué servicios ofrecen, es decir sus puertos abiertos.

Es una de esas herramientas de seguridad imprescindible para cualquier administrador de sistemas, siendo utilizada diariamente en todo el mundo, tanto por piratas informáticos como por administradores de sistemas. Su software se ha utilizado en otros muchos programas y ha sido portado a casi todos los sistemas operativos importantes.

Es un requisito previo, realizar un escanéo de vulnerabilidades con Nessus. Existen también disponibles varios complementos, incluyendo analizadores de salidas del programa, como por ejemplo Nlog.

17.2.1. Características básicas

Entre las características del Nmap podemos encontrar:

- *Flexible*: Soporta técnicas avanzadas para el mapeado de sistemas y redes que estén detrás de filtros IP, firewalls, routers y otros obstáculos. Estas incluyen mecanismos de escaneado de puertos (tanto TCP, como UDP), detección del sistema operativo, escaneos invisibles, conexiones semiabiertas, ...
- *Potente*: Se puede utilizar para escanear redes de ordenadores con cientos de máquinas.
- *Portable*: Existen versiones para la gran mayoría de los sistemas operativos modernos, entre ellos: Linux, Open/Free/Net BSD, Solaris, IRIX, Mac OS X, HP-UX, Sun OS, Windows (fase beta), ...
- *Fácil*: Aunque existen una gran cantidad de opciones disponibles, se puede realizar un sencillo escaneado de puertos con: `#nmap -O -sS <maquina>`
- *Libre*: El objetivo del proyecto Nmap es proveer a los *administradores, auditores e intrusos* de una potente herramienta de seguridad con la que explorar las redes. Nmap se distribuye con licencia GPL por lo que su código fuente está disponible para su descarga.
- *Buena Documentación*: Se ha realizado un gran esfuerzo en mantener actualizados y traducidos tanto las páginas man, como los tutoriales y el resto de documentación relacionada con Nmap.
- *Soportado*: Aunque Nmap viene sin garantía explicita de ningún tipo, se puede escribir al autor o utilizar las diferentes listas de distribución sobre Nmap. Existen varias empresas que incluyen soporte para Nmap entre sus servicios.
- *Premiado*: Nmap ha recibido multitud de premios y reconocimientos concedidos por revistas del sector.
- *Popular*: Diariamente cientos de personas descargan Nmap, además está incluido de serie en muchos sistemas operativos, como Debian. Esta gran popularidad es la mejor garantía de su calidad, soporte y desarrollo.

Tal y como hemos comentado, el uso del Nmap es muy sencillo, por ejemplo, para averiguar los servicios o puertos, accesibles de una determinada máquina, bastará con ejecutar: `#nmap <host> -O`

La salida que obtenermos del servidor que se ha configurado durante el proyecto se puede observar en la siguiente página.

```
#nmap 192.168.0.10 -O

Starting nmap 3.81 ( http://www.insecure.org/nmap/ ) at 2005-06-11 19:27 CEST
Interesting ports on 192.168.0.10:
(The 1652 ports scanned but not shown below are in state: closed)
PORT      STATE SERVICE
9/tcp     open  discard
13/tcp    open  daytime
22/tcp    open  ssh
37/tcp    open  time
80/tcp    open  http
111/tcp   open  rpcbind
113/tcp   open  auth
515/tcp   open  printer
631/tcp   open  ipp
721/tcp   open  unknown
10000/tcp open  snet-sensor-mgmt
Device type: general purpose
Running: Linux 2.4.X|2.5.X|2.6.X
OS details: Linux 2.5.25 - 2.6.3 or Gentoo 1.2 Linux 2.4.19 rc1-rc7), Linux 2.6.3 - 2.6.8
Uptime 0.274 days (since Sat Jun 11 12:52:40 2005)

Nmap finished: 1 IP address (1 host up) scanned in 2.372 seconds
```

Depués de descubrir los servicios que ofrece la máquina, con un simple telnet hemos obtenido el sistema operativo instalado en la máquina y la versión de *ssh*, si se conoce alguna vulnerabilidad de esa versión podría ser atacada: `telnet <host> <puerto>`

```
# telnet 192.168.0.10 22
Trying 192.168.0.10...
Connected to 192.168.0.10.
Escape character is '^]'.
SSH-2.0-OpenSSH_3.8.1p1 Debian-8.sarge.4

Protocol mismatch.
Connection closed by foreign host.
```

Existen muchas más opciones y alternativas, por lo que es más que recomendable acceder a la documentación incluida con Nmap, así como a la página man del mismo.

