Configuración de servicios

Jordi Serra Ruiz

PID_00204293

UOC
Universitat Oberta de Catalunya

Índice

Introducción 5

Objetivos 6

1. Servidor de ficheros e impresión 7
1.1. Servidor de ficheros e impresión a GNU/Linux 8
1.1.1. Instalación del Samba 9
1.1.2. Configuración del Samba 10
1.2. Servidor de ficheros e impresión en Windows Server 2012 16
1.2.1. Configuración de un servidor de ficheros 18
1.2.2. Configuración del servidor de impresión 26

2. Cortafuegos 31
2.1. Conceptos básicos 31
2.2. Recorrido de un paquete IP 32
2.3. Pasos para la creación de un cortafuegos en GNU/Linux 33
2.4. Otras órdenes de *iptables* 37
2.5. Configuración del *firewall* en Windows Server 39

3. Servidor de correo 42
3.1. Análisis de riesgos y prevención 43

4. Servidor de web y FTP 45
4.1. Servidor web a GNU/Linux 47
4.1.1. Instalación del Apache + SSL 47
4.1.2. Configuración Apache 55
4.2. Servidor de FTP a GNU/Linux 56
4.2.1. Instalación del servidor de FTP 57
4.2.2. Configuración del servidor FTP a GNU/Linux 57
4.3. Servidor web y FTP en Windows Server 2012 59
4.3.1. Servidor de información de Internet 59
4.3.2. Mecanismos de autenticación 65
4.3.3. Configuración de un lugar FTP 66
4.3.4. Registro de IIS 66
4.4. Análisis de riesgos y prevención 68
4.4.1. Web 68
4.4.2. FTP 70

5. Protección de puertos 72
5.1. Protección de puertos en GNU/Linux 72
5.1.1. `Xinetd` 73

5.1.2. Restricción de puertos 74
5.2. Protección de puertos en Windows Server 2012 77

Introducción

Una vez se ha instalado el servidor –el sistema operativo–, hay que instalar o poner en marcha, en los casos en que los servicios queden instalados con el sistema operativo mismo, todas las aplicaciones que nos servirán para dar un servicio adicional a nuestro servidor.

Un ejemplo claro de esto son el servidor de ficheros y el servidor de impresión. Se tiene que configurar el servidor para dar acceso a los directorios del servidor de ficheros con el fin de que los usuarios tengan un directorio personal en el disco de usuarios ubicado en el servidor. De este modo, se podrán hacer mucho mejor las copias de seguridad, se podrán controlar mejor y será más rápido hacerlas.

Otro servicio muy importante en una empresa es el correo electrónico. Hay que configurar el servidor para que reciba los mensajes de correo electrónico y los guarde en discos hasta que el cliente de correo electrónico del usuario acceda a los mensajes y los lea. Aquí se ve cómo se tienen que proteger las cuentas de correo para que solo haya un cliente autorizado que acceda a los mensajes enviados a las cuentas de correo personales.

Otra aplicación que requiere una configuración para proteger los datos de la empresa es el servidor web y sobre todo el servidor de FTP, que se tiene que instalar y configurar correctamente para no dejar "agujeros" de seguridad por los cuales puedan acceder los intrusos. Estos protocolos de intercambio de ficheros son ampliamente usados para acceder de manera fraudulenta a la información interna de la empresa.

Finalmente, se tienen que vigilar los puertos que se dejan abiertos en nuestro sistema informático. Un puerto abierto de manera errónea implica que un programa malicioso pueda usarlo para entrar remotamente al sistema y obtener información interna de la empresa.

Un caso típico lo constituye el puerto 25, que es el que usa el correo electrónico. Si se conocen las instrucciones de envío de correo electrónico, se puede acceder a un servidor que tenga abierto este puerto para enviar correos electrónicos como si los hubiera enviado la máquina a la que se accede.

Objetivos

En este módulo pretendemos que conozcáis la manera de configurar de forma segura algunos de los servicios más populares de la red (servidor de ficheros, correo, web y FTP). En el último apartado de este módulo veremos cómo tenemos que proteger los puertos de nuestros servidores.

Los objetivos de este módulo, por lo tanto, son los siguientes:

1. Aprender a instalar y configurar los servicios comunes de protocolos y herramientas de Internet.

2. Conocer la configuración segura de los servicios más comunes de la red.

3. Conocer los riesgos de estos servicios y cómo se tiene que hacer para prevenirlos.

4. Aprender a abrir y cerrar los puertos de nuestros servidores.

1. Servidor de ficheros e impresión

Los servidores de ficheros son los servidores que se encargan de ofrecer a los usuarios de la empresa un espacio de disco, normalmente en la red local, para guardar los documentos internos de la empresa. Una de las ventajas de este tipo de servidores es que, para hacer la copia de seguridad o *backup* de los documentos de los usuarios, solo tenemos que hacer una copia de seguridad del disco del servidor donde están ubicados los directorios o carpetas de los usuarios.

Esto también implica una tarea de educación por parte del administrador de la empresa para enseñar a los usuarios a guardar todos los documentos en las unidades de red y dejar el disco duro local, de cada ordenador de los usuarios, para almacenar el sistema operativo y los programas instalados de manera local. Si se trabaja en carpetas locales o, por ejemplo, con el escritorio, se hace mucho más complicado el control de los documentos y la realización de las copias de seguridad de estos documentos.

Las empresas IBM y Sytek. Inc. diseñaron en 1983 un sistema para construir y comunicar redes pequeñas (LAN). Este sistema incluía una aplicación llamada sistema básico de entrada-salida de red o *network basic input-output system* (NetBIOS). Esta aplicación estaba cargada en la memoria de las máquinas y proporcionaba una interfaz entre los programas. El sistema de identificación que usaba era un nombre de 16 bytes. Las aplicaciones actuales también publican en las redes en NetBIOS sus servicios mediante estos nombres cortos, así dan compatibilidad con otras aplicaciones y equipos que pueda haber en la red. Los nombres NetBIOS tienen que ser identificativos, es decir, tienen que ser únicos en la subred donde están.

Posteriormente, Microsoft añadió una serie de mecanismos para permitir que su sistema operativo MS-DOS fuera capaz de volver a encaminar las entradas y salidas del disco hacia las redes NetBIOS, cosa que permitió compartir los discos mediante la red. A este sistema de compartición de archivos después se le llamó bloque de mensaje del servidor o *server message block* (SMB). Actualmente, este sistema se conoce como sistema de archivos de interfaz común o *common internate file system* (CIFS).

Para saber dónde están las máquinas identificadas por los nombres NetBIOS, se diseñó el protocolo *NetBIOS Name Service* (*NBNS*), que hace funciones análogas al servidor de nombres de dominio (DNS) en el protocolo TCP/IP. Actualmente, este servicio se denomina *Windows Internet naming service* (WINS), que es del todo compatible con el NBNS.

Las aplicaciones SMB/CIFS funcionan aceptando o denegando la petición de acceso al recurso compartido según los privilegios que tenga cada usuario, y utiliza los puertos TCP 137, 138 y 139, por lo tanto, estos puertos los tendremos que tener en cuenta a la hora de abrir y cerrar los puertos de las máquinas.

1.1. Servidor de ficheros e impresión a GNU/Linux

En GNU/Linux hay varias aplicaciones que pueden proporcionar este tipo de servicios, pero hay dos que destacan por encima de las otras por el uso que se hace de ellas y por su utilidad.

- **Sistema de archivo de red** o *network file system* (NFS). Es un protocolo que permite exportar partes del árbol de directorios a otras máquinas. Las máquinas que importan el árbol de directorios lo montan como si se tratara de una partición local. Este tipo de protocolo solo permite exportar disco, pero es muy adecuado en entornos cien por cien GNU/Linux o Unix, aunque no permite exportar discos a entornos Microsoft.

- **El SMB** (más conocido como Samba) es una aplicación de fuente pública u *open source* (GNU) que permite compartir impresoras, además de exportar los discos, ficheros, crear controladores de dominio e incluso el directorio activo de los servidores de Microsoft. Es la reimplementación en software libre de los protocolos SMB/CIFS. Por lo tanto, esta aplicación funciona en entornos mixtos (GNU/Linux y Windows).

La aplicación Samba consiste en dos o tres demonios (*daemons*), dependiendo de las necesidades específicas de cada empresa. En el caso de la configuración mínima, hacen falta dos. Los demonios que forman el servidor Samba son los siguientes:

1) `nmbd`: este demonio maneja todas las peticiones de registro y resolución de nombres. Es una herramienta primaria involucrada en la navegación por la red. Este demonio es uno de los dos que hace falta como mínimo en toda configuración del Samba.

2) `smbd`: este demonio maneja todas las conexiones basadas en el protocolo TCP/IP para los servicios de discos y de las impresoras. También se responsabiliza de la autenticación. Este es el segundo demonio imprescindible en cualquier configuración del Samba.

3) `winbindd`: este demonio solo lo tenemos que iniciar cuando el servidor Samba es miembro de un dominio en el que no es el controlador de dominio principal o *primary domain controller* (PDC). También utilizamos este demonio cuando hay relaciones de confianza con otros dominios. Si queremos arrancar este demonio, debemos configurar en el fichero `smb.conf` los parámetros `idmap uid` y `udmap gid`.

En esta sección veremos cómo se tiene que instalar y configurar la aplicación Samba para trabajar en entornos mixtos.

1.1.1. Instalación del Samba

La instalación de esta aplicación la podemos hacer de dos maneras diferentes. La primera es aprovechando las órdenes que nos ofrece la versión Debian que tenemos instalada:

```
root# apt-get install samba
```

Si queremos que el programa Samba sea compatible con el *Active Directory*, antes de instalar el software del Samba tenemos que instalar los paquetes siguientes:

```
root# apt-get install libkrb5-dev
root# apt-get install krb5-user
```

Como la creación de paquetes Debian se hace después de la aparición de una versión nueva, debido a la manera en que se crean estos paquetes (se instala la versión compilada y después, con una herramienta de creación de paquetes, indicamos qué ficheros o directorios forman parte del paquete), si queremos instalar la última versión del Samba, lo tenemos que hacer por el método tradicional: bajando el código fuente, configurar el software y realizando finalmente una compilación de todo el programa.

Primero tenemos que bajar la última versión del software Samba que encontramos en el sitio web oficial de esta aplicación (http://www.samba.org). En la parte central de este lugar web hay una sección llamada *Current stable releases*, en la que hay un enlace hacia la última versión estable del software Samba, que se denomina Samba 4.0.3 (esta es la última versión en el momento de realizar estos materiales).

Una vez tenemos la aplicación en local, solo hay que hacer clic y bajar en formato comprimido el fichero con el software del Samba y descomprimirlo:

```
root# tar zxvf samba-4.0.3.tar.gz
```

Entramos en el directorio que acabamos de crear y ejecutamos la orden:

```
root# cd samba-4.0.3
root# ./configure --help
```

Elegimos qué opciones de compilación queremos añadir y ejecutamos el comando `configure` con los parámetros que nos interesa:

```
root# ./configure [... arguments ...]
```

Una vez hemos hecho la configuración, si nos interesa el *Active Directory*, nos tenemos que asegurar de que el fichero `include/config.h` contiene dos líneas parecidas a estas:

```
#defineix HAVE_KRB5 1
#defineix HAVE_LDAP 1
```

Llegados a este punto ya podemos compilar:

```
root# make
```

Si la compilación se hace de manera satisfactoria, instalamos la aplicación. Para hacerlo, se pueden instalar solo los ejecutables, solo los manuales o los dos a la vez. Las órdenes para hacer estas instalaciones son, respectivamente, las siguientes:

```
root# make installbin
root# make installman
root# make install
```

Si la instalación que hemos llevado a cabo es una actualización (*upgrade*) desde una versión anterior y queremos volver atrás (volver a la versión que estaba instalada con anterioridad), una vez hecha la instalación tenemos que ejecutar la orden:

```
root# make revert
```

Una vez acabada la instalación ya podemos configurar la aplicación.

1.1.2. Configuración del Samba

En esta sección explicaremos diferentes niveles de configuración del Samba, desde el modelo más sencillo hasta la configuración del Samba como controlador de dominio en un entorno mixto entre máquina GNU/Linux y máquina Windows, que nos permitirá controlar quién tiene acceso a las diferentes máquinas, y por lo tanto, hacer más seguro el acceso por parte de los usuarios. A pesar de que explicaremos los requisitos necesarios para usar herramientas de configuración del Samba utilizando entornos gráficos, todos los ejemplos que se muestran a lo largo de esta sección se basan en la edición de los ficheros.

La configuración del Samba está en el fichero de configuración `smb.conf`. Este fichero normalmente lo encontraremos en el directorio `/etc/samba` o en `/usr/local/samba/lib`, dependiendo de la distribución que hayamos usa-

do. Para configurar el Samba podemos editar este fichero o utilizar una herramienta gráfica basada en un entorno web llamada SWAT que está incluida en la propia instalación del Samba.

El fichero `smb.conf` está configurado en modo texto, de manera que lo podemos editar con nuestro editor preferido. La sintaxis es parecida a los antiguos ficheros de configuración del Windows y consiste en varias secciones independientes. Cada sección empieza con una palabra entre corchetes (`[ ]`) y representa un recurso compartido en el servidor. Hay una sección especial llamada `[global]` que contiene los parámetros que afectan a todos los recursos en general que se comparten con el Samba. Las líneas que hay dentro de cada sección consisten en un par de clave/valor separadas por un símbolo (=).

Los administradores de redes con sistemas Microsoft a menudo utilizan la nomenclatura de controlador de dominio, miembro de un dominio, o servidor *stand-alone* para referirse al servidor que hace las funciones de controlador de acceso a la red y al servidor. También podemos configurar el Samba para actuar en todos estos roles. Cada configuración solo provee un tipo de servidor, pero veremos tres configuraciones diferentes para cubrir los tres tipos de servicios que puede ofrecer.

Modos de seguridad del Samba

En los entornos de redes SMB/CIFS solo hay dos niveles de seguridad:

- *User security level*
- *Shared security level*

La implantación de estos dos niveles del Samba se ha hecho de manera más extensa y proporciona mejores funcionalidades a las iniciales en SMB/CIFS. Actualmente, el Samba ha implementado una versión de *shared security level*, pero ha hecho cuatro implantaciones diferentes del nivel de seguridad de usuario. Para no confundir la nomenclatura, el Samba denomina modos de seguridad a su implantación de los niveles de seguridad de SMB/CIFS. Las cuatro implementaciones del *user security level* se denominan `user`, `domain`, `ads` y `server`.

Antes de ver las configuraciones, veremos cómo son los modos de seguridad del Samba.

1) ***Usermode***. El cliente envía una petición de configuración de la sesión al servidor, que incluye el usuario y la contraseña. El servidor solo puede aceptar o denegar el acceso. En esta fase de la negociación, el servidor desconoce qué recurso compartido quiere utilizar el cliente y, por lo tanto, la aceptación o denegación solo tiene en cuenta el par usuario/contraseña o el nombre de máquina. Si el servidor acepta la conexión, es entonces cuando el cliente intenta acceder a los recursos compartidos, pero en todas las peticiones de uso de estos recursos no indica la contraseña, puesto que el cliente espera que todos los

derechos de acceso se hayan negociado en la configuración de la sesión. Para utilizar este nivel de seguridad en el fichero `smb.conf`, en la sección global, tenemos que poner el parámetro `security = user`. Este es el modo de seguridad por defecto.

2) *Sharedmode*. El cliente se autentica de manera independiente en cada uno de los recursos compartidos que quiere utilizar. Para hacerlo, envía la contraseña en cada petición de uso de un recurso. En esta petición no está el usuario, puesto que el cliente espera que con las contraseñas sea suficiente. A diferencia de otras implantaciones de nivel de seguridad *shared* que permiten autenticación con el par recurso_compartido/contraseña, el Samba siempre trabaja con el par usuario/contraseña (esquema de autenticación de GNU/Linux por defecto). Por suerte, para el Samba hay muchas aplicaciones clientes que envían también el usuario cuando se negocia la conexión. Así, el Samba recuerda los usuarios y, cuando recibe la petición de conexión a un recurso compartido solo con la contraseña, hace una comprobación de esta contraseña con los usuarios que tiene anotados. Si la contraseña de alguno de estos usuarios coincide con la que ha enviado el cliente, el servidor le permite acceder. Para usar este modo tenemos que añadir en el fichero `smb.conf`, en la sección global, el parámetro `Security = shared`. Debido a las pocas garantías de seguridad que nos ofrece este modo, es altamente desaconsejable usarlo, puesto que se está enviando continuamente la contraseña por la red y en el caso de tener dos usuarios con la misma contraseña, se producirían problemas.

3) *Domainmode*. Cuando el Samba opera en el modo de seguridad de dominio (*domain mode*), el servidor Samba tiene una cuenta de confianza (una cuenta de la máquina), que es un entorno en el que hay máquinas Windows y en el que el Samba no es el controlador del dominio, un servidor de la familia Windows. Esto hace que todas las peticiones de autenticación tengan que pasar por el controlador de dominio; dicho de otro modo, esta configuración hace que el servidor del Samba sea miembro del dominio, pero no el controlador. Para usar este modo de seguridad tenemos que añadir en el fichero `smb.conf` los parámetros siguientes:

```
security = domain
workgroup = name (our name group)
```

Para trabajar en este modo tenemos que unir el servidor Samba al dominio de seguridad de Windows:

- Usando el *Server Manager* en el controlador de dominio Windows, añadimos una cuenta de máquina para el servidor Samba.

- En el servidor Samba tenemos que ejecutar la orden usando la cuenta de administrador del controlador de dominio con la contraseña que tiene:

```
root# net rpc join -O administrador%password
```

Este modo también necesita la asignación de identificador de usuario (UID) para los usuarios del Samba. Para hacerlo, tenemos dos opciones: crear cuentas en el servidor Samba o utilizar el demonio `windbindd`.

4) *ADSmode*. El Samba se puede unir a un dominio *Active Directory* de Microsoft. Esto es posible si el dominio se ejecuta en el modo nativo. ¿Por qué nos interesa acceder desde una máquina GNU/Linux a un *Active Directory*? Si nuestro entorno está constituido por máquinas de Microsoft Windows, usando el Kerberos necesitaremos tener acceso al *Active Directory* para aceptar los "*tiques*" de Kerberos y tener por lo tanto acceso a todos los equipos de la red. Para usar este modo de seguridad tenemos que añadir en el fichero `smb.conf` los parámetros siguientes:

```
realm = our.real.kerberos security = ADS
password server = our.server.kerberos
```

5) *Server mode*. Este modo de seguridad se usaba cuando el Samba no era capaz de actuar como miembro de un dominio. Es muy recomendable no usar este modo, puesto que tiene muchos problemas de seguridad y las otras opciones ya cubren este comportamiento.

El modo de seguridad de servidor (*server mode*) actúa de *bypass* en modo usuario. Es decir, la petición que llega del cliente para acceder a los recursos compartidos es enviada al servidor principal del dominio. Si el servidor responde con una aceptación de la petición, el servidor Samba acepta la conexión del cliente. Si no, la deniega.

Para usar este modo de seguridad tenemos que añadir en el fichero `smb.conf` los parámetros siguientes:

```
encrypt password = yes
security = server
password server =
"Netbios_name_of_Domain_Controller(DCO)"
```

Configuración básica del Samba

Para hacer una configuración básica del servidor Samba, basta con poner las líneas siguientes en el fichero `smb.conf`:

```
[Global]
Workgroup = [real group]
Netbios name = [real name]

[Temp] Path = /tmp
```

```
[myhome] Path= /home/jordi
Comment = my home
```

Hay muchos más parámetros que veremos a continuación y que hacen referencia a la seguridad, al tipo de autenticación, etc.

A lo largo de esta sección utilizaremos varias versiones de este fichero de ejemplo, en las cuales `[real group]` es el nombre del grupo de trabajo en el que situaremos esta máquina y `[real name]` es el nombre Netbios del servidor. El campo `Path` indica el emplazamiento del directorio que queremos compartir, mientras que el campo `comment` es una explicación breve del contenido del recurso compartido.

Es importante que, una vez modificado el fichero `smb.conf` y antes de iniciar el servicio, ejecutemos la orden siguiente:

```
root# testparam /etc/samba/smb.conf
```

Esta orden hace "una revisión médica" del fichero `smb.conf` y nos da mensajes de aviso si encuentra errores en la sintaxis o parámetros desconocidos. Si el comando `testparam` se ejecuta de manera correcta, indica que el fichero de configuración es correcto.