- `#nmap --help`
- `#man nmap`
- `#lynx nmap\manpage-es.html`

Estos escaneos de máquinas y redes, suelen dejar huellas de su ejecución en los registros logs de las máquinas escaneadas (por ejemplo, en */var/log/messages*), por lo que es interesante el utilizar alguno de los modos de escaneos invisibles que se pueden ejecutar con Nmap, tales como *-sF,-sX,-sN Stealth FIN, Xmas, or Null scan, ...* de forma que se evitar finalizar la negociación TCP, evitando al mismo tiempo el comentado registro en los ficheros logs.

Fyodor, el desarrollador de esta herramienta, tiene un gran sentido de humor, tal y como lo demuestra al implementar la opción -oS, que muestra la salida del Nmap en un formato que les encantará a los *Script-kiddies*, la podemos observar en el siguiente ejemplo:

```
#nmap -oS - 192.168.0.10+

$taRt|ng nmap V. 2.54B3T431 ( www.1n$ecur3.ORg/nmap/ )
|nt3r3sting pOrtz On debian.example.org (192.168.0.10):
(The 1545 Portz scannEd but nOT sh0wn bel0w ar3 In $tatE: cLOS3D)
POrt       Stat3       S3rv1Ce
22/tcp     OpeN        $$H
25/Tcp     OpEn        smtp
80/tcp     Op3n        htTp
139/tcP    op3n        N3Tb1Oz-Ssn
143/tCP    Open        imap2
515/tcp    f!lt3red    prinT3r
3128/tcp   Op3n        squ|d-HtTP
3306/tCp   Op3n        my$ql
6000/tcp   Op3n        x11

Nmap rUn c0mpl3ted -- 1 !P aDdr3Sz (1 hOst uP) scANnEd !n 3 $econdS
```

17.2.2. Tipos de escaneado

Existen muchos tipos de escaneado que se pueden ejecutar con Nmap. El cuadro 17.1 incluye una lista de los que probablemente usaremos con más frecuencia.

Cuadro 17.1: Tipos de escaneo en Nmap

Tipo de escanéo	Descripción
SYN: `-sS`	Escaneado predeterminado, no completa la comunicación TCP.
TCP Connect: `-sT`	Parecido a SYN, completando la comunicación TCP. Es un método ruidoso y carga en exceso las máquinas examinadas.
Ping Sweep: `-sP`	Un simple ping a todas las direcciones, comprueba las direcciones IP activas.
Escaneado UDP: `-sU`	Comprueba los puertos UDP para localizar los que escuchan.
Escaneado FIN: `-sF`	Escaneado sigiloso, como SYN pero enviando en su lugar un paquete TCP FIN. La mayoría de los hosts devolveran un RST.
Escaneado NULL: `-sN`	Escaneado sigiloso, establece los indicadores de encabezados a nulos. No es un paquete válido y algunos hosts no sabran que hacer con él.
Escaneado XMAS: `-sX`	Similar a NULL pero todos los indicadores del encabezado TCP se activan. Los Windows, por su estructura, no responderán.
Escaneado Bounce: `-n FTP_HOST`	Usa un agujero en el protocolo FTP, para rebotar paquetes fuera de un servidor FTP y hacia una red interna que normalmente no sería accesible.
Escaneado RPC: `-sR`	Busca máquinas que respondan a los servicios RPC.
Escaneado Windows: `-sW`	Se basa en una anomalía en las respuestas a los paquetes ACK en algún sistema operativo para mostrar los puertos que se suponen van a filtrarse.
Escaneado Idle: `-SI zombie_host_probe_port`	Escaneado sigiloso por el que los paquetes rebotan hacia un host externo.

17.2.3. Opciones de descubrimiento

También podemos ajustar la forma en que Nmap descubre la red y determina que hosts están activos. El cuadro 17.2 incluye diversas opciones:

Cuadro 17.2: Opciones de descubrimiento en Nmap

Opción	Descripción
TCP + ICMP: `-PB`	Utiliza paquetes ICMP y TCP para determinar el estado de un anfitrión. Es la forma más fiable y precisa, ya que usa los dos métodos.
TCP Ping: `-PT`	Usa solo el método TCP. Si estamos intentando ser sigilosos esta es la mejor opción.
ICMP Ping: `-PE`	No es una buena opción si el objetivo se encuentra detrás de un cortafuegos, la mayoría de los paquetes serán eliminados.
Dont's Ping: `-P0`	Nmap no intentará conocer primero qué hosts se encuentran activos en la red, en su lugar enviará sus paquetes a todas las IP en el rango especificado, incluso aunque no haya una máquina detrás. Puede ser la única forma de examinar una red bien protegida y que no responde a ICMP.