Configuración del Samba como controlador de dominio principal

Para configurar el Samba como PDC, lo primero que tenemos que hacer es modificar el fichero `/etc/samba/smb.conf`. Este fichero tiene que tener un aspecto parecido a este:

```
[global]
Netbios name = [real name]
Workgroup = [real group]
Domain logons = yes
Domain master = yes
Security = user

Passdb backend = tdbsam
Password server = *
Os level = 33
Preferred master = yes
Local master= yes
Logon path = \\%N\profiles\%o
Logon drive = H:
Logon home = \\homeserver\%o\winprofile
Logon script = logon.cmd<
```

```
[netlogon]
Path = /var/lib/samba/netlogon
Read only = yes
Write list = ntadmin

[profiles]
Path = /var/lib/samba/profiles
Read only = no
Create mask = 0600
Directory mask = 0700
```

Las cinco primeras líneas de este ejemplo son imprescindibles para tener el Samba como PDC (*primary domain controller*). Los recursos compartidos de `netlogon` y `profiles` resultan necesarios en un entorno con sistemas mixtos con Microsoft Windows. En el primero es donde están los *scripts* de entrada al sistema (*logon*) y en el segundo es donde se almacenan los perfiles de usuario que se envían a las máquinas Windows para tener cada usuario un perfil personalizado. Como cada usuario tiene su perfil propio y lo tiene que poder modificar, el recurso compartido `profiles` tiene que ser configurado con permisos de escritura para los usuarios. Es decir, `Read only = no`, de otra manera quedarían fijados y no podrían cambiar nada, ni siquiera los accesos directos del escritorio.

Tenemos que tener presente que, cuando se configura el servidor Samba como PDC, por motivos de seguridad, tenemos que dar los pasos siguientes para que las máquinas que usan Microsoft Windows puedan acceder al dominio:

1) En el servidor Samba tenemos que ejecutar:

```
root# useradd -g machines -d /dev/null -c machine_name -s /bin/false machine_name$
root# passwd -l machine_name$
root# smbpasswd -a -m machine_name
```

2) Tenemos que tener muy presente que `machines` es un grupo del sistema. Por lo tanto, debe haber un grupo en `/etc/group` llamado `machines`, el cual, si hace falta, se tendrá que crear con anterioridad. También tenemos que saber que `machine_name` es el nombre de nuestra máquina; tenemos que sustituir el `machine_name` por el nombre correspondiente, respetando el signo $. Es importante no quitar este signo del comando.

3) Tenemos que configurar todos los sistemas Microsoft Windows como *domain members*. Si nos pide un nombre de usuario y una contraseña para unir la máquina al dominio, tenemos que usar la cuenta de *root* (o una cuenta con privilegios) del servidor Samba; para hacerlo, antes se tiene que haber incluido el usuario *root* en el fichero `/etc/samba/smbpasswd` mediante la orden siguiente:

```
root# smbpasswd -a root
```

(poner la contraseña de *root* para el Samba)

Dependiendo del método de autenticación de los usuarios (LDAP, PAM, Unix, etc.), estos usuarios tienen que tener una cuenta creada en el servidor del Samba. Es decir, si usamos el método de autenticación Unix, los usuarios tienen que tener una cuenta creada en el servidor y tienen que estar dados de alta en el fichero `/etc/samba/smbpasswd`. Si usamos LDAP o PAM, no hace falta crear ningún usuario en el servidor.

Documentación adicional

En el caso de tener máquinas muy antiguas que tengan instalado Windows XP Home Edition o Windows 9x/ME, se tiene que consultar la documentación siguiente:

http://us1.samba.org/samba/docs/man/samba-howto-collection/samba-pdc.html#id2518228

1.2. Servidor de ficheros e impresión en Windows Server 2012

La instalación de un servidor de archivos en el sistema permite poner a disposición de todos los equipos de la red un conjunto de ficheros comunes ubicados en este equipo. Los servidores de archivos de Windows Server 2012 tienen, entre otras, las características siguientes:

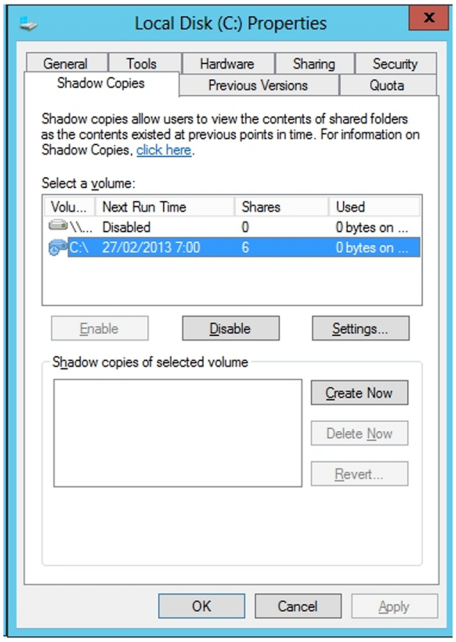
1) Archivos sin conexión: permite crear una copia local de los archivos que hay en una carpeta de red para utilizarlos mientras se está desconectado. Los archivos se actualizan automáticamente al volver a establecer la conexión.

2) Sistema de archivos distribuido o *distributed file system* (DFS): permite que el conjunto de archivos de red esté distribuido en varios servidores de archivos formando una estructura de árbol, cosa que facilita mucho la disponibilidad y la administración de los ficheros en la red.

3) Sistema de archivos NTFS: admite cifrar archivos, agregar espacio de disco a un volumen NTFS sin reiniciarlo, seguir los vínculos distribuidos y las cuotas de disco por usuario, que permiten controlar y limitar el espacio de disco de cada usuario y por lo tanto, también el espacio global ocupado por todos en el disco duro.

4) Mejoras en los permisos: se simplifica el establecimiento y la administración de permisos.

5) Servicio de instantáneas de volumen (VSS): crea una copia exacta de los archivos guardados en carpetas de la red compartidas e, incluso, de archivos que hay abiertos. Las aplicaciones pueden continuar escribiendo datos en el volumen del disco durante el proceso de implementación de *shadow copies*. VSS no elimina la necesidad de hacer copias de seguridad completas y regulares. La figura siguiente muestra cómo se configura esta opción.



El servidor de impresión del Server 2012 gestiona el acceso de los usuarios a las impresoras conectadas a la red. El servidor de impresión de Windows Server 2012 presenta, entre otras, las características siguientes:

- Aumento de las impresoras admitidas: proporciona compatibilidad con la mayoría de las impresoras actuales y relativamente recientes, cosa que permite utilizar las capacidades de impresión de diferentes sistemas operativos.

- Grupos de impresoras: permite configurar un servidor de impresión para compartir impresoras en la red.

- Protocolo de impresión en Internet o *Internet printing protocol* (IPP): permite a los usuarios administrar impresoras mediante un explorador web, imprimir mediante una página web y ver la información de los trabajos de impresión en formato de lenguaje de etiquetado de hipertexto o *hypertext markup language* (HTML). También les permite conectar con impresoras mediante un explorador web, cosa que simplifica el proceso de establecer conexiones de impresoras.

En el caso de que se quiera compartir archivos e impresoras con clientes muy antiguos como Windows 9x, hay que instalar el protocolo NetBEUI, y para clientes Windows XP, el protocolo IPX/SPX/NetBIOS, de este modo el servidor podrá entenderse con los clientes más antiguos.

1.2.1. Configuración de un servidor de ficheros

Creación de carpetas compartidas

Las carpetas compartidas constituyen el mecanismo más fácil para distribuir archivos en una red local. Las carpetas o directorios compartidos permiten poner a disposición de un usuario o de varios usuarios de la red local el contenido de una carpeta situada en otro equipo de la misma red local; por ejemplo, un servidor de ficheros.

Para configurar una carpeta como carpeta compartida, abrimos la ventana de propiedades de la carpeta (haciendo clic con el botón secundario del ratón sobre el icono de la carpeta, y seleccionando la opción "Propiedades"). En la pestaña "Compartir", seleccionamos la opción "Compartir esta carpeta" y ponemos un nombre al recurso compartido, que no hace falta que sea el mismo nombre que el de la carpeta real, pero tiene que ser único en todo el equipo (no debe haber dos recursos compartidos con el mismo nombre).

Si al final del nombre del recurso compartido ponemos el signo $, este recurso compartido será oculto en las búsquedas de carpetas o recursos compartidos de otro sistema. Esto, si bien parece que puede dar un poco de seguridad en el momento de compartir carpetas, no es del todo cierto, puesto que, aunque para usuarios "normales" es cierto, pues no verán los recursos compartidos, para herramientas de búsqueda especializadas no lo será, ya que son capaces de listar todos los recursos compartidos, estén o no ocultos con el signo $ del final.

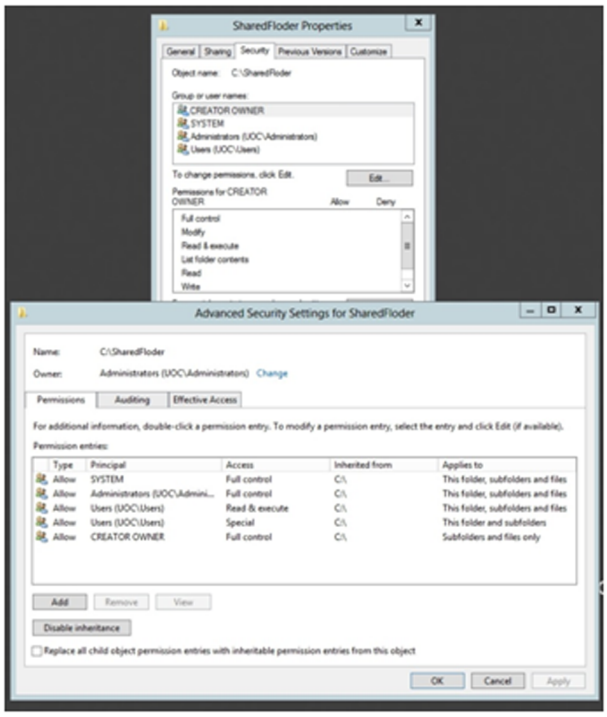
Permisos de acceso a carpetas compartidas

Además de especificar el nombre y un comentario para la carpeta compartida, podemos limitar el acceso a un número máximo de usuarios. Incluso podemos especificar qué usuarios en concreto tienen acceso a la carpeta compartida seleccionando la opción "Permisos". Dentro de las opciones de la carpeta compartida podremos incluir grupos de usuarios creados en el directorio activo, o directamente, ciertos usuarios para que tengan derechos de solo lectura o también de escritura.

La opción por defecto es compartir la carpeta con todos los usuarios. Aunque parezca contradictorio, con esta opción es suficiente, y normalmente es la deseada para no tener problemas posteriormente, a pesar de que se puede añadir y restringir el acceso mediante los botones "Agregar" o "Sacar". Para no tener problemas a la hora de compartir datos, es recomendable asignar permisos para NTFS (en la pestaña "Seguridad"), puesto que siempre se aplican primero los permisos del recurso compartido y después los permisos NTFS. Por lo tanto, lo mejor en muchos casos, no siempre, es tener acceso a todo el mundo, o a casi todo el mundo, y después restringir con más opciones a los usuarios con los permisos de la propia carpeta, y no del recurso compartido. Podríamos

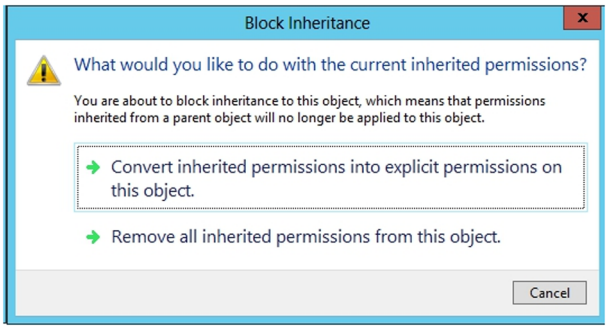
tener problemas y que un usuario tuviera permisos para NTFS, pero no para un recurso compartido. En este caso, el usuario no puede acceder a los datos del recurso compartido. Por eso, la cantidad de permisos NTFS es mucho más grande (figura siguiente). Hay que tener en cuenta que, para que nos salga la pestaña "Seguridad", la partición del disco tiene que estar formateada en NTFS. El sistema de ficheros FAT o FAT32 no proporciona control de acceso sobre los ficheros.

Permisos de la carpeta compartida



Los permisos de acceso a una carpeta compartida se pueden asignar a usuarios individuales o a grupos. Cuando se cambian los permisos de una carpeta principal, estos permisos se propagan al resto de carpetas y archivos que contiene la carpeta inicial. Para evitar la herencia de permisos en una carpeta determinada, se tiene que hacer clic en el botón "Deshabilitar la herencia". Cuando lo hacemos, sale un diálogo con las opciones mostradas en la figura siguiente.

Permisos heredados



- Copiar. Se copian los permisos heredados previamente de la carpeta principal y se deniega la herencia de permisos posterior.

- Sacar. Saca los permisos heredados previamente de la carpeta principal y solo mantiene los permisos que se han asignado explícitamente en la carpeta.

Carpetas compartidas sin conexión

La opción "Memoria caché" de la ventana de propiedades de la carpeta compartida permite activar o desactivar la opción de almacenar en una memoria caché o caché local de los ordenadores clientes los documentos para utilizarlos cuando no se dispone de conexión con el servidor. Por ejemplo, en el caso de usar un portátil, que al llegar a la organización se conecta a la red local y así se pueden sincronizar todos los documentos que han sido modificados.

El encargado de sincronizar los archivos de la red local con los de cliente es el administrador de sincronización, que hace que la versión más reciente de cada archivo esté disponible siempre, tanto en el servidor como en el PC cliente. Se puede configurar el administrador de sincronización para que sincronice los archivos en determinados momentos (al poner en marcha o parar el sistema, a una hora concreta, cuando el sistema está inactivo, etc.). Para sincronizar manualmente los archivos, seleccionamos la opción "Sincronizar" del menú "Herramientas" en una ventana del explorador de Windows.

En la ventana de sincronización seleccionamos los elementos que queremos sincronizar y pulsando el botón "Sincronizar" empezará la acción. El botón "Configurar" abre la ventana de configuración del administrador de sincronización, en la cual podemos especificar en qué momento se activará el administrador automáticamente.

Cuando dos usuarios hacen cambios sobre el mismo fichero, el segundo usuario que intenta sincronizar tiene que decidir qué versión del archivo quiere mantener o si se tienen que mantener las dos versiones.

Si bien es útil poder sincronizar los ficheros para poder trabajar desde fuera de la oficina, esta técnica puede comportar en algunos casos problemas a la hora de tener la última versión del fichero modificado, sobre todo en los casos en los que más de una persona puede editar el fichero.

En el caso de necesitar modificar documentos que se guardan en el servidor y tener mucho más controlado el acceso y las versiones, hay además otras soluciones, como tener configurada una VPN para acceder directamente al fichero, o un sistema de control de versiones de los ficheros instalado en el servidor.

Acceso a carpetas compartidas

Para acceder a las carpetas compartidas creadas en el servidor, se tiene que abrir una ventana del explorador de Windows, y en la barra de direcciones escribimos el nombre del servidor, seguido del nombre de la carpeta compartida:

```
\\server_name\folder_name
```

Si el acceso al recurso compartido y la carpeta se han configurado de forma que solo ciertos usuarios tengan acceso, se pide el usuario y la contraseña correspondientes con los cuales queremos acceder. Pero si se trata de un recurso general, en el que todos los usuarios tienen que tener acceso, no hará falta restringir y hacer que al entrar se pidan las credenciales.

Si se quiere que no salga una carpeta al listar los recursos compartidos de un servidor, `\\server_name`, hay que compartirla como `folder_name$`. En este caso, para acceder a ella tenemos que poner `\\server_name\folder_name$`. Pero esto tiene una desventaja y es que el efecto de ocultar la carpeta se produce para los sistemas operativos Windows como clientes. Desde un cliente con GNU/Linux, con el Samba configurado para poder acceder a recursos compartidos en un entorno Windows, aunque tenga el signo $ al final del nombre, se continúa mostrando, no dando la impresión deseada. Pero aunque se muestre, hay que disponer de permisos NTFS para acceder a la misma.

Se tiene que tener en cuenta que, en algunos casos, siempre se comparte la unidad C$, de forma que se oculta, pero continúa estando compartida.

Cifrado de archivos y carpetas

Para aumentar la protección de archivos y carpetas personales, se pueden cifrar. Para hacerlo, abrimos la ventana de propiedades de la carpeta o archivo que queremos cifrar, pulsamos el botón "Opciones avanzadas" que hay en la pestaña "General" y seleccionamos la opción "Cifrar contenido para proteger datos". Si la carpeta cifrada contiene archivos, tenemos que decidir si los archivos y subcarpetas que contiene se cifran también recursivamente. Los ficheros que se graban después en esta carpeta se cifran automáticamente.

Para descifrar un fichero o una carpeta, desseleccionamos la opción "Cifrar contenido para proteger datos" que se ha activado en el paso anterior. Con esto se garantiza que los usuarios no puedan acceder a esta información que puede estar ubicada en carpetas compartidas por más de un usuario, pero se tiene que tener en cuenta que el administrador del sistema tiene suficientes privilegios para descifrar todos los ficheros que se han cifrado por otros usuarios.

Al mover un archivo o una carpeta cifrados previamente a una partición FAT, los archivos se descifran automáticamente, puesto que el sistema de ficheros FAT no permite el cifrado.

La primera vez que se cifra un archivo o carpeta se crea un certificado autofirmado que permite cifrar y descifrar archivos. Es importante conservar este certificado, puesto que si cambiamos de máquina o reinstalamos el sistema, se volverá a generar otro certificado diferente y no podremos descifrar archivos cifrados con el certificado anterior. Por lo tanto, es muy importante poder guardar en un lugar seguro este certificado o se podría dar el caso de que, aun teniendo los ficheros cifrados, no se pueda acceder a su información por no tener este certificado.

Si se quiere compartir un archivo cifrado con otros usuarios, hay que distribuir también la clave pública del certificado, para que los destinatarios lo puedan descifrar; en caso contrario, a pesar de poder tener acceso al fichero no lo podrán abrir.

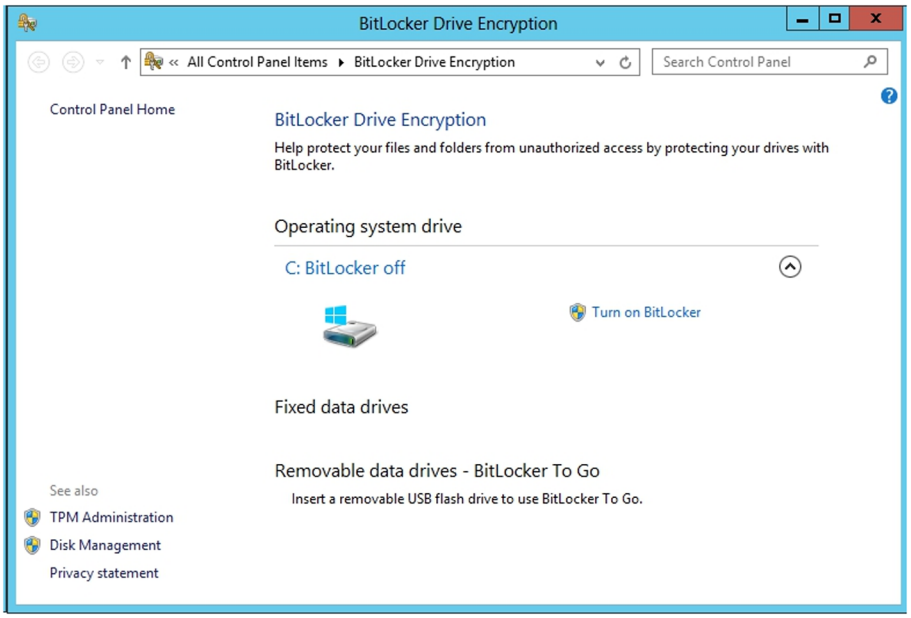
También se dispone de la nueva herramienta Bitlocker, que permite cifrar y proteger de una manera muy fácil todo el sistema de archivos o algún fichero o carpeta.

En la última versión, incorporada en el Windows Server 2012, se ha facilitado mucho más el uso para grandes empresas, agilizando el trabajo de los administradores de sistemas, que no están obligados a cambiar el PIN de acceso de cada usuario cuando no lo recuerdan, o pueden descifrar volúmenes remotamente a través de la red, lo que facilita la instalación remota de software en equipos que tienen que estar cifrados.

Mediante las directivas de grupo el administrador podrá decidir si permite al usuario del ordenador cifrar algunos ficheros o carpetas, o si se cifra todo el disco, o solo aquella parte del disco que realmente se utiliza. Esta configuración se encuentra en la directiva: Configuración del equipo \ Directivas \ Plantillas Administrativas \ Componentes de Windows \ Cifrado de unidad Bitlocker.

Esta herramienta de cifrado la instalaremos, si no está ya en alguno de los pasos anteriores, desde el administrador del servidor, dentro de las características del servidor que hay que instalar. La siguiente figura muestra la pantalla de configuración, donde se puede activar el cifrado del disco o de los archivos en el caso de que se hubiera configurado para poder cifrar ficheros individualmente.

Configuración Bitlocker



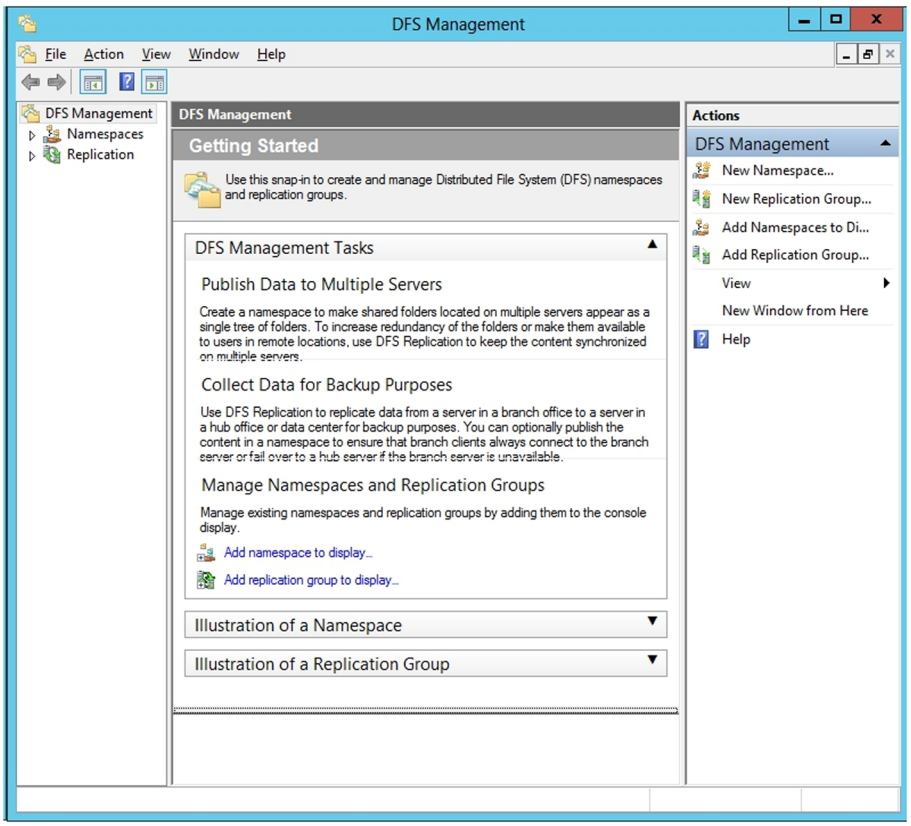
Distributed file system (DFS)

La herramienta "Sistema de archivos distribuido", que se encuentra dentro del rol de administrador de ficheros y que se tiene que instalar aparte de este servidor de ficheros, permite definir una estructura virtual de directorios compartidos entre diferentes servidores de manera unificada en un punto único de la red local. Esta herramienta proporciona transparencia sobre la localización física real de los archivos y sus respectivos recursos compartidos en los que se sitúan. De este modo, un usuario no tiene que recordar el emplazamiento real o el servidor donde está cada recurso compartido ni el nombre de este recurso, sino que tiene suficiente con conocer la ruta en la raíz del DFS. Además, permite aprovechar partes de discos de servidores y equipos que no se usan y hace que si se tiene que mover información de un disco a otro, o a otro servidor, sea completamente transparente y mucho más seguro desde el punto de vista del usuario, que seguirá teniendo la misma ruta para llegar a los ficheros.

Otra característica que también se puede instalar del DFS es la aplicación que hará que los datos de un disco estén replicados en otro disco de un servidor ubicado en la misma red local o incluso a través de la red Internet. Así podemos tener los datos replicados de manera automática. Cosa que nos da mucha más seguridad a la hora de guardar los documentos y poder acceder a ellos rápidamente. Por ejemplo, en el caso de tener dos sedes que tienen que acceder a ficheros comunes, una posible solución sería replicar estos ficheros para poder trabajar mucho más rápidamente.

El DFS define una estructura lógica de directorios en forma de árbol que empieza en un directorio principal compartido, denominado raíz de ficheros, a partir del cual se organizan los enlaces en las carpetas compartidas denominadas vínculos DFS. La figura siguiente muestra la configuración inicial del DFS.

Para crear una nueva raíz de ficheros, seleccionamos la opción "Nueva raíz DFS" del menú de las acciones. El asistente nos pide que seleccionemos el tipo de raíz DFS que queremos crear:



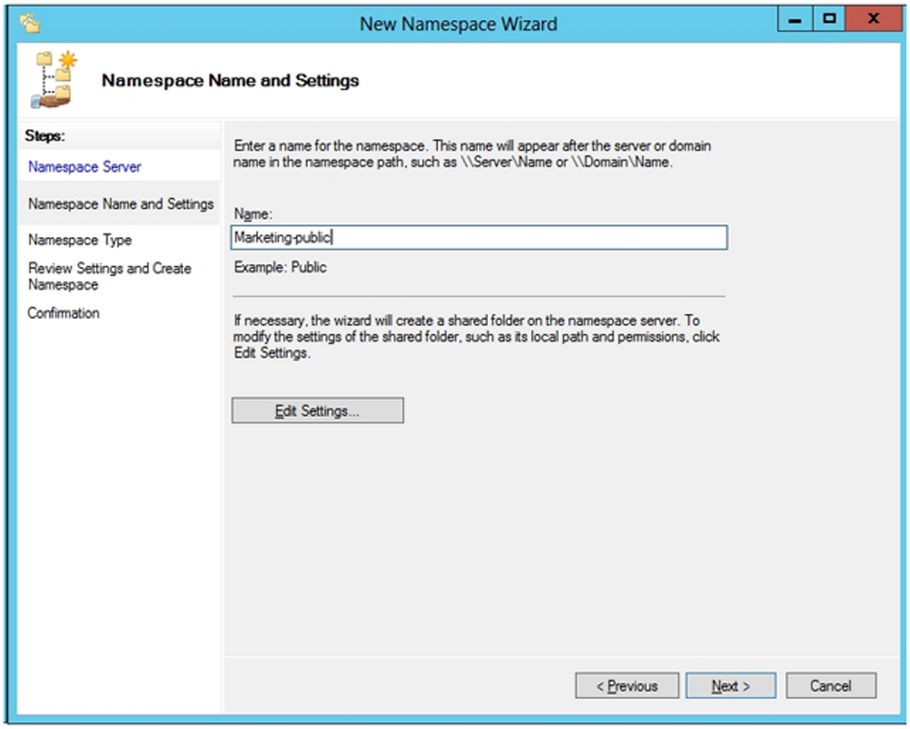
1) Raíz de dominio: raíz DFS basada en el directorio activo; la información del DFS se almacena junto con la información del dominio. Permite hacer duplicación automática y muchos niveles de vínculos.

2) Raíz independiente: raíz independiente del *Active Directory*. La duplicación se tiene que hacer manualmente y solo permite un nivel de vínculos.

La duplicación automática permite copiar la estructura del DFS a otra raíz en otro servidor para aumentar la disponibilidad y el equilibrio de carga. Una vez creada la raíz, podemos añadir vínculos a la misma. Un vínculo es un enlace virtual a un recurso compartido que hay en un servidor de la red. Podemos conseguir que un vínculo sea tolerante a fallos y con distribución automática de carga enlazando un mismo vínculo con otros recursos compartidos. El DFS se encarga de mantener la replicación entre recursos compartidos y de asegurar que, en todo momento, el contenido de todos los recursos compartidos que contiene un vínculo es idéntico.

El asistente de configuración del DFS pide el nombre del servidor (y el dominio si elegimos una raíz de dominio) donde se tiene que situar el DFS, y a continuación el directorio compartido en el que queremos situar la raíz DFS. Finalmente, tenemos que poner un nombre único a la raíz DFS en el dominio o en la red. Tal como se puede ver en la siguiente figura.

Nombre de la raíz de DFS



Para añadir una nueva carpeta compartida a la raíz DFS, seleccionamos la opción "Nuevo vínculo..." del menú contextual del elemento raíz en la lista de la izquierda de la pantalla. En la pantalla de creación del nuevo vínculo, nos piden el nombre y comentario correspondientes, y también la localización física a la cual será dirigido el usuario al seleccionar este vínculo.

Se pueden crear réplicas adicionales de una raíz DFS o de un vínculo DFS en concreto para aumentar la disponibilidad. La replicación de una raíz DFS solo se puede hacer si se trata de una raíz de dominio, y un mismo servidor solo puede almacenar una única raíz de dominio.

Para crear una réplica de raíz DFS, pulsamos el botón secundario del ratón sobre el elemento raíz, y seleccionamos la opción "Nuevo destino de raíz..." del menú contextual. El asistente pedirá el servidor donde se quiere crear la réplica, y la carpeta compartida a partir de la cual se almacenarán los vínculos DFS.

Una vez creada la réplica, definimos la política de replicación automática seleccionando la opción "Directiva de réplica" del menú contextual del elemento raíz. En la ventana de configuración vemos una lista de las réplicas de la raíz. Podemos seleccionar una raíz y hacer clic sobre "Habilitar" o "Deshabili-

tar" para incluir la raíz DFS en el proceso de replicación automática. Una de las raíces tiene que marcar el contenido que tienen que tener el resto de raíces. Esta raíz se denomina maestra. Para cambiar la raíz maestra, seleccionamos la nueva raíz y pulsamos "Establecer maestra".

Por otro lado, podemos crear réplicas de un vínculo DFS seleccionando la opción "Nuevo recurso..." del menú contextual del vínculo en la lista de la izquierda e introduciendo los datos del nuevo recurso compartido que tiene que contener la réplica del primero. La existencia de esta réplica tiene sentido solo si es en otro servidor de la red, puesto que, si cae uno de los dos servidores, siempre tendremos una de las réplicas a punto. Podemos definir cuál es la copia maestra (*master*) o la que replicará sobre los otros vínculos.

Si la raíz DFS es independiente, tenemos que copiar los archivos de manera manual, y también las posibles actualizaciones en el futuro. En cambio, si la raíz se basa en el directorio activo, seleccionamos la opción de réplica automática, de forma que el servicio de replicación de archivos se encargue de copiar y actualizar los archivos.

Si se selecciona la replicación automática, tenemos que especificar qué réplicas se actualizan automáticamente y cuál hace de maestra, como en la replicación de la raíz DFS.

Para acceder a un recurso de una raíz DFS desde un cliente, abrimos una ventana del explorador de Windows y en la barra de direcciones escribimos el nombre del servidor, seguido del nombre de la raíz DFS. Exactamente igual que si fuera una carpeta compartida por el propio servidor, para el usuario es completamente transparente. A partir de aquí podemos navegar por toda la estructura lógica DFS, sin saber a qué servidor y a qué carpeta concreta estamos accediendo. Los permisos de acceso a las carpetas reales físicas determinan si un usuario tiene acceso a un vínculo DFS o no. Si es una raíz DFS de dominio, basta con que accedamos al nombre de dominio:

```
\\domain_name\namespace
```

1.2.2. Configuración del servidor de impresión

Al igual que se pueden compartir carpetas y ficheros en una red local, también se pueden compartir impresoras, de forma que estén disponibles para algunos usuarios de la empresa. La impresora se añade a un servidor (servidor de impresión) y a continuación se comparte en la red.

Instalar y compartir impresoras

Para instalar y administrar impresoras de un servidor de impresión, utilizamos la herramienta normal de instalar las impresoras, la herramienta "Impresoras y dispositivos" del panel de control. Para agregar una impresora, hacemos doble

clic sobre el icono "Agregar impresora", y nos sale el asistente correspondiente. En la ventana "Impresora local o de red" seleccionamos si la impresora está conectada con el servidor o es una impresora con interfaz de red. Últimamente lo más usual es tener las impresoras conectadas directamente a la red local.

Una vez instalada la impresora, se tiene que compartir, para que los otros ordenadores puedan verla en el caso de que esté conectada directamente al servidor por USB o con puerto paralelo. En el caso de tenerla en red, no hace falta compartirla, pues ya lo está por estar conectada directamente a la red.

Además de compartir la impresora, tenemos que asignar los permisos de los usuarios que la pueden utilizar desde la pestaña "Seguridad". No todo el mundo podrá imprimir en todas las impresoras, o no hace falta que se le muestren todas las impresoras que hay en una organización, únicamente hace falta que pueda instalarse aquellas que tiene más cerca o sean del propio departamento, o con las características adecuadas para su trabajo.

Una vez configurada, los clientes del dominio pueden conectar la impresora desde cualquier explorador con `\\server_name` y haciendo doble clic sobre la impresora compartida. Los programes controladores necesarios se instalan de manera automática.

Especificación de la carpeta de cola de impresión

Cuando los clientes envían trabajos de impresión a una impresora, los datos que se envían se almacenan temporalmente (se ponen a la cola) en la carpeta de cola de impresión, que está por defecto en el directorio `system32\spool\printers` del directorio de instalación de Windows Server 2012.

Como los archivos temporales de impresión y los archivos del sistema operativo están en la misma unidad física, es posible que se reduzca el rendimiento de Windows y el del servicio de impresión. Por eso es aconsejable cambiar el emplazamiento de la carpeta de cola de impresión a otra unidad que tenga un controlador de disco independiente y con suficiente espacio libre para almacenar todos los archivos temporales de impresión.

Para cambiar la carpeta de cola de impresión, abrimos la herramienta "Impresoras y dispositivos" del panel de control y seleccionamos la opción "Propiedades del servidor" del menú "Archivo". En la pestaña "Opciones avanzadas" especificamos la unidad y el directorio en los cuales se guardarán los archivos temporales, y también más opciones relacionadas con los acontecimientos que se tienen que registrar en el servidor y notificaciones de impresión.

Permisos de acceso a impresoras

Los permisos de acceso a una impresora se configuran del mismo modo que los permisos de acceso a una carpeta. Para hacerlo, abrimos la herramienta "Dispositivos e impresoras", pulsamos el botón secundario del ratón sobre la impresora y seleccionamos la opción "Propiedades". En la pestaña "Seguridad" configuramos los usuarios o grupos que tienen permisos sobre la impresora y qué permisos tienen concretamente.

Impresoras compartidas con el *Active Directory*

Cuando se instalan y comparten impresoras en una red desde un servidor Windows Server 2012, este las publica por defecto en el *Active Directory*, cosa que facilita la localización de las impresoras según el tipo o emplazamiento que tienen.

Una vez abierta la herramienta "Usuarios y equipos *de Active Directory*", hacemos clic con el botón secundario del ratón sobre el elemento en el cual queremos publicar la impresora y seleccionamos la opción "Nueva" y, a continuación, "Impresora". Finalmente, introducimos el emplazamiento de la impresora.

Esta herramienta es útil en entornos con muchas impresoras compartidas y en emplazamientos distantes. Permite localizar las impresoras compartidas por equipos del dominio de manera rápida utilizando criterios de búsqueda, como por ejemplo impresoras con posibilidad de color, de impresión a dos caras, etc.

Análisis de riesgos y prevención

Ya se ha explicado cómo se configura un servidor para ofrecer diferentes servicios de manera segura. Este apartado introduce el concepto de los cortafuegos o *firewalls*. Un cortafuegos es un dispositivo de red que ejerce un papel muy importante en la seguridad. Es el encargado de dejar pasar un paquete de datos de la red hacia el interior de red o bloquearlo, y por lo tanto, no dejarlo pasar ni hacia fuera de la institución ni hacia dentro si la procedencia es externa. La configuración de estos equipos tiene que permitir que todos los servicios públicos que ofrece la empresa o institución sean visibles desde Internet. Por lo tanto, como los servicios ofrecidos no pueden ser protegidos por el cortafuegos, los tiene que proteger el sistema.

El caso del servicio de directorio es un caso diferente. Estos servicios no están pensados para ofrecerlos por Internet, sino para facilitar el trabajo de compartir documentos entre los trabajadores de la empresa y ofrecer un espacio de disco a los usuarios para guardar los documentos importantes, ya sean los compartidos o los personales. Por lo tanto, no podemos acceder a estos servi-

cios de manera directa desde Internet. El cortafuegos bloquea, o tendría que bloquear, todos los intentos de acceder a él remotamente. Es decir, no podemos utilizar el servidor de discos de la oficina desde nuestra casa.

Para acceder remotamente al espacio de disco de la empresa se han desarrollado otros servicios que nos permiten hacer esta conexión remota. Estos servicios son, por un lado, el FTP (claramente inseguro) y su homólogo, que utiliza puertos seguros SFTP, y por otro lado, las VPN, que permiten un acceso seguro al espacio de disco desde fuera de la empresa o institución. El caso del protocolo FTP lo veremos más adelante. El caso de las VPN lo hemos tratado en el módulo anterior.

Así pues, ¿cuáles son los riesgos de un servidor de discos si no podemos acceder desde Internet a este servicio?

Tenemos que tener presente que, sea por trabajadores descontentos, por errores humanos a la hora de tratar la información o por competencia desleal, la mayoría de los ataques serios que sufre una empresa tienen el origen dentro de la empresa; y a pesar de que también recibe muchos intentos de ataques del exterior, si está mínimamente configurada su seguridad, serán pocos los que puedan llegar a obtener datos de la empresa o entidad.

En la mayoría de los casos, un ataque desde dentro se produce por los motivos siguientes:

- Una configuración errónea en el servidor (el usuario puede acceder a documentos a los cuales en principio no tiene que poder acceder).

- Un trabajador sabe la contraseña de otro usuario de la empresa.

Estos ataques son muy difíciles de prevenir, puesto que por funcionamiento de la empresa tenemos que dar servicio a todos los trabajadores. Por lo tanto, no se pueden utilizar cortafuegos, ni restricciones de puertos, ni claves de autenticación si no están bien configurados, ya que podríamos cortar el acceso a la información por parte de usuarios legítimos.

Para prevenir estos tipos de ataques nos tenemos que asegurar de los puntos siguientes:

- Todos los trabajadores de la empresa tienen los privilegios que tienen que tener dentro del directorio activo.

- Todos los usuarios dados de alta en el servidor son los trabajadores actuales de la empresa. Es decir, cuando un trabajador deja la empresa, se elimina su usuario y se eliminan los privilegios de acceso de este usuario a todos los recursos.

- La política de contraseñas es suficientemente restrictiva como para no permitir contraseñas simples y obligar a los usuarios a cambiarlas a menudo.

Aunque sigamos todos estos puntos preventivos, sin embargo, no podemos asegurar que un trabajador no consiga la contraseña de otro trabajador, o que utilice un ordenador que no es el suyo mientras el compañero ha abandonado el ordenador temporalmente.

2. Cortafuegos

Un cortafuegos es un dispositivo de red que filtra el tráfico entre dos redes. Es un elemento básico en cualquier red con un mínimo de seguridad. El objetivo principal que tiene consiste en analizar los paquetes que la atraviesan y decidir, según un conjunto de reglas, si se tiene que descartar el paquete o se tiene que aceptar que continúe hacia el destino que tiene asignado.

En una máquina Linux podemos implementar un cortafuegos local o bien como cortafuegos que protege un conjunto de máquinas de una red interna. Para hacerlo, necesitamos la herramienta *iptables*, que se comunica con el núcleo o *kernel* de la máquina para indicar mediante un conjunto de reglas qué acción tiene que hacer con los paquetes.

La versión de Debian que se está usando ya tiene el apoyo integrado en el núcleo para *iptables* y, aún más, ya tiene instaladas las herramientas que se ejecutan en espacio de usuario *iptables*. En el caso de que no estuvieran instaladas, basta con ejecutar la siguiente orden:

```
root# apt-get install iptables
```

La herramienta *iptables* ofrece diferentes funcionalidades, como pueden ser: traducción de direcciones IP internas (NAT), priorización de paquetes, etc. Nos centraremos únicamente en el filtrado de paquetes para implementar un sencillo cortafuegos.

2.1. Conceptos básicos

Antes de empezar la parte más práctica de esta herramienta, introduciremos algunos conceptos que ayudarán más adelante a entender la configuración de un *script* (fichero de instrucciones) con *iptables*:

- Regla: Define una serie de atributos que tiene que contener el paquete de datos de la red para que se aplique la acción que hay asociada a esta regla. Estos atributos corresponden a valores de las cabeceras IP o TCP/UDP/ICMP del paquete que se está tratando.

- Acción: Hay una acción (*target*) por cada regla. La acción indica cómo se tiene que proceder con el paquete si el conjunto de atributos definidos en la regla coinciden con el paquete que se trata (*match*). Así, por ejemplo, se puede descartar un paquete o aceptarlo y pasarlo a niveles superiores, etc.

- Tabla: En *iptables* se definen tres tablas, cada una con un objetivo diferente:

– Tabla *Filter*. Para el filtrado de paquetes. Es la tabla que se usa para crear el cortafuegos.

– Tabla NAT. Para traducir direcciones IP.

– Tabla *Mangle*. Para modificar algunos campos del paquete de datos del protocolo IP.