17.2.4. Opciones de ajuste de frecuencia de Nmap

Nmap nos ofrece la opción de agilizar o ralentizar la frecuencia con la que envía sus paquetes de escáner. Si estámos preocupado por la cantidad de tráfico de red (o estámos intentando ser sigilosos), podemos ralentizar el nivel. Sólo hay que tener en cuenta que cuanto más lejos los enviamos, más tiempo tardará el escaneado, algo que puede aumentar exponencialmente el tiempo de escaneado en redes grandes. Por otro lado, si tenemos prisa y no nos preocupa el tráfico de red adicional, podemos aumentar el nivel. Podemos ver los distintos niveles de frecuencia en el cuadro 17.3.

Cuadro 17.3: Configuraciones de frecuencia en Nmap

Frecuencia	Parámetro	Frecuencia de paquete	Comentarios
Paranoid	`-F 0`	Una vez cada 5 minutos	No utilizar esta opción en escaneados de varios hosts, ya que el escaneado nunca terminará.
Sneaky	`-F 1`	Una vez cada 15 segundos	
Polite	`-F 2`	Una vez cada 4 segundos	
Normal	`-F 3`	Tan rápida como permita el SO	Configuración predeterminada.
Agressive	`-F 4`	Igual que Normal pero la frecuencia del paquete se recorta a 5 minutos por host y 1,25 segundos por paquete de sondeo	
Insane	`-F 5`	0,75 segundos por host y 0,3 segunos por paquete de sondeo	Este método no funciona bien a no ser que estemos en una red muy rápida y usemos un computador realmente rápido, e incluso así podríamos perder datos.

17.2.5. Otras opciones de Nmap

El cuadro 17.4 recoge una lista de otras opciones para Nmap que controlan cosas como la resolución DNS, la identificación de SO y otras opciones. Existen más opciones para ajustar nuestros escaneados disponibles utilizando la interfaz de línea de comandos. Si queremos más detalles podemos recurrir al manual de Nmap.

Cuadro 17.4: Opciones deversas de Nmap

Opción	descripción
Don't Resolve: `-n`	Agiliza el escaneado, pero podemos perder hosts, sobre todo en redes con DHCP.
Fast Scan: `-F`	Escanea los puertos especificados, generalmente los puertos conocidos por debajo de 1024.
Port Range: `-p port_range`	De forma predeterminada Nmap examina los 65.536 puertos disponibles, con esta opción solo examina ese rango.
Use Decoy: `-D decoy_address1, decoy_address2, ...`	Se introducen IPs señuelo en el trafico mandado a la máquina que se esta examinando, así le resulta más dificil saber que máquina le esta escaneando.
Fragmentation: `-f`	Opción sigilosa, que fragmenta los paquetes de escaneado a mediada que sale. Los paquetes son montados en la máquina atacada y algunas veces pueden burlar los cortafuegos e IDS.
Get Indentd Info: `-I`	El servicio *Identd* que se ejecuta en algunas máquinas puede proporcionar información adicional sobre el host consultado.
Resolve All: `-R`	Esta opción intenta resolver las direcciones en el rango, incluso aunque no estén respondiendo.
SO Identification `-o`	Analiza la "huella digital" de las respuestas para determinar el SO
Send on Device: `-e interface_name`	Obliga a los paquetes de escaneado a salir de una interfaz específica.