- Cadena: Una cadena contiene un conjunto de reglas de filtrado que se aplican a los paquetes que atraviesan esta cadena. Cada cadena tiene un objetivo concreto. La tabla *Filter* contiene las cadenas *de input*, *output* y *forward*.

- Política: Es el comportamiento por defecto que tiene una cadena si no hay ninguna regla que indique la acción concreta que se tiene que hacer sobre el paquete concreto. Hay dos políticas generales que dependiendo del caso se usan de manera contraria. La primera descarta todos los paquetes excepto los que se indican explícitamente. La segunda hace todo lo contrario, es decir, acepta todos los paquetes por defecto y rechaza los que se indican explícitamente. En general, se utiliza la primera política, porque es la más segura por defecto.

2.2. Recorrido de un paquete IP

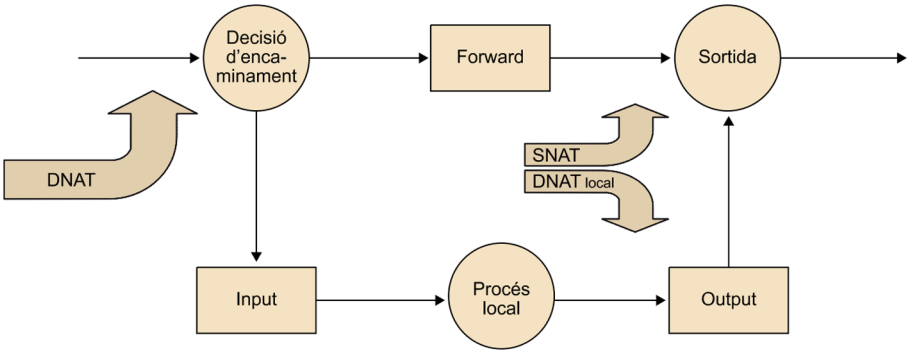
Cuando un paquete de datos entra por la interfaz de red, pasa por un conjunto de estados. De entrada se comprueba la dirección IP destino del paquete. Si coincide con la dirección IP de la máquina, indica que es un paquete destinado a la máquina propia, y por lo tanto, pasa por el conjunto de reglas de la cadena *de input*. Si se acepta el paquete, el proceso que la espera, es decir, el servicio o aplicación que está escuchando para recibir un determinado paquete, lo recibe y lo procesa.

Si, en cambio, la dirección IP es otra, el paquete se mueve hacia la cadena de *forward*. Si la máquina tiene el encaminamiento (*forwarding*) deshabilitado, el paquete es descartado automáticamente, mientras que si lo tiene habilitado, el conjunto de reglas en esta cadena decide el futuro del paquete y es encaminado o no hacia el destino final que tiene.

Finalmente, cualquier aplicación que genera tráfico de red envía los paquetes directamente a la cadena de *output*. Si las reglas aceptan el paquete, es dirigido hacia la interfaz de salida.

La figura siguiente resume el recorrido que puede llegar a hacer un paquete desde que entra por la interfaz de red hasta que se decide qué acción se hace:

Recorrido de los paquetes IP



Las siglas *DNAT* y *SNAT* indican en el diagrama anterior cuándo se puede hacer una traducción de direcciones IP (NAT), sea de la dirección origen (SNAT) o de la dirección destino (DNAT).

2.3. Pasos para la creación de un cortafuegos en GNU/Linux

En este apartado se muestra cómo se tiene que configurar un cortafuegos local en una máquina, por ejemplo, el servidor que se está configurando. Como la máquina tiene el encaminamiento deshabilitado, es decir, no se está haciendo de puente entre otras máquinas, no hace falta configurar nada dentro de la cadena de *forward*. Por lo tanto, toda la configuración se centra en las cadenas *de input* y *output* de la tabla *Filter*.

Como ya hemos comentado, los paquetes pasan por un conjunto de reglas, con una serie de condiciones que se tienen que cumplir para aplicar la acción que está asociada a la regla. Si al final se han aplicado todas las reglas sin que haya ninguna que coincida con el paquete que se está tratando, se aplica a la cadena la política por defecto, normalmente, cerrar todo el resto de puertos que no se han abierto con una regla específica para el puerto en cuestión.

Web recomendada

Si queréis más información sobre estos aspectos, consultad la página web siguiente:
http://www.netfilter.org/documentation/index.html

En este ejemplo inicial configuramos un *script* sin ninguna regla explícita, y cambiamos la política por defecto en las cadenas *de input* y *output* a *drop* (descartado). Haciendo esto indicamos que, por defecto, se descarten todos los paquetes que entran y salen de la máquina. Este *script* no es un cortafuegos propiamente, puesto que descartamos todo el tráfico e impedimos, a la vez, el acceso a Internet en nuestra propia máquina. Pero en muchos casos en los que se necesita poder aislar completamente la máquina debido a un incidente de seguridad, es una buena opción a tener en cuenta, ya que la máquina queda aislada de la red Ethernet.

```
root# cat script.sh
#!/bin/bash

/sbin/iptables -P INPUT DROP
/sbin/iptables -P OUTPUT DROP
```

```
/sbin/iptables -P FORWARD DROP
```

Mediante el parámetro `-P` configuramos la política por defecto de la cadena, que puede ser *accept* o *drop*. Hay que destacar que al aplicar este *script* perdemos conectividad con el exterior, de modo que hay que ejecutarlo desde la consola local de la máquina y nunca remotamente, pues no se podrá volver a recuperar la conexión. Para ejecutar el *script*, basta con dar permisos de ejecución y ejecutarlo con privilegios de *root*:

```
root# chmod +x script.sh
root# ./script.sh
```

Ahora, al *script* anterior le añadimos las reglas para que la máquina pueda servir páginas HTTP/HTTPS y además permita conexiones entrantes del protocolo SSH.

```
root# cat script.sh
#!/bin/bash

/sbin/iptables -P INPUT DROP
/sbin/iptables -P OUTPUT DROP
/sbin/iptables -P FORWARD DROP

/sbin/iptables -A INPUT -p tcp --dport 80 -j ACCEPT
/sbin/iptables -A INPUT -p tcp --dport 443 -j ACCEPT
/sbin/iptables -A INPUT -p tcp --dport 22 -j ACCEPT
```

Con el parámetro `-A` (*append*) añadimos una nueva regla al final de la cadena. El parámetro `-p` indica el tipo de protocolo superior, que en este caso es `TCP`. El parámetro `--dport` indica el puerto de destino del paquete, y en este caso hay una regla para los puertos de HTTP, HTTPS y SSH que se quieren configurar (80, 443 y 22 respectivamente). Finalmente, el parámetro `-j` indica la acción que se tiene que hacer con los paquetes que lleguen por cada uno de los puertos configurados, que puede ser *accept*, *drop*, *log*, etc.

De este modo todo lo que no esté configurado como *accept* en las últimas reglas se descartará automáticamente para las últimas reglas del *iptables*.

Para visualizar el conjunto de reglas que tenemos configuradas hasta ahora, ejecutamos la línea siguiente:

```
root# iptables -L -n
```

El parámetro `-L` hace que la orden liste el conjunto de reglas que se han aplicado y el parámetro `-n` nos omite la resolución de direcciones IP a nombres. Podemos consultar otros parámetros mediante la orden `man iptables`. El resultado de esta orden es el siguiente:

```
root# iptables -L -n
Chain INPUT (policy DROP)
target  prot  opt   source  destination
ACCEPT  tcp   --   0.0.0.0/0 0.0.0.0/0   tcp dpt:80
ACCEPT  tcp   --   0.0.0.0/0 0.0.0.0/0   tcp dpt:443
ACCEPT  tcp   --   0.0.0.0/0 0.0.0.0/0   tcp dpt:22
Chain FORWARD (policy DROP)
target  prot  opt   source   destination
Chain OUTPUT (policy DROP)
target  prot  opt   source   destination
```

Hay que destacar que, como no tenemos ninguna regla sobre los paquetes de salida (*output*) de la máquina hacia la red y como tenemos la política por defecto a *drop*, el tráfico de respuesta de nuestra máquina queda descartado a la salida, por lo tanto, si bien puede recibir paquetes de los protocolos que se han abierto, no puede enviar peticiones ni nada más por los mismos puertos ya que están cerrados. Se tiene que añadir una regla que permita el tráfico de respuesta. Por ejemplo una general:

```
/sbin/iptables -P OUTPUT ACCEPT
```

Con esta regla cambiamos la política por defecto a *accept*, cosa que permite todo el tráfico de salida de nuestra máquina al exterior, y esto incluye, por lo tanto, las respuestas a las peticiones de entrada en los puertos 80, 443 y 22. Pero no hay ningún control de puertos de salida, es decir, es como si no existiera el cortafuegos. Por lo tanto, mejor si nos decidimos por otra alternativa, siendo algo más restrictivos, la de permitir salir únicamente el tráfico que hemos dejado entrar:

```
/sbin/iptables -A OUTPUT -p tcp --sport 80 -j ACCEPT
/sbin/iptables -A OUTPUT -p tcp --sport 22 -j ACCEPT
/sbin/iptables -A OUTPUT -p tcp --sport 25 -j ACCEPT
```

Finalmente, la opción más cómoda y segura consiste en dejar salir únicamente lo que hemos dejado entrar, es decir, el tráfico que pertenece a conexiones que ya hay establecidas:

```
/sbin/iptables -A OUTPUT -m state --state ESTABLISHED -j ACCEPT
```

Hay que decir también que no se ha especificado ninguna interfaz de red en ninguna de las reglas, de modo que se aplican las reglas con independencia de la interfaz de red por donde entran o salen los paquetes.

No existen más que dos reglas para escribir: tener algo que decir y decirlo. Oscar Wilde.

Si visualizamos las reglas, tenemos la salida siguiente:

```
root# iptables -L -n
Chain INPUT (policy DROP)
target  prot  opt   source   destination
ACCEPT  tcp   --   0.0.0.0/0 0.0.0.0/0   tcp dpt:80
ACCEPT  tcp   --   0.0.0.0/0 0.0.0.0/0   tcp dpt:443
ACCEPT  tcp   --   0.0.0.0/0 0.0.0.0/0   tcp dpt:22
Chain FORWARD (policy DROP)
target  prot  opt   source   destination
Chain OUTPUT (policy DROP)
target  prot  opt   source
ACCEPT  all   --   0.0.0.0/0   0.0.0.0/0 state ESTABLISHED
```

Unas sucesivas ejecuciones del *script* irán añadiendo nuevas reglas al final de las cadenas de *input* y *output*, de manera que antes de volver a ejecutar el *script* hay que borrar las reglas anteriores. Para borrar las reglas se utiliza el parámetro `-F` (*flush*). Este paso es importante, ya que si se está probando con diferentes configuraciones y no se borran las reglas anteriores, puede ser que el sistema no responda correctamente a lo que queremos hacer con una regla en concreto.

El *script* queda de la manera siguiente:

```
root# cat script.sh
#!/bin/bash

/sbin/iptables -F
/sbin/iptables -P INPUT DROP
/sbin/iptables -P OUTPUT DROP
/sbin/iptables -P FORWARD DROP

/sbin/iptables -A INPUT -p tcp --dport 80 -j ACCEPT
/sbin/iptables -A INPUT -p tcp --dport 443 -j ACCEPT
/sbin/iptables -A INPUT -p tcp --dport 22 -j ACCEPT
/sbin/iptables -A OUTPUT -m state --state ESTABLISHED -j ACCEPT
```

El hecho de tener la política por defecto a *drop* sobre el tráfico de salida implica que impedimos que la máquina tenga conectividad al exterior. Esto puede comportar un problema, puesto que, por ejemplo, no podremos recibir actua-

lizaciones de software, pero es una ventaja, ya que, en caso de un acceso no autorizado a la máquina por robo de contraseña, evitamos la bajada de software malicioso.

2.4. Otras órdenes de *iptables*

Con el simple ejemplo anterior hemos visto algunas de las órdenes *de iptables* especificadas con los parámetros `-F`, `-A`, `-L`, etc.

Otro parámetro muy útil que acompaña al parámetro `-L`, es `-v`, que permite ver los contadores de cuántos paquetes y bytes han coincidido en cada una de las reglas. Finalmente, el parámetro `-Z` (cero) pone los contadores a cero.

```
root# iptables -L -n -V
Chain INPUT (policy DROP 6 packets, 1353 bytes)
 pkts bytes target  prot opt in out source  destination
  1  60 ACCEPT  tcp  --  *  * 0.0.0.0/0 0.0.0.0/0 tcp dpt:80
  0   0 ACCEPT  tcp  --  *  * 0.0.0.0/0 0.0.0.0/0 tcp dpt:443
  27 3066 ACCEPT  tcp  --  *  * 0.0.0.0/0 0.0.0.0/0 tcp dpt:22

Chain FORWARD (policy DROP 0 packets, 0 bytes)
 pkts bytes target  prot opt in out source  destination

Chain OUTPUT (policy DROP 23 packets, 7028 bytes)
 pkts bytes target  prot opt in out source  destination
  25 4670 ACCEPT  all  --  *  * 0.0.0.0/0 0.0.0.0/0 state ESTABLISHED

root# iptables -Z
```

En la inserción de reglas, el único parámetro que se ha visto hasta ahora ha sido el `-A` (*append*), pero también se puede usar el parámetro `-Y` para insertar una regla nueva en una posición concreta de la cadena, o el parámetro `-D` para eliminar una regla de la cadena. A continuación veremos una serie de ejemplos que utilizan las órdenes anteriores. De entrada tenemos el siguiente conjunto de reglas en la cadena de *input*:

```
root# iptables -L -n
Chain INPUT (policy DROP)
Target  prot opt source  destination
ACCEPT  tcp  --  0.0.0.0/0 0.0.0.0/0 tcp dpt:80
ACCEPT  tcp  --  0.0.0.0/0 0.0.0.0/0 tcp dpt:443
ACCEPT  tcp  --  0.0.0.0/0 0.0.0.0/0 tcp dpt:22
```

A continuación eliminamos la regla en lo referente al puerto 80 y visualizamos el resultado obtenido:

```
root# /sbin/iptables -D input 1
```

```
root# iptables -L -n
Chain INPUT (policy DROP)
Target  prot opt source  destination
ACCEPT  tcp  --  0.0.0.0/0 0.0.0.0/0 tcp dpt:443
ACCEPT  tcp  --  0.0.0.0/0 0.0.0.0/0 tcp dpt:22
```

Finalmente, en lugar de añadir la regla en lo referente al puerto 80 al final de la cadena, la insertamos en segunda posición:

```
root# iptables -L -n
Chain INPUT (policy DROP)
Target prot opt source destination
ACCEPT tcp -- 0.0.0.0/0 0.0.0.0/0 tcp dpt:443
ACCEPT tcp -- 0.0.0.0/0 0.0.0.0/0 tcp dpt:80
ACCEPT tcp -- 0.0.0.0/0 0.0.0.0/0 tcp dpt:22
```

Otra opción muy útil es la redirección de puertos que se puede usar o para esconder de manera sencilla, a pesar de que no sería del todo segura pero sí sencilla, los puertos estándares de comunicación, y también, en el caso de tener un servidor con máquinas virtuales, poder disponer de conexión de entrada desde la red hacia las máquinas sin disponer de IP externa las máquinas virtuales.

En este caso lo que se tiene que hacer es crear una regla en el PREROUTING, es decir, que se procesará antes que el resto, con el cambio de puertos y direcciones IP que son necesarios. Por ejemplo, si tenemos una máquina virtual que tiene el puerto de escucha del ssh configurado en el puerto 45022, y además se quiere que el puerto de escucha de la máquina real para responder a este puerto sea el 55022, se puede hacer de la siguiente manera.

```
root# iptables -t nat -A PREROUTING -p tcp --dport 55022
-j DNAT --to-destination 192.168.122.23:45022
```

De este modo la máquina virtual que tiene dirección IP 192.168.122.23, estará escuchando por el puerto 45022 lo que la máquina huésped está recibiendo por el puerto 55022. Esto permite tener diferentes máquinas que usen el mismo puerto, por ejemplo, el puerto 22 u 80, y tener el NAT en la máquina huésped con diferentes puertos y la misma IP (la del huésped).

Para ver las reglas configuradas para hacer NAT en la máquina huésped, se hace mediante la orden:

```
root# iptables -t nat -L
```

Las reglas que se van declarando en la configuración del cortafuegos se van guardando en el fichero `/etc/iptables/rules.v4`, por lo tanto, se puede modificar este fichero directamente para modificar las reglas del cortafuegos.

Con el comando `iptables-save` se puede guardar la configuración de las reglas del *firewall* que hay en funcionamiento y con el comando `iptables-restore` se pueden recuperar las reglas a partir de un fichero.

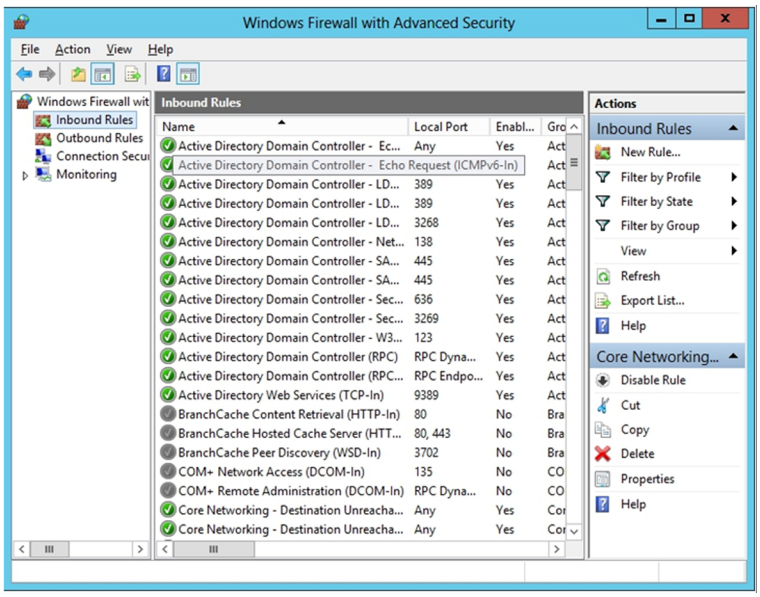
Para restaurar las reglas que se tienen guardadas, en un fichero propio o en el fichero `rules.v4`, solo es necesario hacer `iptables-restore < rules.v4`, de este modo se cargarán las reglas que haya en este fichero.

2.5. Configuración del *firewall* en Windows Server

La última versión del Windows Server 2012 ya dispone del *firewall* instalado y en funcionamiento desde el mismo momento de hacer la instalación. Por defecto, y no como en las primeras versiones en las que no lo tenía, el *firewall* ya viene configurado para mirar cuáles son los servicios que están configurados y adaptar la configuración. Así, si se instala un servicio nuevo sobre el servidor, este ya modifica las reglas del *firewall* para que todo funcione correctamente.

El cortafuegos lo podemos encontrar en el menú de herramientas del administrador del servidor.

Reglas del *firewall*



Las reglas se pueden configurar por separado, tanto de entrada como de salida. Las reglas de entrada son las que el servidor recibe desde fuera del propio sistema y se inician, por lo tanto, desde otro equipo informático, como por ejemplo, las validaciones al directorio activo, las peticiones DNS, el acceso a páginas web, etc.

Para poder responder a las peticiones externas, se tiene que disponer de una regla abierta por cada puerto o aplicación que está escuchando las peticiones de la red. Por lo tanto, para tener más seguro el sistema, todo aquello que no se usa, todos los puertos que no hace falta tener abiertos, no tendrán una regla abierta en el *firewall*, por lo que se bloqueará todo el tráfico que quiera llegar al servidor por estos puertos o aplicaciones que no están permitidos.

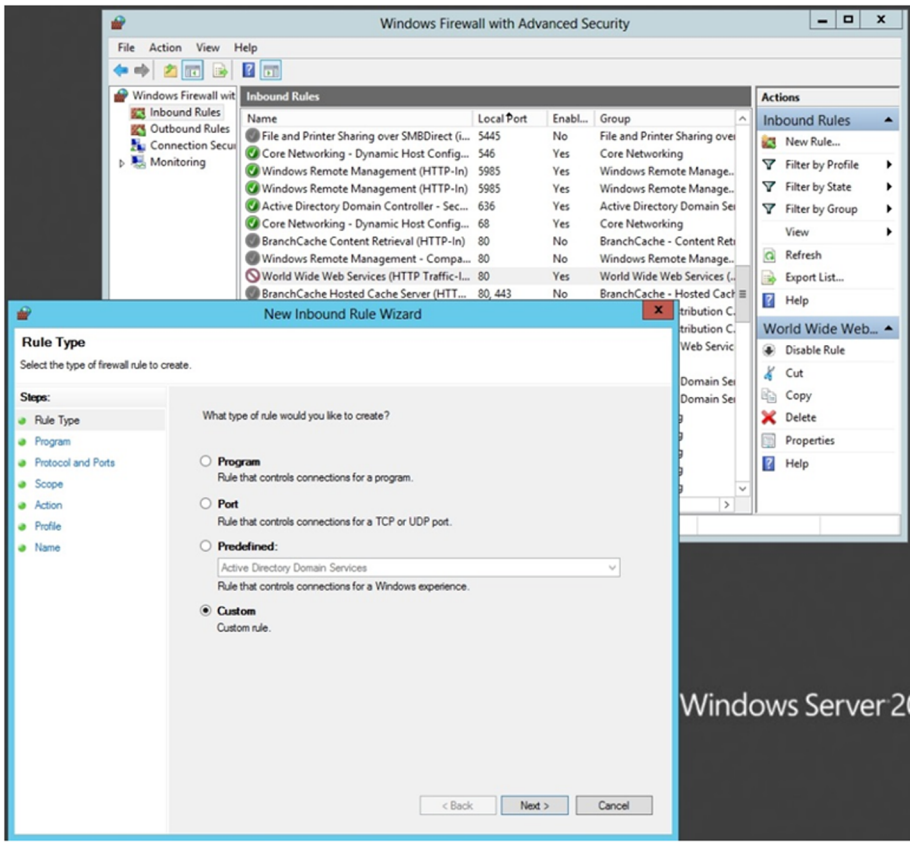
Las reglas de salida funcionan a la inversa de estas, son las que nos permiten o no dejar salir hacia el exterior.

Se pueden configurar nuevas reglas en el cortafuegos que sean de entrada o salida del sistema, a partir de la aplicación que se quiere autorizar a hacer la entrada y salida, o directamente autorizando a un puerto en concreto la comunicación en un sentido u otro, es decir, quién origina la comunicación. Se pueden definir reglas a partir de los puertos que necesitamos abrir, e incluso por servicios que ya están predeterminados y por lo tanto, es mucho más fácil de configurar, tanto por las reglas de entrada como por las reglas de salida.

Al monitorizar el propio cortafuegos los acontecimientos que va detectando, en la pestaña del monitor se pueden ver las reglas activas y los logs que van dejando las peticiones desde fuera.

Es muy importante tener completamente cerrados todos los puertos y aplicaciones que no se usen, tanto de entrada como de salida, puesto que así minimizamos los riesgos de ataques y tenemos mucho más controlado lo que está pasando en la red local. La regla general es tener todo cerrado a menos que se necesite tenerlo abierto porque hay un programa o protocolo que usa un puerto concreto de comunicaciones.

Configuración de las reglas propias



3. Servidor de correo

Actualmente, el correo electrónico es una de las aplicaciones de Internet que tiene más popularidad. En el uso del correo electrónico intervienen varios protocolos ya fijados. Dos de estos protocolos hacen las funciones de enviar los mensajes a los usuarios. Estos protocolos son el protocolo de acceso a los mensajes de Internet o *Internet message acces protocol* (IMAP) y la versión 3 del protocolo de oficina de correos o *post office protocol 3* (POP3).

El IMAP utiliza el puerto TCP número 143. La utilización de IMAP deja todos los mensajes y las carpetas donde se almacenan estos mensajes en el servidor. Esto permite acceder a la cuenta de correo desde más de un ordenador (el de casa, el de la oficina, etc.).

El POP3 en cambio utiliza el puerto TCP número 110. El funcionamiento de POP3 consiste en borrar los mensajes del servidor en el mismo momento en que el cliente baja los mensajes. Por lo tanto, si intentamos consultar el correo desde más de una máquina, nos encontramos que en cada una de las máquinas que utilizamos para consultar el correo electrónico hay una parte de todos nuestros mensajes. Para solucionar este problema, el POP3 incorpora actualmente una opción que permite dejar los mensajes en el servidor durante un cierto tiempo.

¿Qué ventaja tiene IMAP sobre POP3? Con IMAP podemos dejar en el servidor no solo los mensajes, sino también una estructura de carpetas donde se van almacenando los mensajes, mientras que en POP3 solo podemos dejar los mensajes directamente en la única carpeta que hay, ya que las carpetas se crean en local.

Tenemos que tener una cosa muy clara: tanto IMAP como POP3 solo son protocolos para recibir mensajes. Si queremos enviar un correo electrónico, tenemos que utilizar el protocolo simple de transferencia de correo o *simple mail transfer protocol* (SMTP). Este protocolo utiliza el puerto TCP número 25.

Los protocolos que se usan para enviar los mensajes utilizan canales no cifrados. Si queremos hacer estas funciones con protocolos cifrados, tenemos que trabajar con IMAPS y POP3S, que usan certificados SSL para cifrar el canal de comunicación. Estos protocolos utilizan los puertos TCP 993 en el caso de IMAPS y 995 en el caso de POP3S.

Estas configuraciones se tendrán que tener en cuenta a la hora de configurar correctamente el cortafuegos, puesto que si no creamos una regla por el *iptables* que deje pasar hacia fuera los paquetes por el puerto 993 (IMAP seguro), no se podrán enviar correos electrónicos desde aquella máquina o desde los clientes que usen este servidor para hacer de servidor de envío de correos electrónicos.

3.1. Análisis de riesgos y prevención

En estos últimos años, el correo electrónico se ha convertido en una herramienta de comunicación muy arraigada, tanto en la empresa como en el entorno doméstico. Por lo tanto, cualquier incidencia en este servicio repercute mucho en los usuarios.

El principal riesgo del correo electrónico es que con la lectura o ejecución de un mensaje se comprometa la seguridad de la máquina desde donde leemos el correo (no del servidor). Este tipo de ataques resulta especialmente peligroso porque la vulnerabilidad, como está adjuntada a un mensaje de correo leído ya dentro de la organización, puede atravesar los cortafuegos y acceder a la red interna de la empresa. Además, los ataques modernos se autopropagan, es decir, una vez infectada una máquina, se distribuyen (mediante el correo electrónico y los discos en red) hacia otras máquinas o usuarios. Los ataques que han tenido una difusión a escala mundial han sido el *iloveyou* y el *blaster*, que han sido distribuidos por correo electrónico.

Para acabarlo de complicar más para el administrador, los usuarios del correo electrónico esperan que, cuando mandan un mensaje, el receptor lo reciba casi en el acto. Antes esto era así, pero el retraso que se da entre el envío y la recepción es cada vez más grande. Muchas veces esto se debe a las medidas de seguridad que han tomado los administradores de sistemas de las máquinas que intervienen en el proceso de comunicación.

Todos estos puntos hacen que la solución a este tipo de problemas no sea nada trivial. Los métodos preventivos modernos se basan en motores de comparación de patrones que buscan posibles ataques (tanto si son virus como gusanos o caballos de Troya) y algoritmos bayesianos que intentan detectar cuáles son los mensajes buenos y cuáles son basura. Estos métodos, como son empíricos, generan algunos “falsos positivos”, es decir, catalogan mensajes como virus o como correos basura sin que lo sean realmente. Por lo tanto, nos podemos encontrar con que los mensajes enviados no llegan al destino que tienen debido a las medidas de seguridad que se han tomado.

Los falsos positivos aumentan o disminuyen proporcionalmente respecto a las medidas de seguridad. Si estas medidas son muy restrictivas, aumentan los falsos positivos, pero en cambio disminuye el riesgo de recibir ataques del exterior. Si las medidas de seguridad son muy permisivas, disminuyen los falsos positivos, pero aumenta el riesgo de que un ataque tenga éxito. Por lo tanto, como comprobamos, actualmente no hay una solución de mucha seguridad

y pocos falsos positivos. El trabajo de supervisar frecuentemente estas herramientas de prevención para que tengan el nivel de compromiso de seguridad adecuado a la política de seguridad de la empresa es del administrador de sistemas y de las máquinas que alojan el servidor de correo. Es decir, la que determina el nivel de seguridad de estas herramientas es la política de seguridad de la empresa, y la responsabilidad del administrador es adecuar esta política al servidor de correo.

No solo tenemos problemas en la recepción de los mensajes, sino que también podemos tener problemas con el servidor de correo mismo. Antiguamente, cualquier servidor de correo enviaba el mensaje de salida usando el protocolo SMTP de manera directa al destino; el puerto SMTP era abierto completamente y no se vigilaba. Esto motivó que se usaran estos puertos abiertos para enviar el típico correo basura a otros usuarios. Este hecho tenía consecuencias negativas para nuestro servidor: por un lado, la sobrecarga de la cola por el hecho de tener que enviar tantos mensajes y, por otro, los servidores que tenían reglas antiinundación consideraban nuestro servidor de correo como una máquina *non grata*, cosa que hacía que los mensajes de los usuarios reales no pudieran llegar al destino que tenían. Por lo tanto, no solo tenemos que introducir medidas de seguridad en las colas de mensajes entrantes, sino también en los ficheros de configuración de los servicios de mensajería.

Configurar nuestro SMTP para únicamente enviar y recibir mensajes de las máquinas que considera de confianza hace que haya cierta jerarquía entre máquinas y que, en consecuencia, los mensajes se concentren. Es decir, si todos los mensajes que envía nuestro servidor no los envía a los destinatarios sino al servidor de correo de nuestro proveedor de ISP, puede hacer que aumente el tiempo de espera de los mensajes en las colas entrantes.

Añadir a nuestro SMTP un motor de comparación que rastrea las colas de mensajes en busca de diferentes patrones predefinidos (los antivirus). Como cada día los ataques son más sofisticados, los antivirus de hoy en día no solo buscan virus, sino que han ampliado el radio de acción a caballos de Troya, gusanos e incluso algún tipo de correo basura.

Si tenemos en cuenta que actualmente la mayoría de servidores de correo tienen mecanismos de seguridad, puede llegar a pasar que, si nuestro servidor es considerado no grato, no podremos entregar todos los mensajes que envían los usuarios. Por lo tanto, los administradores se tienen que preocupar no solo de no recibir ningún tipo de mensaje no querido, sino del posible error de no enviarlo. En resumen, los motores de comparación de patrones no solamente recorren la cola de mensajes entrantes, sino que en muchos casos también recorren la cola de mensajes salientes.

4. Servidor de web y FTP

Hoy en día es difícil encontrar a alguien que no se haya conectado a Internet alguna vez. La mayoría de estas personas que utilizan la red de comunicaciones describirían el funcionamiento de una petición de la manera siguiente: "mi navegador genera una conexión a un servidor web, solicita una página y la recibe". Esta visión de funcionamiento de una solicitud de una página web, a pesar de que es correcta, es muy superficial.

Imaginémonos que queremos visualizar la página web siguiente:

http://www.uoc.edu/web/cat/index.html

El procedimiento visto más exhaustivamente es el siguiente:

1) La URL del navegador se divide en tres partes:

a) El protocolo utilizado (HTTP).

b) El nombre del servidor (www.uoc.edu).

c) El camino que se tiene que seguir (/web/cat/) hasta el fichero deseado (index.html).

2) Lo primero que hace el navegador es traducir el nombre del servidor a una dirección IP. Para hacerlo, lleva a cabo una petición DNS al servidor que tiene configurado para esta finalidad. Dependiendo del ISP que utilizamos, este servidor cambia. Y recibimos la respuesta con la dirección IP del servidor web.

3) Entonces, el navegador hace una conexión al puerto 80 de la máquina identificada por la dirección IP correspondiente al nombre (www.uoc.edu). Es decir, hace una conexión TCP al servidor web de la Universitat Oberta de Catalunya (UOC).

4) Mediante el protocolo HTTP, el navegador solicita obtener el archivo index.html.

5) El servidor envía al cliente el archivo.

6) El navegador del cliente interpreta las órdenes del archivo index.html y las muestra por pantalla.

En este ejemplo de conexión, además del funcionamiento de una petición, vemos que el modelo de conexión que utiliza el servidor web es de dos capas: la capa de presentación (que está en el cliente) y la capa de dominio (que está en el servidor).

A pesar de que parece un servicio sencillo, el servidor web hace más funciones que únicamente servir páginas web. Entre estas funciones está la de procesar información mediante *scripts* CGI y añadir algún nivel de seguridad a las tareas que tiene.

Se habla de procesamiento de información cuando en una página web sale un formulario para introducir texto y se obtiene un resultado diferente según el texto introducido. Un ejemplo de procesamiento de información es una búsqueda en una página web.

Si visitamos una página web y el navegador nos muestra una ventana de diálogo en la que nos pide el nombre de usuario y la contraseña, esta protección de páginas por contraseña también la hace el servidor web. Los servidores antiguos permitían que fuera el administrador de la máquina quien mantuviera una lista con los usuarios y sus respectivas contraseñas para acceder a estas páginas protegidas. Los servidores se responsabilizaban de autenticar los parámetros introducidos por los usuarios, comparándolos con los que había en la lista de acceso.

Hoy en día, estos niveles básicos de seguridad han quedado obsoletos, puesto que los servidores modernos, para hacer funciones de autenticación y de privacidad, utilizan canales cifrados con SSL. El puerto TCP de servicio web cifrado es el 443.

El protocolo de transferencia de ficheros o *file transfer protocol* (*FTP*), que pertenece a la familia de protocolos TCP/IP, empezó siendo una utilidad incluida en el propio sistema operativo Unix, que se usaba para transferir archivos entre los equipos conectados a una red. Actualmente, ya es un protocolo estándar. El funcionamiento del FTP se basa en la comunicación cliente-servidor.

A diferencia de otros protocolos, el FTP utiliza dos puertos de TCP para hacer la comunicación:

- El puerto de control, donde se hace el diálogo entre el cliente y el servidor. Es el puerto 21.

- El puerto de datos, donde se hace la transferencia de información entre el cliente y el servidor. Es el puerto 20.

El protocolo FTP permite copiar ficheros entre dos máquinas mediante red. Hay dos modalidades de FTP:

- El FTP propiamente dicho: hay que tener cuenta de usuario en la máquina a la cual se pretende acceder. Se suele usar para acceder a "nuestros" ficheros ubicados en el servidor. Es decir, nosotros, como usuarios, tenemos un espacio de disco situado en el servidor para guardar los documentos; cuando trabajamos en la empresa accedemos a este espacio mediante las unidades compartidas (o unidades de red) y cuando estamos en casa accedemos mediante el FTP.

- El FTP anónimo: no hay que tener cuenta. Se suele usar para conseguir programas de dominio público (software de prueba o *shareware*). Como nombre de usuario, la identificación es *anonymous*; como clave se usa el nombre de dominio y normalmente también el correo electrónico del usuario que pide la conexión.

Finalmente, tenemos que saber que el FTP es capaz de hacer transferencia de ficheros binarios y de texto.

4.1. Servidor web a GNU/Linux

4.1.1. Instalación del Apache + SSL

Partimos de la base de que ya está instalado el servidor de páginas web, en todo caso solo hay que hacer la instalación básica a partir de:

```
root# apt-get install apache2
```

Con este simple comando se instala el servidor web y configura una página web que ya es accesible. Pero a continuación se mostrará cómo hacer que esta página web pase a ser segura, es decir, que utilice el protocolo HTTPS en lugar del normal con HTTP.

Lo primero que hay que hacer es activar el módulo SSL del Apache, haciendo:

```
root# a2enmod ssl
```

Y reiniciar posteriormente el servidor de páginas web Apache, haciendo:

```
root# /etc/init.d/apache2 restart
```

Esto hará que tome ya la configuración del servicio SSL y por lo tanto este proceso de servicio de páginas escuche peticiones con el protocolo seguro, y no solo en el protocolo HTTP. Podemos ver si realmente está escuchando o no por el puerto del protocolo HTTPS usando el siguiente comando y mirar si realmente existe el https en la respuesta.

```
root# netstat -tap | grep https
```

```
....
tcp6   0  0 [::]:https    [::]:*    LISTEN   1238/apache2
root#
```

Como podemos ver en la respuesta, el servidor web está también escuchando las peticiones por HTTPS, tal y como esperábamos que se produjera.

A partir de ahora crearemos el vhost `www.uoc-test.net` para hacer la configuración de esta página web en el servidor y que funcione con el protocolo seguro de HTTPS. Lo primero que hay que hacer es crear el directorio donde se tiene que colocar toda la información relacionada con este portal.

```
root# mkdir /var/www/www.uoc-test.net
```

El servidor Apache toma la configuración por defecto que tiene en el fichero `/etc/apache2/sites-available/default-ssl`, pero necesitamos cambiar algunas cosas para que funcione por el nuevo sitio que estamos construyendo. Así, podemos partir de la base de este fichero y editarlo para modificar lo necesario.

```
root# cp /etc/apache2/sites-available/default-ssl /etc/apache2/sites-available/
www.uoc-test.net-ssl
root# joe /etc/apache2/sites-available/www.uoc-test.net-ssl
```

y modificaremos el contenido para que quede de la siguiente manera, donde se modifica el correo electrónico y el nombre del servidor, además de los directorios donde están ubicados los documentos de las páginas web.

```
<IfModule mod_ssl.c>
<VirtualHost _default_:443>
#   DocumentRoot /srv/www/mydomain.com/public_html/
#   ErrorLog /srv/www/mydomain.com/logs/error.log
#   CustomLog /srv/www/mydomain.com/logs/access.log combined



 ServerAdmin webmaster@localhost
  ServerName www.uoc-test.net:443

 DocumentRoot /var/www/www.uoc-test.net
 <Directory />
  Options FollowSymLinks
  AllowOverride None
 </Directory>
 <Directory /var/www/www.uoc-test.net/>
  Options Indexes FollowSymLinks MultiViews
  AllowOverride None
  Order allow,deny
```

```
 allow from all
</Directory>

ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
<Directory "/usr/lib/cgi-bin">
 AllowOverride None
 Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
 Order allow,deny
 Allow from all
</Directory>

ErrorLog ${APACHE_LOG_DIR}/error.log

# Possible values include: debug, info, notice, warn, error, crit,
# alert, emerg.
LogLevel warn

CustomLog ${APACHE_LOG_DIR}/ssl_access.log combined

#  SSL Engine Switch:
#  Enable/Disable SSL for this virtual host.
SSLEngine on

#  A self-signed (snakeoil) certificate can be created by installing
#  the ssl-cert package. See
#  /usr/share/doc/apache2.2-common/README.Debian.gz for more info.
#  If both key and certificate are stored in the same file, only the
#  SSLCertificateFile directive is needed.
SSLCertificateFile  /etc/ssl/certs/ssl-cert-snakeoil.pem
 SSLCertificateKeyFile /etc/ssl/private/ssl-cert-snakeoil.key

#  Server Certificate Chain:
#  Point SSLCertificateChainFile at a file containing the
#  concatenation of PEM encoded CA certificates which form the
#  certificate chain for the server certificate. Alternatively
#  the referenced file can be the same as SSLCertificateFile
#  when the CA certificates are directly appended to the server
#  certificate for convinience.
#SSLCertificateChainFile /etc/apache2/ssl.crt/server-ca.crt

#  Certificate Authority (CA):
#  Set the CA certificate verification path where to find CA
#  certificates for client authentication or alternatively one
#  huge file containing all of them (file must be PEM encoded)
#  Note: Inside SSLCACertificatePath you need hash symlinks
#     to point to the certificate files. Use the provided
#     Makefile to update the hash symlinks after changes.
```

```
#SSLCACertificatePath /etc/ssl/certs/
#SSLCACertificateFile /etc/apache2/ssl.crt/ca-bundle.crt

#  Certificate Revocation Lists (CRL):
#  Set the CA revocation path where to find CA CRLs for client
#  authentication or alternatively one huge file containing all
#  of them (file must be PEM encoded)
#  Note: Inside SSLCARevocationPath you need hash symlinks
#     to point to the certificate files. Use the provided
#     Makefile to update the hash symlinks after changes.
#SSLCARevocationPath /etc/apache2/ssl.crl/
#SSLCARevocationFile /etc/apache2/ssl.crl/ca-bundle.crl

#  Client Authentication (Type):
#  Client certificate verification type and depth. Types are
#  none, optional, require and optional_no_ca. Depth is a
#  number which specifies how deeply to verify the certificate
#  issuer chain before deciding the certificate is not valid.
#SSLVerifyClient require
#SSLVerifyDepth 10

#  Access Control:
#  With SSLRequire you can do per-directory access control based
#  on arbitrary complex boolean expressions containing server
#  variable checks and other lookup directives. The syntax is a
#  mixture between C and Perl. See the mod_ssl documentation
#  for more details.
#<Location />
#SSLRequire (  %{SSL_CIPHER} !~ m/^(EXP|NULL)/ \
#      and %{SSL_CLIENT_S_DN_O} eq "Snake Oil, Ltd." \
#      and %{SSL_CLIENT_S_DN_OU} in {"Staff", "CA", "Dev"} \
#      and %{TIME_WDAY} >= 1 and %{TIME_WDAY} <= 5 \
#      and %{TIME_HOUR} >= 8 and %{TIME_HOUR} <= 20    ) \
#      or %{REMOTE_ADDR} =~ m/^192\.76\.162\.[0-9]+$/
#</Location>

#  SSL Engine Options:
#  Set various options for the SSL engine.
#  o FakeBasicAuth:
#   Translate the client X.509 into a Basic Authorisation. This means that
#   the standard Auth/DBMAuth methods can be used for access control. The
#   user name is the `one line' version of the client's X.509 certificate.
#   Note that no password is obtained from the user. Every entry in the user
#   file needs this password: `xxj31ZMTZzkVA'.
#  o ExportCertData:
#   This exports two additional environment variables: SSL_CLIENT_CERT and
#   SSL_SERVER_CERT. These contain the PEM-encoded certificates of the
```

```
#   server (always existing) and the client (only existing when client
#   authentication is used). This can be used to import the certificates
#   into CGI scripts.
#  o StdEnvVars:
#   This exports the standard SSL/TLS related `SSL_*' environment variables.
#   Per default this exportation is switched off for performance reasons,
#   because the extraction step is an expensive operation and is usually
#   useless for serving static content. So one usually enables the
#   exportation for CGI and SSI requests only.
#  o StrictRequire:
#   This denies access when "SSLRequireSSL" or "SSLRequire" applied even
#   under a "Satisfy any" situation, i.e. when it applies access is denied
#   and no other module can change it.
#  o OptRenegotiate:
#   This enables optimized SSL connection renegotiation handling when SSL
#   directives are used in per-directory context.
#SSLOptions +FakeBasicAuth +ExportCertData +StrictRequire
<FilesMatch "\.(cgi|shtml|phtml|php)$">
 SSLOptions +StdEnvVars
</FilesMatch>
<Directory /usr/lib/cgi-bin>
 SSLOptions +StdEnvVars
</Directory>

#  SSL Protocol Adjustments:
#  The safe and default but still SSL/TLS standard compliant shutdown
#  approach is that mod_ssl sends the close notify alert but doesn't wait for
#  the close notify alert from client. When you need a different shutdown
#  approach you can use one of the following variables:
#  o ssl-unclean-shutdown:
#   This forces an unclean shutdown when the connection is closed, i.e. no
#   SSL close notify alert is send or allowed to received. This violates
#   the SSL/TLS standard but is needed for some brain-dead browsers. Use
#   this when you receive I/O errors because of the standard approach where
#   mod_ssl sends the close notify alert.
#  o ssl-accurate-shutdown:
#   This forces an accurate shutdown when the connection is closed, i.e. a
#   SSL close notify alert is send and mod_ssl waits for the close notify
#   alert of the client. This is 100% SSL/TLS standard compliant, but in
#   practice often causes hanging connections with brain-dead browsers. Use
#   this only for browsers where you know that their SSL implementation
#   works correctly.
#  Notice: Most problems of broken clients are also related to the HTTP
#  keep-alive facility, so you usually additionally want to disable
#  keep-alive for those clients, too. Use variable "nokeepalive" for this.
#  Similarly, one has to force some clients to use HTTP/1.0 to workaround
#  their broken HTTP/1.1 implementation. Use variables "downgrade-1.0" and
```

```
  #  "force-response-1.0" for this.
  BrowserMatch "MSIE [2-6]" \
   nokeepalive ssl-unclean-shutdown \
   downgrade-1.0 force-response-1.0
  # MSIE 7 and newer should be able to use keepalive
  BrowserMatch "MSIE [17-9]" ssl-unclean-shutdown

 </VirtualHost>
 </IfModule>
```