17.2.6. Salida de Nmap

Nmap produce un informe que muestra cada dirección IP encontrada, los puertos descubiertos escuchando dicha IP y el nombre conocido del servicio (si lo tiene). También muestra si el puerto se ha abierto, filtrado o cerrado. Sin embargo, sólo porque Nmap obtenga una respuesta sobre el puerto 80 e imprima "http", no significa que un servidor Web se está ejecutando en el host, aunque es algo bastante probable. Siempre se puede verificar cualquier puerto sospechoso abierto consultando con dicha dirección IP sobre el número de puerto especificado y observando la respuesta obtenida. Si existe un servidor Web ejecutándose ahí, normalmente podremos obtener una respuesta mediante la introduccion del comando *GET/HTTP*. Así se devolverá la página inicial de índice como HTML (no como una bonita página Web), pero podremos verificar si un servidor se está ejecutando. Con otros servicios como FTP o SMTP podemos llevar a cabo tareas similares. Nmap también codifica con colores los puertos encontrados, según la siguiente tabla:

Cuadro 17.5: Codificación de color de la salida de Nmap

Color	Descripción
Rojo	Este número de puerto está asignado a un servicio que ofrece alguna forma directa de inicio de sesión en la máquina, como Telnet o FTP. Estos son los más atractivos para los intrusos.
Azul	Este número de puerto representa un servicio de correo como SMTP o POP.
Negrita	Estos son servicios que pueden proporcionar alguna información sobre la máquina o el sistema operativo.
Negro	Cualquier otro servicio o puerto identificado.

Como podemos comprobar en el cuadro 17.5, la salida nos permite examinar un informe y determinar rápidamente si hay más servicios o puertos con los que tenemos que tener cuidado, lo que no significa que deberíamos ignorar cualquier número inusual que no esté resaltado o en negrita. Los troyanos y el software de conversación se muestran normalmente como servicios desconocidos, pero podemos buscar un puerto misterioso en una lista de puertos malignos conocidos para determinar rápidamente si el puerto abierto es algo de lo que tenemos que preocuparnos. Si no lo podemos encontrar en dicha lista, tendremos que cuestionarnos cuál será ese servicio extraño que se está ejecutando en la máquina y que no utiliza un número de puerto conocido.

Podemos guardar los registros Nmap como números de formato, incluyendo el texto simple o legible por la máquina, e importarlos en otro programa. Sin embargo, si dichas opciones no fuesen suficientes, Nlog (sin licencia GPL) o alguna herramienta parecida puede ayudarnos a interpretar la salida Nmap. Su ejecución sobre redes muy grandes puede servirnos de salvavidas ya que el examen cuidadoso de cientos de páginas de salidas Nmap nos puede volver locos rápidamente.

17.2.7. Configuración gráfica de Nmap, interfaz Nmapfe

Incluido con Nmap se encuentra *nmapfe* que es un interfaz gráfica, que permite ejecutar Nmap usando el ratón. También indicar que existen otras interfaces gráficas para facilitar aún más el uso de esta potente aplicación (*KNmap, KNmapFE, QNMap, Kmap, Web-NMap, vnmap, ...*)

Para instalarla realizaremos un apt:
`#apt-get install nmapfe`

Y para ejecutarla: `#nmapfe` o `xnmap`

Nmap es una herramienta ideal para Verificar/auditar el Firewall, tal como dice uno de los banner de su página oficial, "Audite la seguridad de sus Redes antes que los chicos malos lo hagan" (Audit your network security before the bad guys do).

La página oficial de Nmap es: *http://www.insecure.org/nmap/*

17.3. Pruebas de carga

Aunqué en un principio pense en realizar una serie de pruebas de carga con diferentes clientes, para determinar el grado de sostenibilidad del sistema, no va a hacer falta. Carece de sentido probar algo que no va a aportar ningún dato nuevo, sino la confirmación de algo que temia desde un principio. Tales eran mis dudas que lo llege a comentar con el director del proyecto.

Durante la elaboración del proyecto, en todas y cada una de las secciones, en la información que he consultado, se advertía de que el uso de servicios de gran consumo de recursos en la misma máquina provocarían la ralentización del sistema hasta niveles inaceptables. Un ejemplo de esto sería el IDS Snort, trabajando junto a la base de datos MySQL o el el servidor Syslog.

Puesto que el servidor que utilizo es un ordenador portatil con procesador Intel Centrino a 1,6 Ghz. con 512 Mb de RAM, en vez de un supercomputador con con varios procesadores, no es necesario realizar estas pruebas para confirmar lo evidente. Considero que mi capacidad de optimización del sistema, probablemente, sea menor que la de muchas de esas personas que describen como "suicidio", respecto a rendimiento y seguridad, centralizar los servicios de una corporación en una única máquina.