Como se puede ver, en el código anterior la página web está firmada con los certificados autofirmados que tiene por defecto Debian. Ahora vamos a probar si funcionan realmente los certificados y el servidor mediante el servicio HTTPS.

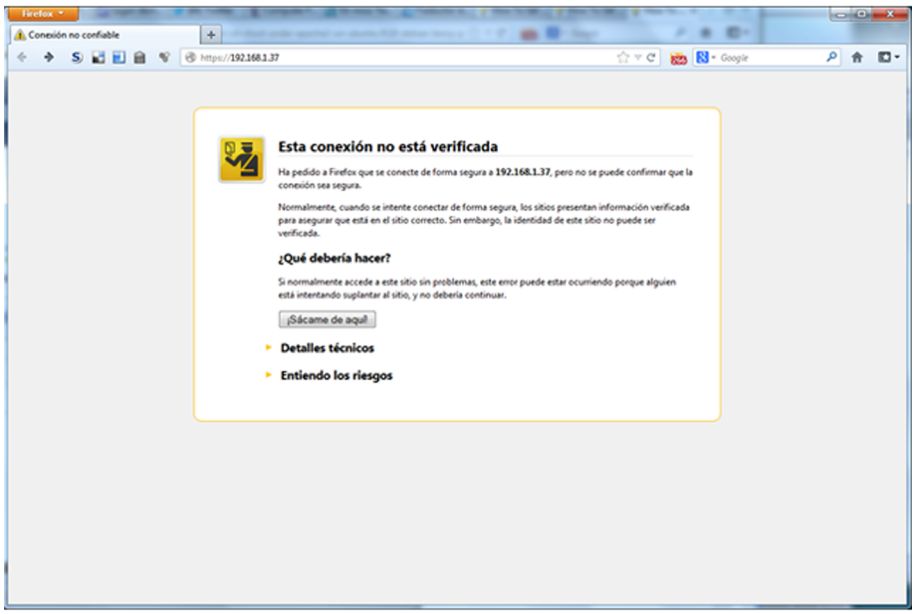
Lo que tenemos que hacer es cambiar el vhost por defecto por el que tenemos configurado.

```
root# a2dissite default-ssl
root# a2ensite www.hostmauritius.com-ssl
root# /etc/init.d/apache2 reload
```

Ahora ya podemos abrir un navegador y ver qué pasa cuando hagamos el llamamiento a la máquina que acabamos de configurar. Pero como se están usando los certificados autofirmados de Debian, no verificados por ningún organismo de certificación, el navegador avisará de esto, y nos deja seguir o no a partir de la responsabilidad de la persona que quiere visitar la página web, de si piensa o no que esta página web, firmada con certificados propios, es segura.

Por lo tanto, como muestra la figura siguiente, tener una página web con un certificado propio es relativamente sencillo y rápido de hacer, pero ya es cada persona la que tiene que saber si el acceso a esta página web es segura o no.

Página web creada



Mostramos ahora cómo se tienen que crear los certificados propios que después usaremos para acceder de manera segura a la página web.

Primero hace falta instalar el software que creará los certificados por ssl:

```
root# apt-get install ssl-cert
```

y crearemos los certificados propios haciendo la siguiente orden:

```
root# make-ssl-cert /usr/share/ssl-cert/ssleay.cnf /etc/ssl/private/www.uoc-test.net.crt
```

En la ejecución pedirá el nombre de la página web, donde tendremos que poner el nuestro (en este caso de prueba sería `www.uoc-test.net`). Esto creará un fichero `/etc/ssl/private/www.uoc-test.net.crt` con los dos certificados, tanto el público como el privado, y por lo tanto, los tendremos que separar en los dos ficheros correspondientes.

Si mostramos el contenido del certificado, veremos claramente dónde empieza uno y dónde el otro, por lo tanto, copiando y editando los dos ficheros podemos eliminar la parte que no interesa de cada uno de ellos. La parte privada del certificado (la primera) la guardaremos en el fichero `/etc/ssl/private/www.uoc-test.net.key` y el certificado autofirmado público lo dejaremos en el fichero `/etc/ssl/ciertos/www.uoc-test.net.pem`. Es importante que a la parte privada del certificado solo tenga acceso el usuario *root* de la máquina, para que nadie la pueda copiar o manipular. Por eso hay que hacer la siguiente orden que nos asegura que nadie tiene acceso al fichero.

```
root# chmod 600 /etc/ssl/private/www.uoc-test.net.key
```

Y también hay que borrar completamente el fichero generado inicialmente, puesto que tiene la parte privada del certificado incluida y por lo tanto se podría extraer de allí muy fácilmente:

```
root# rm -f /etc/ssl/private/www.uoc-test.net.crt
```

Ahora ya únicamente hace falta cambiar los vhost del SSL para que tengan los nuevos certificados generados por nosotros mismos. A pesar de que no solucionaremos el problema de la confianza en el certificado, tampoco serán los certificados por defecto que usa Debian y que cualquier persona puede obtener fácilmente, y podrían suplantar nuestra página web con el mismo certificado que estuviéramos usando en esta página. Por lo tanto, solo hay que editar el fichero:

```
root# joe /etc/apache2/sites-available/www.uoc-test.net-ssl
```

y hacer los cambios para que tengamos en cuenta los siguientes certificados:

```
  #  A self-signed (snakeoil) certificate can be created by installing
  #  the ssl-cert package. See
  #  /usr/share/doc/apache2.2-common/README.Debian.gz for more info.
  #  If both key and certificate are stored in the same file, only the
  #  SSLCertificateFile directive is needed.
SSLCertificateFile /etc/ssl/certs/www.uoc-test.net.pem
SSLCertificateKeyFile /etc/ssl/private/www.uoc-test.net.key
```

Ahora ya solo queda reiniciar el servicio Apache haciendo que vuelva a leer toda la configuración, mediante:

```
root# /etc/init.d/apache2 reload
```

Ahora ya solo queda crear un certificado que se pueda enviar a una entidad certificadora como puede ser Verising, Thawte o Comodo para que sean ellos quienes certifiquen que el dominio es legítimo y tendremos un certificado para nuestro dominio. Hay que notar que estas entidades certificadoras cobran por ofrecer este servicio, por lo que solo mostraremos los pasos hasta enviar el certificado a estas entidades.

Crearemos un directorio para poder trabajar cómodamente.

```
root# mkdir /etc/ssl/csr
```

A partir de la clave privada del certificado que se ha creado antes, crearemos un nuevo certificado, que será el que se usará para las páginas web, haremos:

```
root# openssl req -new -key /etc/ssl/private/www.uoc-test.net.key
```

```
-out /etc/ssl/csr/www.uoc-test.net.csr
```

Irá preguntando datos que se tienen que llenar correctamente para que salga bien la información en el certificado, en caso contrario, la entidad certificadora podría no emitirlo.

Es importante para poder verificar el dominio que en el campo de "Common name" se ponga el nombre del dominio que se tiene (en nuestro caso `www.uoc-test.net`) y que el resto de la información sea correcta.

Al final del proceso dispondremos del certificado que se tiene que enviar en el fichero /etc/ssl/csr/www.uoc-test.net.com.csr

La entidad certificadora nos enviará un nuevo certificado público, con extensión .PEM, que habrá que copiar en el directorio donde tenemos ahora el antiguo certificado autofirmado. En `/etc/ssl/certs/www.uoc-test.net.pem`

Hay entidades certificadoras que además del certificado que han enviado, también hacen que se haya de instalar uno propio. Para poder llevar a cabo la certificación, solo habrá que copiar el certificado en la misma carpeta y modificar el fichero de los vhost donde tenemos el camino de nuestros certificados para incluir el certificado de la entidad

```
root# joe /etc/apache2/sites-available/www.uoc-test.net-ssl

...
SSLCertificateFile  /etc/ssl/certs/www.uoc-test.net.pem
SSLCertificateKeyFile /etc/ssl/private/www.uoc-test.net.key

#  Server Certificate Chain:
#  Point SSLCertificateChainFile at a file containing the
#  concatenation of PEM encoded CA certificates which form the
#  certificate chain for the server certificate. Alternatively
#  the referenced file can be the same as SSLCertificateFile
#  when the CA certificates are directly appended to the server
#  certificate for convinience.
SSLCertificateChainFile /etc/ssl/certs/CAcert_chain.pem
...
```

4.1.2. Configuración Apache

Acabada la configuración inicial del servidor de páginas web Apache, miramos el fichero de configuración del servidor para ver que todo está correctamente configurado. Esta configuración puede estar dividida en diferentes ficheros de configuración. Para garantizar el correcto funcionamiento de las versiones antiguas de Apache, se ha optado por continuar teniendo los mismos ficheros y añadir la configuración de las nuevas versiones a otros ficheros.

Así, tenemos los ficheros `httpd.conf`, para las versiones antiguas, el fichero `apache2.conf`, y los ficheros de configuración de los vhost que se han mostrado anteriormente y que están ubicados para cada uno de los huéspedes virtuales que se tengan configurados en el servidor Apache. En el caso de instalar una nueva versión del Apache, el fichero `httpd.conf` estará completamente vacío y toda la configuración se pondrá en el fichero `apache2.conf`.

Encontramos también las variables de entorno del servidor web en un fichero aparte (`/etc/apache2/envvars`), donde estarán todas aquellas variables que se pueden modificar y que afectan al funcionamiento del servidor de páginas web. Una de las más importantes, desde el punto de vista de la seguridad, es el usuario con el que se ejecuta la aplicación Apache. Antiguamente, si el Apache se instalaba desde el usuario *root*, este era con el que se ejecutaba el servidor después, por lo tanto, cuando se explotaba una vulnerabilidad y se obtenía el control de la aplicación Apache, se podía tener acceso al sistema como usuario *root*.

Actualmente, durante la instalación de la aplicación Apache se crea un usuario y un grupo que se dedican únicamente a ejecutar la aplicación. El usuario y grupo son el www-data, y se puede comprobar que realmente se está usando este usuario y este grupo en el fichero de variables `/etc/apache2/envvars`, donde tiene que estar declarado el usuario y el grupo correspondiente:

```
export APACHE_RUN_USER=www-data
export APACHE_RUN_GROUP=www-data
```

4.2. Servidor de FTP a GNU/Linux

El servidor FTP permite compartir de manera rápida ficheros entre diferentes equipos ubicados en la red. Es un protocolo que no es seguro, puesto que la transferencia de los ficheros no está cifrada y, por lo tanto, se pueden obtener si alguien tiene monitorizada la red y puede leer todo lo que pasa por ella.

Si lo que se necesita es poder compartir ficheros entre usuarios de la misma entidad, lo mejor es instalar el servicio SSH/SSL en el servidor de ficheros, y hacer así que la compartición de estos archivos sea cifrada y que únicamente los usuarios validados puedan ver el contenido del servidor. Con un cliente de SSH podremos iniciar sesión en el servidor, y con un cliente de SFTP se podrá entrar en el servidor y obtener todos los archivos necesarios.

Pero si lo que se necesita es tener un tipo de repositorio, donde todo el mundo pueda entrar y obtener los ficheros que ponemos en él, entonces se tiene que optar por tener un servidor FTP, que hará que no sea necesaria la validación de los usuarios dentro de este, haciendo mucho más ágil la transferencia de ficheros.

4.2.1. Instalación del servidor de FTP

El protocolo FTP es uno de los protocolos más antiguos de la familia TCP/IP. Una prueba de esto es la multitud de servidores FTP que hay. En esta sección mostraremos la instalación del VSFTPD.

Miramos la descripción:

```
root# apt-cache search vsftpd
vsftpd - lightweight, efficient FTP server written for security
root#
```

Para hacer la instalación, tenemos que ejecutar la orden siguiente:

```
root# apt-get install vsftpd
```

Una vez acabada la breve instalación, configuramos el servidor.

4.2.2. Configuración del servidor FTP a GNU/Linux

El fichero de configuración del servidor FTP es `/etc/vsftpd.conf`. Editamos este archivo y descomentamos, si no lo están ya, las líneas siguientes:

```
# Allow anonymous FTP? (Beware - allowed by default if you comment this out).
anonymous_enable = YES

# Uncomment this to allow local users to log in.
local_enable=YES
```

El primer parámetro nos permite hacer un FTP de tipo anónimo, es decir, que no hace falta tener usuario en el servidor, y el segundo permite a los usuarios que tienen cuenta en el servidor acceder por FTP. Si no queremos que se pueda acceder de manera anónima, se tiene que poner el símbolo `#` para comentar esta línea.

Si queremos modificar el mensaje de bienvenida que le sale a los usuarios cuando se conectan por FTP, tenemos que modificar el contenido del parámetro `ftp_banner` y poner el mensaje:

```
# You may fully customise the login banner string:
ftpd_banner = New welcome to blah FTP service.
```

Con la configuración hecha hasta ahora ya habría suficiente, porque los usuarios, cuando se autentican, acceden a su `home_directory`, pero no hay nada que les impida navegar por el resto de directorios del servidor. Si queremos que nuestros usuarios no puedan acceder a ningún directorio que no sea `home_directory`, tenemos que descomentar el parámetro siguiente:

```
# You may restrict local users to their home directories. See the FAQ for
# the possible risks in this before using chroot_local_user or
# chroot_list_enable below.
chroot_local_user=YES
```

Si queremos dejar que nuestros usuarios no solamente bajen archivos de su directorio sino que también pongan datos, tenemos que configurar los parámetros siguientes:

```
# Uncomment this to enable any form of FTP write command.
write_enable = YES

# By default the server will pretend to allow ASCII mode but in fact ignore
# the request. Turn on the below options to have the server actually do ASCII
# mangling on files when in ASCII mode.
# Beware that on some FTP servers, ASCII support allows a denial of service
# attack (DoS) via the command "SIZE /big/file" in ASCII mode. vsftpd
# predicted this attack and has always been safe, reporting the size of the
# raw file.
# ASCII mangling is a horrible feature of the protocol.
ascii_upload_enable = YES
ascii_download_enable = YES
```

Podemos limitar el número de conexiones activas simultáneas, tanto desde las IP como de los clientes con las opciones siguientes, así será más seguro, puesto que únicamente permitimos que un usuario se conecte desde una IP.

```
# If vsftpd is in standalone mode, this is the maximum number of clients
# which may be connected. Any additional clients connecting will get
# an error message.
max_clients=5

# If vsftpd is in standalone mode, this is the maximum number of clients
# which may be connected from the same source internet address. A client
# will get an error message if they go over this limit.
max_per_ip=1
```

Podemos limitar qué usuarios no se pueden autenticar en el servidor TFP, así nos aseguramos de que estos usuarios no podrán entrar nunca. Por ejemplo estos:

```
root# cat /etc/ftpusers
# /etc/ftpusers: list of users disallowed FTP access. See ftpusers(5).

root
daemon
bin
sys
sync
games
man
lp
mail
news
uucp
nobody
www-data
root#
```

El directorio por defecto que crea para los usuarios anónimos está ubicado en `/srv/ftp`, y es aquí donde se tendrán que poner todos los ficheros que se quieran compartir directamente.

Hay muchas más opciones de configuración, como limitar el ancho de banda para usuarios, para las conexiones anónimas, hacer que los usuarios en lugar de ser del sistema estén en una base de datos, etc. Estas opciones nos permiten configurar el servidor de manera algo más segura. Si queremos conocer estas opciones, tenemos que consultar:

```
root# man 5 vsftpd.conf
```

Finalmente, solo falta indicar cómo arrancar el servidor. Cuando hemos hecho la instalación, se ha creado un archivo de arranque en `/etc/init.d` denominado `vsftpd`. Para cargar todas las nuevas opciones hay que parar y arrancar el servidor ejecutando las órdenes siguientes:

```
root# /etc/init.d/vsftpd restart
```

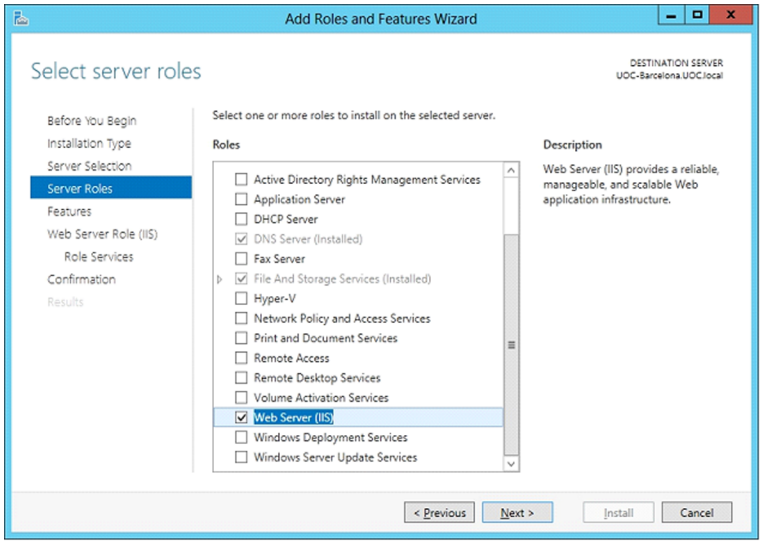
4.3. Servidor web y FTP en Windows Server 2012

4.3.1. Servidor de información de Internet

IIS es el conjunto de servicios de Internet que proporciona Windows Server 2012. Entre estos servicios, está el servidor de páginas web y el de transferencia de ficheros para protocolo FTP, y también el apoyo para páginas dinámicas con ASP o ASP.NET, el servicio de protocolo simple de transferencia de correo (SMTP), el servicio de protocolo de transporte de noticias de red (NNTP), la

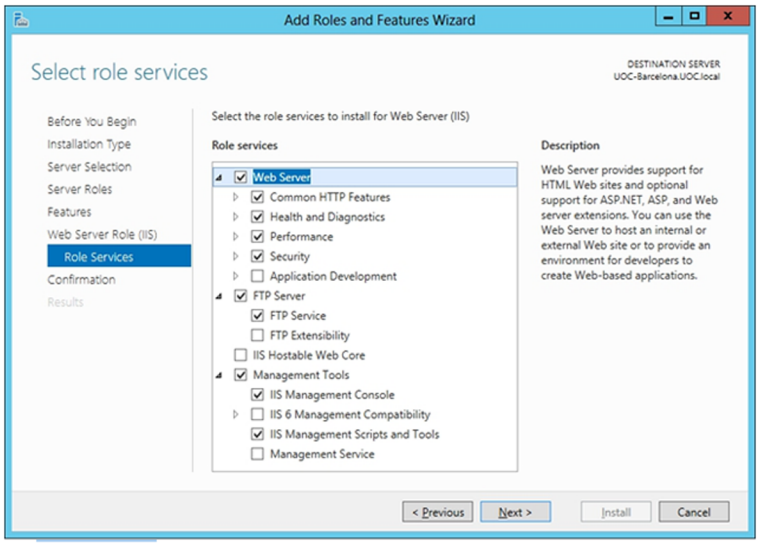
impresión mediante protocolo HTTP (protocolo de impresión por Internet o *Internet printing protocol*, *IPP*), las extensiones de transferencia inteligente de ficheros (BITS) y servicios de reproducción en tiempo real o *streaming* de medios Microsoft Windows Media (difusión de presentaciones multimedia de alta calidad por Internet).

Instalación IIS



IIS no se instala por defecto y por lo tanto, si se quiere instalar, se tiene que hacer con el configurador del servidor, instalando el rol correspondiente al IIS. Una vez seleccionado, nos aparecerá una página para incluir todos los servicios que se quieran tener en el servidor, como puede ser el FTP.

Servicios a instalar



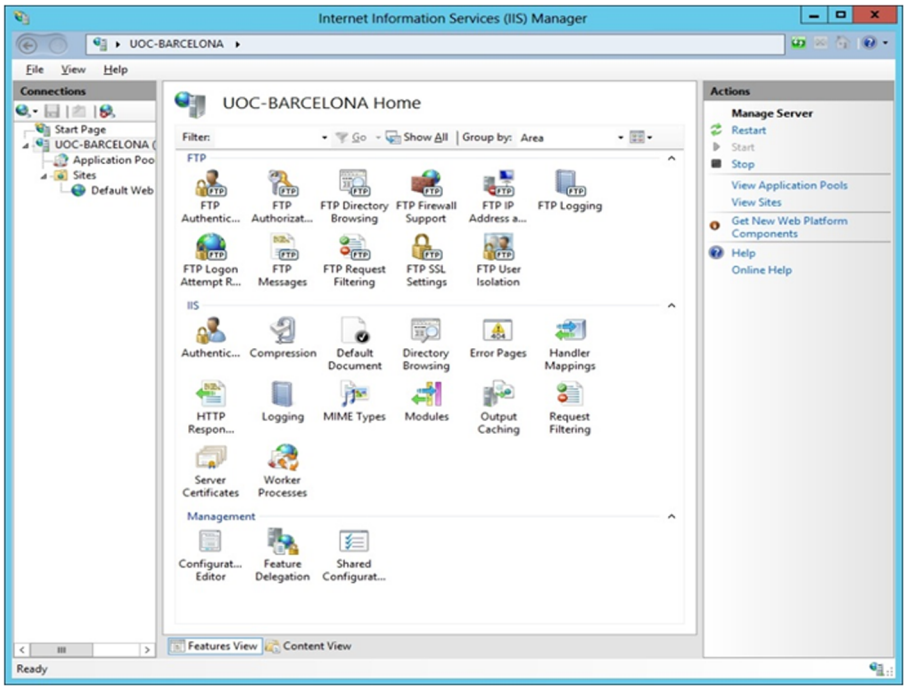
Es importante seleccionar solo los servicios o componentes de IIS que son necesarios, puesto que cuantos más servicios hay instalados, más grande es la superficie de ataque de nuestro servidor y, por lo tanto, más grande es la pro-

babilidad de sufrir un ataque de seguridad. Servicios como FTP, SMTP o NNTP, que no son muy habituales, vale más no instalarlos salvo que sean necesarios para la organización.

La instalación de IIS crea en el disco duro, en el cual ha instalado el sistema operativo, una estructura de carpetas dentro del directorio `Inetpub`. Dentro de este directorio encontramos dos carpetas (`wwwroot` y `ftproot`) que, como indica su nombre, son la raíz de los directorios públicos por defecto de web y FTP.

Para configurar y administrar la estructura web y FTP, utilizamos la herramienta "Administrador de Internet Information Services", que encontraremos dentro del menú de herramientas del administrador del servidor. Esta herramienta muestra de una manera muy rápida y gráfica todos los componentes instalados y cómo se tienen que configurar.

Configuración IIS



Configuración de un sitio web

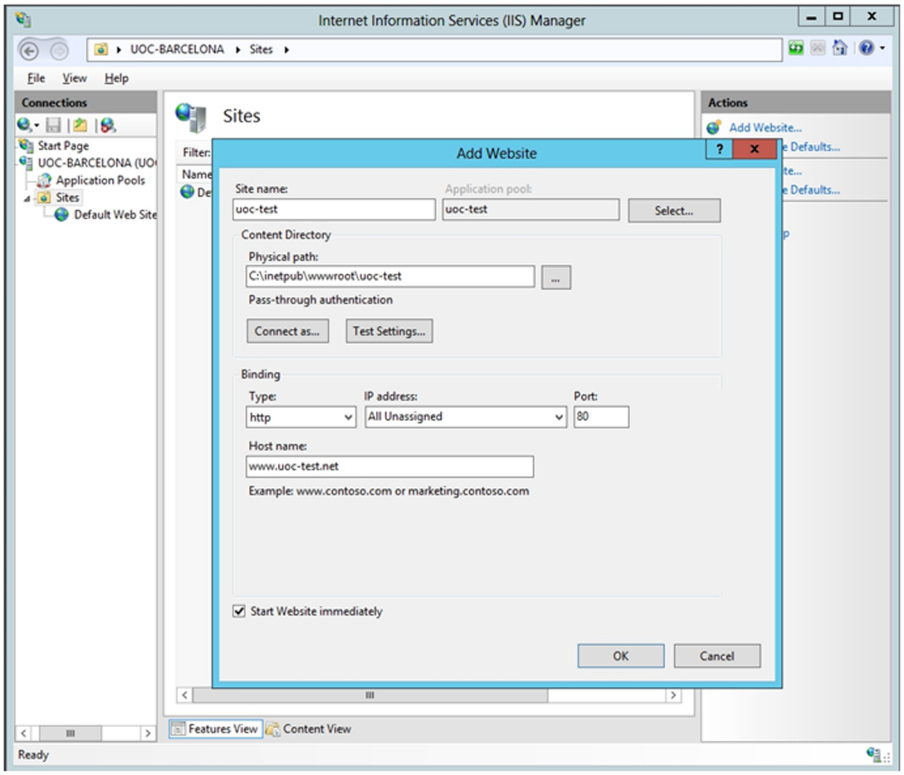
IIS crea un sitio web predeterminado situado en `Inetpub\wwwroot`. Podemos crear otros sitios web adicionales en otros directorios del disco duro del servidor, pero solo un sitio web puede escuchar sobre un puerto TCP a la vez.

Para crear un sitio web, seleccionamos el elemento *sites* que contiene todos los lugares web del servidor, que está dentro del servidor local y, con el botón derecho, abrimos el menú contextual y seleccionamos la opción "Añadir sitio web". El asistente de creación de lugares nos solicita la información necesaria para crearlos: primero, una descripción del contenido del sitio web y, a continuación, la dirección IP y el puerto TCP por el cual se accederá. El puerto por

defecto es el 80 y, aunque se puede cambiar, esto implica que los usuarios no podrán acceder directamente al lugar web desde el navegador si no saben el puerto concreto que utiliza el servicio.

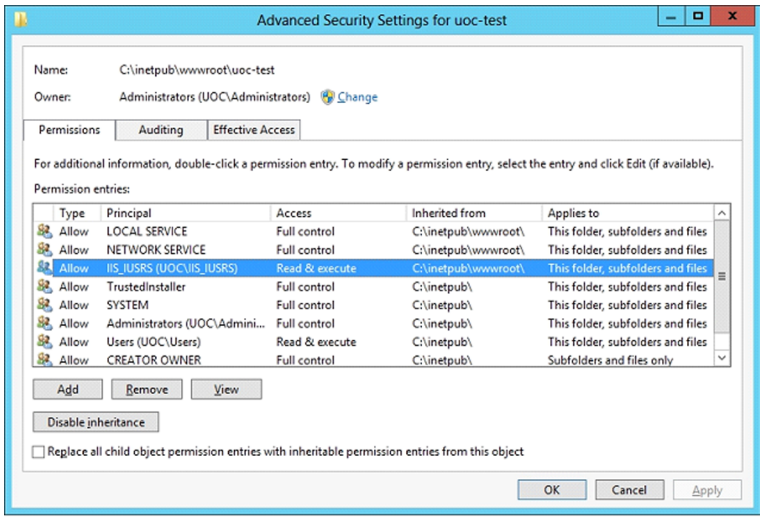
En la figura siguiente podemos ver cómo también se puede configurar para que sea una página segura usando HTTPS en lugar de HTTP. En este caso se tendrá que seleccionar un certificado validado por una entidad certificadora, como se ha visto en el apartado anterior.

Añadiendo un nuevo sitio



Escogeremos también la ruta de acceso raíz al sitio web, a partir de la cual se sitúan el resto de los archivos del sitio web. Con esto ya se crea la infraestructura dentro del servidor para tener una página web. Podemos mirar que los usuarios que accedan a la web lo hagan con los pertinentes permisos, y lo podremos ver en las propiedades del nuevo sitio que acabamos de configurar. Salvo que sea necesario algún otro permiso, los únicos que tienen que estar seleccionados para que sea más seguro son el de "Lectura" y el de "Ejecución" si se trata de páginas ASP o ASP.NET. Es importante mirar esto y eliminar a todos los usuarios que no tienen que tener acceso de escritura, y dejar únicamente a los que sí lo necesitan.

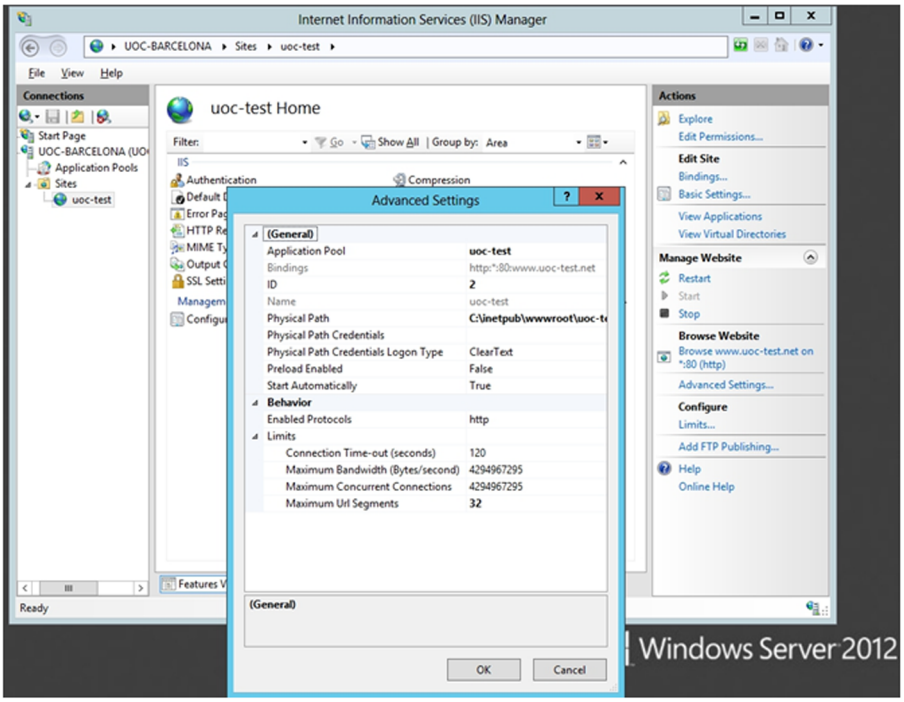
Permisos por defecto



Una vez tenemos el sitio web configurado inicialmente, podemos añadir carpetas y ficheros dentro de la carpeta raíz de este sitio. Estas carpetas y estos ficheros se pueden explorar desde la lista de la izquierda y el contenido que tienen se puede visualizar a mano derecha. También podemos añadir directorios virtuales, que están en otras carpetas del disco duro (no dentro de la carpeta raíz) o en otros equipos. El asistente pide un alias para identificar el directorio virtual, la localización concreta y los permisos que tiene asignados, se tiene que tener cuidado con estos permisos, pues son directorios externos al árbol del sitio web.

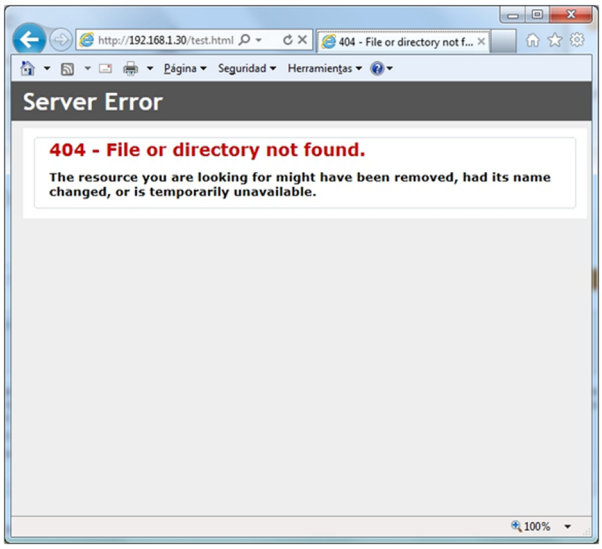
Podemos modificar también la configuración avanzada, donde se puede modificar el ancho de banda que se quiere ofrecer, el número máximo de conexiones, el *time-out*.

Configuraciones avanzadas



Uno de los aspectos a tener en cuenta es la información que se da de manera automática a los atacantes. La versión del sistema operativo y del IIS es importante minimizarla o incluso anularla. Uno de los aspectos a tener en cuenta es cambiar los mensajes de error que devuelven los servidores web, así como las bases de datos. Por lo tanto, se puede cambiar la página de los errores para que sean las que venden por defecto.

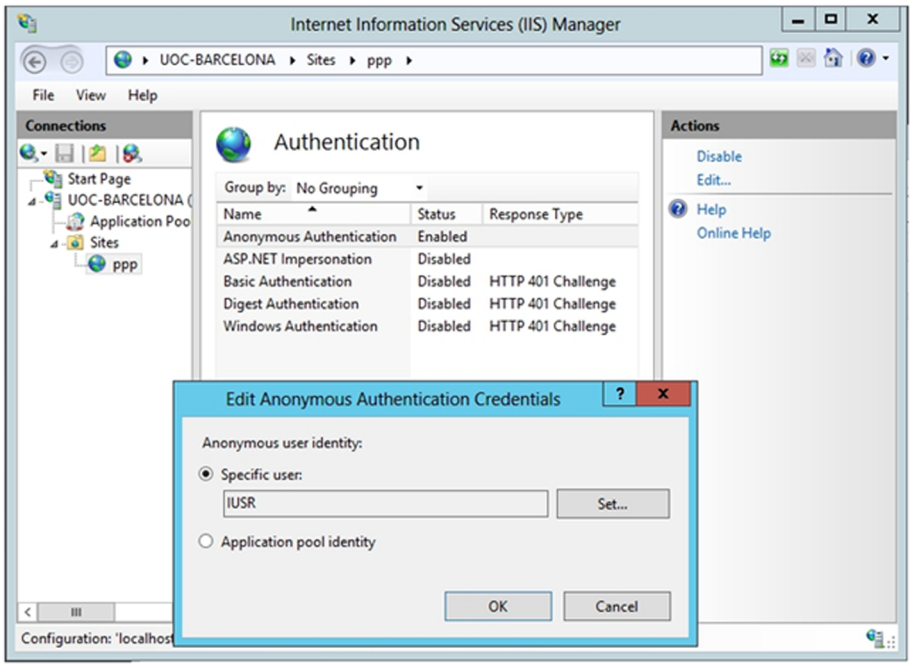
Página por defecto del error no encontrado (*not found*)



4.3.2. Mecanismos de autenticación

Hay varios métodos de autenticación en IIS, los encontraremos en el rol del IIS, dentro de la rama de seguridad en el momento de instalar el rol. Podemos cambiar el mecanismo de autenticación de un elemento desde la ventana del sitio web al enlace de autenticación.

Configuración acceso a la página web



Como podemos ver, es importante que el usuario que ejecuta las páginas web de los usuarios anónimos, es decir, de los que no se validan en el servidor, lo haga con el usuario IUSR del sistema, puesto que tiene únicamente privilegios de lectura y ejecución de páginas web y no tiene ningún privilegio sobre el sistema. Esto dará más seguridad al sistema en el caso de explotar una vulnerabilidad en el servidor IIS.

La autenticación anónima proporciona acceso a las áreas públicas del sitio web sin solicitar el nombre de usuario ni la contraseña. Se asigna el usuario a la cuenta **IUSR**, que está incluido en el grupo de invitados, sometido a unas restricciones de seguridad impuestas por los permisos de acceso a los directorios y carpetas de la web. Si hacemos clic sobre "Modificar", podemos modificar la cuenta que se asocia al usuario anónimo (por defecto, IUSR). Si no es necesario el uso del acceso anónimo, se recomienda desactivarlo, puesto que es un punto de entrada al sistema que puede provocar problemas de seguridad, pero esto hará que la página web no sea accesible si no se dispone de un usuario válido del sistema.

También se disponía además de otros métodos de autenticación:

- Autenticación básica. Se solicita un nombre de usuario y una contraseña antes de entrar en la web. El usuario y la contraseña tienen que corresponder a una cuenta de Windows válida, es decir, que solo podrán acceder a la web los usuarios internos del sistema. Las contraseñas se transmiten por la red sin cifrar, de manera que no se recomienda este mecanismo salvo que sea necesario. En este caso, se recomienda acompañar este tipo de autenticación con el nivel *de sockets* seguro (SSL).

- Autenticación implícita. Tiene las mismas características que la autenticación básica, pero el nombre de usuario y la contraseña se transmiten mediante un proceso de *hashing*. Este tipo de autenticación solo está disponible en servidores que pertenecen a un mismo dominio.

- Autenticación integrada de Windows. Es un mecanismo seguro de autenticación, puesto que sí que se protege el nombre de usuario y la contraseña. La información de autenticación de un usuario es la utilizada para acceder al equipo cliente desde el cual se inicia la petición. Este mecanismo solo es compatible con Microsoft Explorer.

Además, también podemos añadir filtros de entrada y salida al propio sitio web, haciendo que, por ejemplo, solo se pueda acceder a las páginas web desde dentro de la misma institución, o filtrando contenido, etc.

4.3.3. Configuración de un lugar FTP

La configuración de un lugar FTP en IIS es análoga a la de un lugar web. Se crea un lugar FTP inicial con la opción "añadir lugar FTP nuevo" del elemento correspondiente al servidor en el árbol de la izquierda. A la vez, podemos crear carpetas o directorios virtuales dentro del lugar FTP y configurar las propiedades, los permisos y los mecanismos de autorización del mismo modo que en la configuración del sitio web.

Se puede forzar a que el FTP sea a través del protocolo SSH, teniendo así algo más de seguridad. Pero esto hace que el servicio deje de ser anónimo y los usuarios necesiten un usuario en el servidor. E igual que antes, se puede añadir filtros y restricciones de acceso, así como el ancho de banda que se quiere dar a este servicio.

4.3.4. Registro de IIS

Cada acceso al servidor de páginas web o al de FTP crea una entrada en el registro de eventos de IIS, y también los accesos fallidos, páginas no encontradas o usuarios no autorizados. Para habilitar el uso del registro, abrimos la ventana de propiedades del sitio web y en la pestaña "Lugar web" seleccionamos la opción "Logs" y podremos configurar todo lo que se necesite: dónde se guarda, cómo y qué se guarda, etc. A pesar de que si se abre un fichero de logs del IIS se verá que guarda cada una de las peticiones que se hacen al servidor, por

lo que se hace muy difícil el estudio de estos logs, se necesitarán herramientas especializadas en extraer información para poder entender rápidamente lo que está pasado con el servidor IIS.