Las razones que me llevan a esta decisión son varias:

- He asumido desde un principio, que el sistema no es capaz de ejecutar y soportar el uso simultaneo de los demonios de los servicios implementados. Esto se puede observar, por ejemplo, cuando arranco el servidor y Tripwire deja el sistema colapsado, al actualizar las sus bases de datos de archivos (Incluso con una prioridad nice baja (+10).
- Por el tamaño de la empresa que vaya a implementar el proyecto:
  - Para empresas pequeñas carece de sentido tener todos los tipos de servidores disponibles, ya que la mayoría no se utilizarán.
  - Para empresas grandes que necesiten todos los tipos de servidores, se disponen de presupuestos adecuados para distribuir los servicios entre varias máquinas. La limitación no esta en el coste de las máquinas sino en el personal que necesitan para configurarlas y administrarlas, además del coste que supone, el espacio físico que ocupan (en cuartos climatizados), medidas de físicas de seguridad, etc.
- Por motivos de eficiencia no se debe centralizar los servicios:
  - La máquina funcionaría al 100 % de su capacidad, obteniendose de ella un 10 % del rendimiento teórico.
  - Si se produjerá una caida fortuita de la máquina, como un fallo eléctrico o mecánico, todos los servicios de la red caerían con ella.
- Por motivos de seguridad no se debe centralizar los servicios:
  - Si un servicio tiene un *exploit* y un intruso obtiene el control de la máquina, obtiene a la misma vez el control de todos los servicios de esa máquina.
  - Los ataques dirigidos sobre una sola máquina son más eficientes que sobre varias.

Por todo ello, no creo necesario realizar pruebas de carga para determinar que no es una buena elección situar todos los servicios en una única máquina. Y que se debería de buscar una solución para descentralizar el sistema.

Capítulo 18

Estudio Económico

Mediante los siguientes esquemas se determinará el coste del proyecto de haber sido encargado por una empresa.

18.1. Recursos

En el siguiente cuadro se detallan los recursos que fueron necesarios para la elaboración del proyecto con una baremación aproximada de sus costes.

Cuadro 18.1: Recursos asignados al proyecto

Nombre	**Nombre corto**	**Grupo**	**Tipo**	**Coste**
Jose Antonio Escartín Vigo	Jefe de proyecto	Personal	Obra	40 €/h
Jose Antonio Escartín Vigo	Analista	Personal	Obra	30 €/h
Jose Antonio Escartín Vigo	Administrador de sistemas	Personal	Obra	25 €/h
Jose Antonio Escartín Vigo	Documentalista	Personal	Obra	15 €/h
Servidor	Portatil	Infraestructura	Material	1150 €
Cliente	Duron Linux/Windows	Infraestructura	Material	600 €
Cliente de alquiler 1	Windows1	Infraestructura	Material	100 €
Cliente de alquiler 2	Windows2	Infraestructura	Material	100 €
Portatil de alquiler	ClienteWifi	Infraestructura	Material	150 €
Impresora HP Deskjet 815	HP815	Infraestructura	Material	240 €
Router, Swich y cables	Redes	Infraestructura	Material	300 €
Material de oficina	Oficina	Infraestructura	Material	300 €
Local, luz e internet	Mantenimiento	Gastos	Material	1500 €
Desplazamientos	Desplanzamientos	Gastos	Material	100 €
Manual de administración Linux	Linux1	Libros	Material	48 €
Linux a fondo	Linux2	Libros	Material	30 €
Todo Linux	Linux3	Libros	Material	68 €
Sw libre: Herramientas de seguridad	Seguridad	Libros	Material	42 €
El libro de LaTeX	LaTeX	Libros	Material	38 €
Resto de libros e información proveniente de bibliotecas e Internet	Información	Libros	Material	— €

18.2. Costes

En la siguiente tabla, se puede encontrar una valoración aproximada de los costes totales del proyecto, basada en los sueldos/hora asignados a cada cargo del personal.