Seguridad en IIS

Han pasado muchos años desde que Microsoft presentó la primera versión de IIS y ha mejorado el producto a medida que presentaba versiones nuevas. De hecho, la versión incluida en Windows 2000, IIS 5, fue famosa por los célebres agujeros de seguridad que tuvo. Por lo tanto, tuvo una gran repercusión gracias a los virus o gusanos que explotaban las vulnerabilidades que tenía. Después de todo lo que pasó, Microsoft decidió empezar de cero y desarrollar IIS 6.0 desde los fundamentos. No se aprovechó ni una línea de código. Los resultados posteriores, con las últimas versiones más nuevas, son un producto que está preparado contra los ataques más comunes, que tolera fallos o caídas del servicio y que es autorrecuperable, puesto que los procesos son capaces de reciclarse por sí mismos.

Además, se mejoró la arquitectura interna, de tal manera que, en caso de ataque y caída del servicio IIS, el servidor no quede comprometido gracias a un sistema de aislamiento de aplicaciones, basado en un conjunto de aplicaciones (*pools*) que se ejecutan con *Worker Process* separadas por cada aplicación.

Se tiene que tener cuidado con los carácter Unicode, puesto que se pueden codificar los caracteres (`/`) y (`..`) dentro de la URL con caracteres Unicode, de forma que al evaluar el servidor IIS los permisos de las carpetas no encuentren ningún problema, pero al traducir los caracteres Unicode se puedan formar direcciones que permitan acceder a recursos del sistema.

Por lo tanto, para aumentar la seguridad de IIS hay que tener en cuenta ciertas recomendaciones, algunas de las cuales ya hemos comentado:

- Eliminar el acceso anónimo si no es necesario o, si lo es, delimitarlo a los directorios virtuales que lo requieran.

- Utilizar SSH con la autenticación.

- No instalar los servicios de IIS que no sean necesarios.

- Eliminar los ficheros de ejemplo que se instalan junto con IIS.

Es muy importante, sobre todo, tener el sistema actualizado con los últimos "parches" que genera Microsoft a medida que se encuentran fallos en la seguridad de IIS o, en general, del sistema. En el capítulo siguiente trataremos con más detenimiento el tema de las actualizaciones.

4.4. Análisis de riesgos y prevención

4.4.1. Web

Durante los últimos años, cualquier empresa que se quiere dar a conocer mínimamente está conectada a Internet y tiene un sitio web, en el cual la empresa trata de mostrar una imagen corporativa o vender algún producto. Si lo que queremos es que la gente conozca la empresa (o institución), tenemos que permitir que el sitio web sea accesible desde cualquier parte. Esto implica que los puertos TCP donde se ejecuta un servidor web no tienen que tener ningún tipo de restricción de acceso. Ahora bien, tenemos que ser conscientes de que el hecho de tener una puerta abierta al mundo también quiere decir tener una puerta abierta a los posibles atacantes. Hay muchos ataques que nos pueden llegar por los mismos puertos por donde escucha el servidor web.

Los ataques de modificación de los lugares web son muy "vistosos", sobre todo si se trata de una empresa más o menos grande o entidad conocida por mucha gente. La noticia de que un lugar web de estos ha sido pirateado corre por Internet en cuestión de minutos. Incluso, hay recopiladores de sitios web pirateados. En alguno de estos casos, dependiendo de la importancia de la empresa, la noticia puede llegar incluso a la prensa. En estos casos, la imagen de la empresa o entidad que ha sido atacada se puede ver gravemente dañada. Hay otro tipo de ataques que, aunque no son tan frecuentes, resultan más peligrosos. Este otro tipo de ataque no pretende modificar el sitio web, sino que, mediante una vulnerabilidad del servidor, pretende acceder a la máquina, al servidor. La mayoría de estos ataques tienen éxito debido a una configuración errónea o defectuosa del servidor, aunque también es posible que sean debidos a vulnerabilidades (errores o *bugs*) del servidor mismo y, muchas veces, corregibles actualizando el sistema (*Windows Update*).

En las empresas grandes, los servidores web suelen ser muy complejos: alta disponibilidad, redundancia de servidores, balanceo de carga, gestión de contenidos dinámicos, etc. Estos sistemas son complejos de administrar y securizar y es posible que, si no se lleva a cabo de manera correcta, se produzcan errores de configuración. Las empresas pequeñas muchas veces utilizan un servidor simple que requiere muy poco mantenimiento y es fácil de administrar, pero, a su vez, estos sistemas simples no suelen tener sistemas de seguridad muy complicados.

Cualquier analizador de vulnerabilidades que podamos ejecutar contra los sistemas es capaz de mostrar mucha información que nos puede resultar muy útil a la hora de reforzar la seguridad de nuestros servidores.

Un ejemplo de ejecución de un analizador contra un servidor web es el siguiente:

```
root# nmap -sV 192.168.1.30
Starting Nmap 5.00 ( http://nmap.org ) at 19:24 CEST
Interesting ports on 192.168.1.30:
Not shown: 983 filtered ports
PORT   STATE SERVICE   VERSION
53/tcp  open domain    Microsoft DNS
80/tcp  open http     Microsoft IIS webserver 8.0
88/tcp  open kerberos-sec Microsoft Windows kerberos-sec
135/tcp  open msrpc    Microsoft Windows RPC
139/tcp  open netbios-ssn
389/tcp  open ldap
445/tcp  open netbios-ssn
464/tcp  open kpasswd5?
593/tcp  open ncacn_http  Microsoft Windows RPC over HTTP 1.0
636/tcp  open tcpwrapped
3268/tcp open ldap
3269/tcp open tcpwrapped
49154/tcp open msrpc    Microsoft Windows RPC
49155/tcp open msrpc    Microsoft Windows RPC
49157/tcp open ncacn_http  Microsoft Windows RPC over HTTP 1.0
49158/tcp open msrpc    Microsoft Windows RPC
49175/tcp open msrpc    Microsoft Windows RPC
MAC Address: 00:21:5A:FD:00:00 (Hewlett Packard)
Service Info: OS: Windows

Service detection performed. Please report any incorrect results at http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 66.10 seconds
```

Como apreciamos, los analizadores nos pueden mostrar las versiones de las partes del servidor (*http*, *dns*, *ldap*, *ssl*, *php*, *perl*, etc.) que tenemos instaladas. Incluso nos pueden informar de directorios que tenemos abiertos en el servidor web. Cuando el atacante consigue esta información, solo tiene que buscar entre las vulnerabilidades de nuestros servicios instalados.

Un primer paso para evitar estos problemas consiste en eliminar del servidor cualquier directorio o CGI que se instale de manera automática (por defecto) en el servidor. Este tipo de directorios suelen ser de ejemplo o de documentación, que no son especialmente críticos. El caso de las CGI es diferente, puesto que hay alguno de estos *scripts* que puede llegar a abrir agujeros importantes en el servidor.

El paso siguiente para añadir seguridad a nuestro servicio es deshabilitar el *Directory Indexing* de nuestro servidor. Este parámetro permite hacer una lista del contenido de un directorio en caso de no encontrar un fichero `index.html`. En principio, en el `DirectoryRoot` solo tiene que haber los ficheros que son imprescindibles para visualizar correctamente las páginas web de nuestro servidor, pero la mayoría de veces encontramos archivos comprimidos que son

los códigos fuente de alguna aplicación que se ejecuta en la página web. Si los atacantes saben que existen estos ficheros, los pueden bajar y obtener una información muy útil a la hora de atacar el servidor. La mejor manera de evitar este tipo de ataques es que no haya nada que nos pueda comprometer bajo el `DirectoryRoot`, pero si hay algo (por necesidades del sistema), tenemos que sacar los permisos de lectura (para que no lo pueda leer el usuario que ejecuta el servidor web) y deshabilitar el *Directory Indexing* para que nadie sepa que estos ficheros están aquí.

El punto siguiente que se tiene que tener en cuenta es el usuario con el cual se ejecuta el servidor web. Es muy desaconsejable que este usuario sea `root`. Tenemos que usar un usuario sin ningún privilegio (por ejemplo, `www-data`). Para evitar que nos modifiquen los ficheros `.html`, el usuario `www-data` solo tiene que tener los privilegios de lectura y ejecución (solo cuando sea necesario) sobre todos los ficheros que hay por debajo del `DocumentRoot`, de este modo no podrá reescribir su contenido.

Para acabar, es recomendable usar el protocolo HTTPS para llevar a cabo las tareas de autenticación de usuarios y de acceso a información confidencial por la red.

4.4.2. FTP

El FTP es una herramienta de transferencia de ficheros muy práctica. Tenemos que saber, sin embargo, en qué entornos podemos utilizar esta aplicación. Si queremos ofrecer un servicio de bajada de algún tipo de software, el FTP es nuestra herramienta, puesto que lo hacemos con el FTP anónimo. El cliente de bajada es el mismo navegador web, cosa que ofrece mucha flexibilidad.

Si queremos que nuestros usuarios se puedan conectar desde casa para acceder a su espacio de disco y bajar los documentos, entonces el FTP, a pesar de que satisface las necesidades de los usuarios, no es la herramienta adecuada, puesto que no tiene ningún tipo de cifrado del canal; por lo tanto, la autenticación de los usuarios y la transferencia de ficheros se hace en plano, es decir, sin cifrar.

Hoy en día encontramos servidores de FTP que cifran el canal de comunicaciones. El protocolo que cifra la comunicación se denomina SFTP. Los servidores de SSH contienen un subsistema que permite hacer transferencias de ficheros.

Además, el FTP es, tal como hemos comentado, un protocolo decano dentro de las comunicaciones TCP/IP, y esto implica que es muy conocido y, por lo tanto, también está muy estudiado. Los servidores antiguos de FTP tienen muchos agujeros para entrar en la máquina. Por suerte, las versiones más modernas de estos servidores ofrecen medidas de seguridad mucho mejores.

Es importante que, a pesar de que el FTP permite transferencias en los dos sentidos, solo se puedan bajar ficheros del servidor. Potencialmente es muy peligroso permitir escribir en el servidor, puesto que no sabemos quién hay al otro lado de la comunicación y, por lo tanto, no podemos conocer sus intenciones.

5. Protección de puertos

A lo largo de este material hemos hablado mucho sobre los puertos TCP. En esta sección veremos cómo tenemos que cerrar determinados puertos, cómo los tenemos que abrir y cómo los tenemos que proteger.

Sin embargo, antes de empezar a hablar de ello, tenemos que saber qué puertos tenemos abiertos en nuestra máquina. Hay ciertos puertos que tienen que estar abiertos, y son los que coinciden con los servicios que ofrecemos, pero en algunas versiones de sistemas operativos se mantienen abiertos otros. ¿Cómo podemos saber cuáles son los que tenemos abiertos?

5.1. Protección de puertos en GNU/Linux

En un módulo anterior hemos explicado cómo se tiene que instalar la herramienta de comprobación nmap. Para ver qué puertos tenemos abiertos, tenemos que ejecutar la orden:

```
root# nmap host -sV
```

Como vemos en la figura siguiente, esta instrucción nos muestra los puertos que tenemos abiertos y qué aplicación hay asociada a cada uno de ellos.

```
root# nmap -sV localhost

Starting Nmap 5.00 ( http://nmap.org ) at 2013-04-01 19:58 CEST
Interesting ports on localhost (127.0.0.1):
Not shown: 989 closed ports
PORT   STATE SERVICE VERSION
21/tcp  open ftp  vsftp
25/tcp  open smtp   Postfix smtpd
80/tcp  open http   Apache httpd 2.2.16 ((Debian))
110/tcp open pop3   Courier pop3d
111/tcp open rpcbind
143/tcp open imap   Courier Imapd (released 2010)
443/tcp open ssl/http Apache httpd 2.2.16 ((Debian))
631/tcp open ipp   CUPS 1.4
993/tcp open ssl/imap Courier Imapd (released 2010)
995/tcp open ssl/pop3 Courier pop3d
3306/tcp open mysql  MySQL 5.1.66-0+squeeze1

Service detection performed. Please report any incorrect results at http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 23.68 seconds
```

Para cerrar un puerto, normalmente tenemos que parar el servicio que hay escuchando en este puerto. La mayoría de los servicios están en `/etc/init.d`. Una vez estamos en este directorio, tenemos que ejecutar el fichero que tiene el mismo nombre que el servicio que queremos parar, seguido de la palabra `stop`. Por ejemplo, si queremos parar `portmap` –un servicio encargado de escuchar las llamadas a procedimientos remotos o *remote procedure calls* (RPC) que se usa, entre otras cosas, para servicios de sistemas de archivos de red o *network file system* (NFS) y sistemas de información de red o *network information service* (NIS)–, tenemos que hacer lo siguiente:

```
root# /etc/init.d/portmap stop
```

Para abrir un puerto tenemos que poner en marcha el servidor que se ejecuta en este puerto. Para hacerlo, seguimos los mismos pasos que para pararlo, pero ejecutamos el fichero seguido de la palabra `start`:

```
root# /etc/init.d/portmap start
```

Los puertos, sin embargo, no se abren y se cierran todos de este modo. Hay determinados puertos que se controlan desde el superservidor y por lo tanto no se pueden modificar ni desde la cuenta de superusuario.

5.1.1. Xinetd

En la mayoría de las versiones actuales de GNU/Linux, entre los paquetes que tienen disponibles está el `Xinetd`. Esta aplicación también se denomina superservidor. La función principal de esta aplicación consiste en escuchar muchos puertos y, cuando oye una petición para uno de estos puertos, despierta el demonio que está asociado a ella. De este modo, tenemos una sola herramienta que escucha todos los puertos en lugar de tener una aplicación para cada puerto. El `Xinetd`, a diferencia de su antecesor (`Inetd`), tiene por defecto todos los puertos cerrados. Si tenemos instalado el `Inetd`, es muy recomendable (por motivos de seguridad interna de la aplicación) que actualicemos el superservidor en la versión `Xinetd`.

Si queremos instalar el Xinetd, tenemos que ejecutar la orden siguiente:

```
root# apt-get install xinetd
```

Para configurar este servicio se tiene que editar el fichero `/etc/xinetd.conf`. Este fichero indica qué servicios arrancan cuando se pone en marcha el servidor. La configuración de cada uno de los servicios se encuentra en el directorio `/etc/xinetd.d`.

Si se quiere habilitar un servicio, se tiene que editar el fichero correspondiente y modificar el parámetro siguiente:

```
disable = yes
```

Y modificarlo de una de las maneras siguientes:

```
disable = no
# disable = yes
```

Un fichero de configuración de servicios (el de telnet, por ejemplo) tiene un aspecto parecido al siguiente:

```
service telnet
{
flags = REUSE socket_type = stream wait = no
user = root
server = /usr/sbin/in.telnetd log_on_failure += USERID disable = no
}
```

Una vez hemos hecho los cambios en los ficheros, tenemos que reiniciar el demonio `Xinetd`. Para hacerlo, ejecutamos la orden siguiente:

```
root# /etc/init.d/xinetd restart
```

5.1.2. Restricción de puertos

Además de abrir los puertos, podemos restringir su acceso. Este acceso se controla con los ficheros `/etc/hosts.allow` y `/etc/hosts.deny`. En el primero se especifican todas las máquinas que tienen acceso, y en el segundo, todas las máquinas que no tienen que tener acceso. Estos dos ficheros nos permiten diseñar diferentes "políticas" de acceso a las máquinas, desde unas muy permisivas (permitirlo a pesar de denegar algún servicio), hasta políticas muy restrictivas (denegarlo a pesar de permitir solo algunos servicios).

Para usar estos ficheros hace falta que los servidores (de las diferentes aplicaciones) sean compatibles con TCP Wrapper. TCP Wrapper es el servicio que verifica el origen de las conexiones con su base de datos `/etc/hosts.allow` (equipos autorizados) y `/etc/hosts.deny` (equipos a los cuales se deniega la conexión).

La sintaxis de los dos ficheros es la misma: `Servicio: máquinas`.

El servicio es el nombre del demonio o del servidor (`sshd`, `telnet`, `FTP`, etc.), mientras que el nombre de la máquina puede ser una dirección IP, el nombre o un rango de red. Si queremos especificar más de una máquina, tienen que salir separadas por comas.

Estos ficheros admiten también un número reducido de palabras:

1) `All`: todo (se puede referir tanto a servicio como a máquinas).

2) `Local`: las máquinas del dominio (en el nombre no tiene que haber un punto).

3) `Paranoid`: solo las máquinas dadas de alta en el DNS.

4) `Known`: máquinas conocidas (están en el fichero `/etc/hosts`).

5) `Unknown`: máquinas desconocidas.

Un ejemplo de política muy restrictiva es el siguiente:

```
Host.allow
SSHD: LOCAL

Host.deny
ALL: ALL
```

Con esta configuración solo permitimos acceder al servidor de SSH en las máquinas locales.

Actualmente, hay pocas máquinas que configuren el acceso a sus servicios mediante estos dos ficheros. La mayoría de los controles de acceso se hacen con cortafuegos, puesto que permiten tener más versatilidad en las configuraciones de redes y en el tipo de acceso.

Según cuál sea nuestra configuración de la red, podemos tener un cortafuegos que permita el acceso a la red a determinadas máquinas y después podemos tener un segundo nivel de protección en cada uno de los servidores. Este segundo nivel se implementa con la aplicación `iptables` (en la mayoría de los casos, esta aplicación se usa para implantar un cortafuegos software) para definir, entre todas las máquinas de la red, cuáles tienen acceso a determinados servicios.

También hay otra opción, que es implementar el *port knocking* en el que se tienen los puertos escondidos detrás de una secuencia establecida de comunicación. Esto quiere decir que tendremos los diferentes puertos cerrados desde fuera, pero seguiremos teniendo los servicios escuchando a través del puerto correcto. El puerto *knocking*, al recibir una secuencia válida de puertos, abrirá el puerto correspondiente. Para implementar esta técnica, es necesario instalar el demonio `knockd` y configurar algunas de las llamadas para que abra el puerto correspondiente.

```
root# apt-get install knockd
```

Hace falta configurarlo para que se pueda usar. La configuración la tenemos en `/etc/knockd.conf`, donde tenemos por defecto el protocolo SSH configurado. Se puede ver cómo, si arrancamos el servidor, el puerto del servicio ssh se mostrará cerrado a menos que se haga una llamada previa a los puertos 7000, 8000 y 9000 en este orden. Y se mantendrá abierto hasta que se haga otra secuencia con los puertos 9000, 8000 y 7000.

```
[options]
    UseSyslog

[openSSH]
    sequence  = 7000,8000,9000
    seq_timeout = 5
    command   = /sbin/iptables -A INPUT -s %IP% -p tcp --dport 22 -j ACCEPT
    tcpflags  = syn

[closeSSH]
    sequence  = 9000,8000,7000
    seq_timeout = 5
    command   = /sbin/iptables -D INPUT -s %IP% -p tcp --dport 22 -j ACCEPT
    tcpflags  = syn
```

Para arrancar el servicio de `knockd` hay que editar el fichero `/etc/default/knockd` y tal como se muestra, cambiar el parámetro a 1 para que arranque el servicio desde el inicio.

```
root# joe /etc/default/knockd
################################################
#
# knockd's default file, for generic sys config
#
################################################
# control if we start knockd at init or not
# 1 = start
# anything else = don't start
#
# PLEASE EDIT /etc/knockd.conf BEFORE ENABLING
START_KNOCKD=0

# command line options
#KNOCKD_OPTS="-i eth1"
```

Para poder abrir los puertos hay aplicaciones tanto en GNU/Linux como en Windows que gestionan las llamadas a los diferentes puertos para abrir aquel que realmente se quiere tener operativo. El propio servicio `knockd` tiene `knock`, que hace las llamadas a GNU/Linux, y en Windows existe la aplicación `KnockKnock`, que se puede configurar para hacer lo mismo.

A pesar de ser una buena técnica para esconder puertos, a veces resulta poco efectiva, ya que es necesario abrir antes los puertos y además es también muy fácil dejarlos abiertos. Por lo tanto, se tiene que tener cuidado con esta técnica, para no dejar abiertos los puertos.

5.2. Protección de puertos en Windows Server 2012

El servicio web y el servicio FTP de IIS son aplicaciones que reciben y envían datos por un determinado puerto TCP o UDP de la máquina. Hay otros muchos servicios o aplicaciones del propio sistema u otros fabricantes que utilizan diferentes puertos de la máquina para comunicarse.

Como un puerto es un punto de entrada al sistema, es importante desactivar los puertos que no necesitaremos en el servidor. Para saber qué puertos hay activos actualmente en el sistema, podemos utilizar la herramienta de consola de órdenes `netstat`. Para hacerlo, abrimos una ventana del intérprete de órdenes y escribimos lo siguiente:

```
netstat
```

La ejecución de esta orden nos muestra una lista de puertos activos, junto con el protocolo (TCP o UDP) que utiliza y el estado actual.

Una manera de configurar puertos de forma completa es utilizando el cortafuegos que viene con Windows Server 2012 o instalando un cortafuegos hardware o software, que permita, además de filtrar los puertos, filtrar según la IP origen de la petición y configurar también los accesos entrantes y las peticiones que salen del servidor mismo.