Cuadro 18.2: Costes del proyecto

Nombre	**Nombre corto**	**Tareas asignadas**	**N.º de horas**	**Coste**
Jose Antonio Escartín Vigo	Jefe de proyecto	Establecer la planificación y fijar objetivos	20 h.	800€
Jose Antonio Escartín Vigo	Analista	Seleccionar herramientas y analizar el grado de cumplimiento de los objetivos	50 h.	1.500€
Jose Antonio Escartín Vigo	Administrador de sistemas	Instalar sistema, configurar servicios y realizar las pruebas	220 h.	5.500€
Jose Antonio Escartín Vigo	Documentalista	Generar informes y documentar el proyecto	160 h.	2.400€
Material	Varios	—	—	4.766€
			TOTAL:	14.966€

18.3. Resumen económico

El coste del proyecto esta muy por encima de lo esperado, con un valor aproximado de 15.000€. Parece claro, que el factor que influye de una forma determinante es la mano de obra necesaria, estableciendo el valor del material en solamente un tercio del total, por dos tercios la mano de obra.

Manejando estas cantidades, un proyecto de este estilo solo está al alcance de empresas donde se vaya a utilizar este documento de manera intensiva, haciendo inviable un estudio de este tipo en *pymes*.

18.4. Modificaciones a los costes económicos

Debido a la falta de experiencia en sistemas Linux y en la creación de manuales, esta planificación económica se debería ver incrementada de la siguiente forma:

- Hasta las 350 horas en Administración de sistemas: Para la asimilación de los entornos, la configuración de los servicios y las pruebas necesarias.
- Hasta las 400 horas en Documentación: Debido al cambio de orientación del proyecto, hacia la elaboración de un "Manual de instalación de servidores en Linux".

Esto supodría un aumento, de valor respecto al proyecto original, de unos 7.000€, subiendo el precio total aproximado a los 22.000€.

Este aumento en el número de horas en ningún caso a derivado en un retraso en los plazos de entrega, ya que estos han sido respetados con riguridad. Y atienden a un mejor aprovechamiento y usabilidad de los contenidos del proyecto. Se podría decir que respecto a los objetivos iniciales, de "instalar un servidor", elaborando un "Manual de instalación para servidores Linux" se ha conseguido que el uso del proyecto pueda llegar a más ámbitos y convertirlo en útil para muchas personas.

Capítulo 19

Conclusiones

La conclusión principal es que los objetivos marcados inicialmente fueron erróneos:

- No es viable un servidor centralizado:
  - El poco coste del material, para una empresa, no compensa las desventajas.
  - La descentralización se apoya en: Modularidad, seguridad y redimiento.
- No es viable un servidor portatil corporativo e itinerante:
  - Sólo puede ser utilizado para servicios concretos.
  - Las redes inalámbricas producen graves problemas de seguridad intrinsecos e inevitables.
  - Las conexiones itinerantes a Internet, ofrecidas por los ISP, todavía estan en pañales.
  - La conexión es un problema, desde los siguientes puntos de vista:
    1. Se necesita un punto de acceso (AP) que acompañe al servidor, como podría ser un router.
    2. La conexión EAP-TLS a través de certificados digitales no es muy flexible a la incorporación de nuevos usuarios, que por ejemplo podrían ser los empleados de otra oficina, en otra ciudad. Este problema lo podríamos solventar agregando a nuestro sistema un programa de validación de conexión en nuestro sistema por web, como *NoCat*, donde los usuarios wifi acceden al servidor mediante usuario y contraseña.

El aumento de las horas dedicadas, que no los plazos de entrega, fue debida a que el proyecto evoluciono hacia la elaboración de un: *"Manual de instalación para servidores Linux"*, que abarca desde la elección de la distribución, pasando por la instalación del sistema y finalmente la configuración de la gran mayoría de servicios disponibles para entornos corporativos.

Este cambio de orientación, fue decidio a la mitad del proyecto, de nada servía documentar la instalación de un servidor Linux, si el proceso no podía ser reproduccido, a menos que se dispusiera de la misma máquina y la misma infraestructura. El proyecto se ha adaptado a un entorno más general, basado en la distribución Debian Sarge, para que los conociemientos adquiridos, puedan ser aprovechados por un número mayor de usuarios de escritorio y administradores de sistemas.

A modo de resumen me quedaré con la siguiente frase que lei mientras me documentaba y que esta en concordancia con el espíritu de este proyecto: *"En Internet sólo el paranoico sobrevive. Puede ser un lugar muy sucio, lleno de virus, gusanos, troyanos, spammers y abogados de litigios de patentes; lo último que alguien querría es ponerse en línea sin protección alguna."*

Es decir, el uso de:

- Conexiones seguras como: SSH (telnet), SSL (web), PGP (cifrado de datos), EAP-TSL (wifi), ...
- Firewalls, es decir sistemas de barrera contra ataques exteriores y fugas de información.
- IDS, sistemas de detección de intrusos: NIDS como Snort, IDS como Tripwire para garantizar la integridad en archivos, Honeypots como Honeyd para entretener los ataques de los intrusos, detectores de escaners de puertos, de sniffers y de rootkits, ...
- Monitorización del sistema: Top, Kismet Wireless, AirSonrt, ...
- Chequeos del sistema: Escaneo de vulnerabilidades con Nessus, escaneo de puertos y servicios en red con Nmap, escaneo de logs con Logcheck, ...
- ...

No garantiza la seguridad. Si un intruso quiere entrar, y tiene los conocimientos y/o herramientas suficientes, entrará. Las herramientas aquí propuestas son de carácter preventivo, disuasorio y forense. De esta forma es mucho más facil que el intruso se desespere y se vaya a hacer lo que quiere hacer, a otro sitio.

Los sistemas de este estilo, no son la panacea, exigen la dedicación de mucho tiempo y una administración rigurosa. Es decir, exigen de mucho tiempo y mucho dinero para que resulten efectivos.

No todas las empresas se podrán permitir este tipo de gastos. Hay que analizar hasta donde conviene proteger y aplicar una solución, que no será estandar, sino adaptada al perfil de nuestra organización o empresa.

A nivel personal a sido un proyecto muy gratificante y muy desesperante al mismo tiempo, esto a sido debido a varios motivos:

- El primero y principal, el proyecto lo plantee demasiado ambicioso. Partía con mucha voluntad, ganas e interes; pero con una gran falta de conocimientos de base, esto muy pronto se hizo patente.
- La planificación fue bastante imprecisa: Debido a mi falta de experiencia y conocimientos, los plazos se han aumentado muchisimo, llegando suplir esta falta de previsión con unas 300 horas extras respecto a la planificación inicial.
- A pocos días de la entrega final, se lanzo la versión estable del proyecto Debian Sarge. Esto derivo en dos consecuencias importantes:
  1. La primera positiva: La elección de Debian Sarge fue muy acertada, si hubiera elegido Debian Woody, habría terminado un proyecto que desde el principio sería obsoleto. Por contra, ahora está totalmente actualizado y será válido por un largo periodo de tiempo.
  2. En contra: La mayor parte del documento está escrito pensado que la versión estable se lanzaría en un futuro inmediato. Este futuro se ha adelantado un mes, la solución a pasado por agregar un anexo (véase anexo G) donde se explica que ha pasado y que implicaciones tiene esto en el documento.
- El proyecto lo podría clasificar de interés personal, ya que mi pasión de siempre, han sido los sistemas operativos. Por razones diversas no me había adentrado en el mundo de Linux, más que a un nivel muy superficial. Este proyecto me ha permitido profundizar e investigar sobre el mundo de la seguridad informática, sobre todo en entornos Linux y es muy probable que en el futuro, mi perfil profesional se ubique en este sector.
- Con el proyecto finalizo la Ingeniería técnica en Informática de Sistemas, y me servido para afianzarme en la idea de terminar mis estudios en la Ingeniera superior. Gracias a los conocimientos adquiridos, el proyecto de la superior me resultará mucho más facil de afrontar y probablemente también este enfocado hacia alguna parte concreta de la seguridad informática, sector que actualmetne está teniendo un gran auge.

- Los conocimientos adquiridos en el entorno de composición de textos LaTeX me permitirán, a partir de ahora, realizar documentación mucho más profesional.
- El descubrimiento del mundo del software libre, me ha abierto el horizonte, las licencias GPL permiten observar que hay futuro más haya de las empresas y los entornos propietarios. Cada uno tiene su sector de mercado y deben cooperar en vez de enfrentarse, asumiendo el papel que han elegido. Después de la elaboración de este proyecto, yo me he situado claramente en el sector GPL.

Por todo ello, lo considero una experiencia muy positiva y seguramente orientará mi futuro profesional al sector de la seguridad informática.

Una objetivo que me gustaría conseguir, si es posible (es decir, si la universidad y mi director de proyecto están de acuerdo), es poner a disposición de la comunidad de usuarios Linux este documento. De está forma mi trabajo podrá ser aprovechado por un mayor número de personas, ya que este no es el típico proyecto con una utilidad muy limitada, sino que tiene un uso mucho más amplio y puede llegar a ser útil a mucha gente.