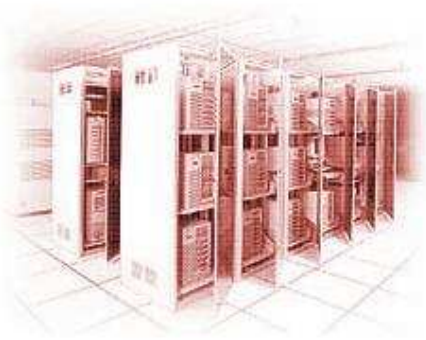
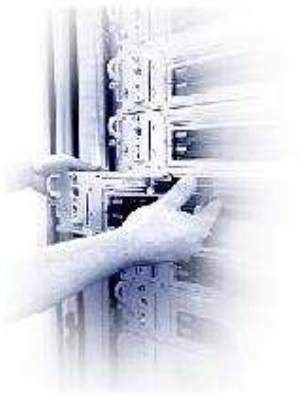
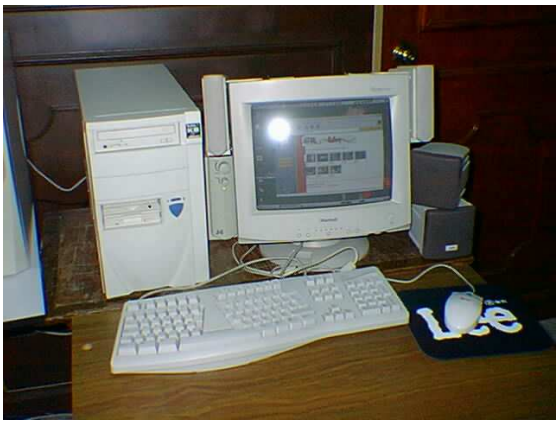
MinServ

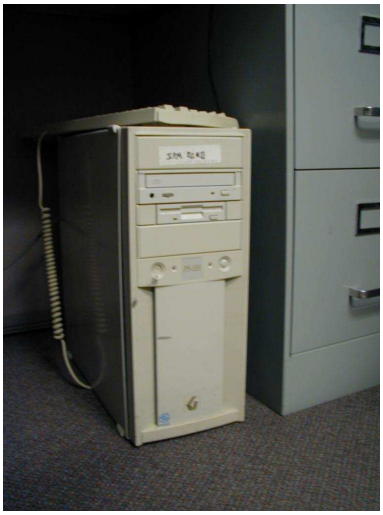
Por que cuando decimos “servidores” los relacionamos con ordenadores muy potentes y bastante grandes como por ejemplo:





Y no un simple ordenador que puede ser el que tienes en tu casa como:





Pues esto es posible gracias a =>

Linux.

Índice

1. Introducción y preparación a nuestro MinServ

1.a. Contexto

Soy un alumno del instituto IES La Marisma, en Huelva, usuario de Linux y Webmaster. Este proyecto consiste en darle una utilidad a un ordenador que tengamos en desuso por culpa de la falta de rendimiento correspondiente a los programas actuales, este ordenador a partir de ahora lo vamos a llamar MinServ, este PC llevará el sistema operativo Linux y sólo y otros programas de software libre.

MinServ pondrá al alcance de los alumnos, profesores y visitantes una página Web con contenidos dinámicos e interactuando con una base de datos para poder tener diferentes usuarios para que accedan a diferentes secciones y con diferentes privilegios dentro de la base de datos, la posibilidad de actualizar la pagina Web desde otro PC en cualquier otro sistema operativo, ya que le vamos a dar soporte para que tenga compatibilidad entre Linux y Windows, nuestro MinServ va a tener un servidor SSH para poder hacer modificaciones en el servidor desde cualquier punto del mundo, ya que por las actualizaciones de la pagina Web tengan problemas y siempre tengan una puerta abierta para la posible configuración desde el exterior y utilizaremos un firewall-gateway para la protección del mismo MinServ, de la red interna que vamos a tener.

Vamos a utilizar Linux, ya que al ser software libre no tendremos que comprar programas, ya que todo el software usado es gratuito y legal. Además, hará el instituto mucho más profesional al utilizar un sistema operativo del nivel de Linux.

1.b. **Justificación**

Mis motivos por haber cogido este tema son por que me atraía la idea de que los alumnos puedan saber sus notas y así llevarán un control sobre ellos mismos y hacerlo a través de una interface amigable, además me gusta mucho Linux y creo que puede ser muy instructivo en mi formación ya que el mercado del software libre esta creciendo y por el simple hecho de darme una satisfacción personal de tener una idea y llevarla a cabo a pesar de las complicaciones que pasé durante el proceso de la realización de MinServ. Este proyecto esta realizado sobre la plataforma Linux que cada día tiene más adaptación en los PC domésticos y pienso que todos deberían saber que existe Linux y que hay alternativas a Windows, con esto no digo que sea mejor ni peor, solo que cada usuario tiene una necesidad a satisfacer y pueda elegir la que le diese mejor resultado. Este trabajo no servirá para demostrar que Linux es muy fácil de usar, porque precisamente usaremos una de las distribuciones más complicadas “Debian”.

Hoy en día creo que poner un sistema operativo como Linux, para el que existen miles de aplicaciones profesionales desde hace tiempo, elevará el nivel de las instalaciones informáticas del centro, ya que abrirá las puertas a muchísimas posibilidades para los alumnos y profesores.

Una de la razones que implanto MinServ con sistema Linux es por que llevo trabajando con él unos 2 años y he comprobado en mas de una ocasión la seguridad y la robustez del sistema comparándolo a otros sistemas operativos, por la cantidad mínima de recursos que utiliza el sistema y sobre todo por el control total que tienes sobre tu PC, ya que sabes en todo momento que procesos estas ejecutando y que rendimiento de la cpu usa, esto es una gran ventaja ya que lo configuras sólo para lo que lo necesitas, ni una aplicación de más y por supuesto ni una de menos.

1.c. Objetivos

Este proyecto tiene de objetivo:

- Dar una información al estudiante al día de sus calificaciones
- Dar una facilidad para adelantar trabajo en casa
- Dar una facilidad al director para corregir datos de alumnos
- Utilizar aplicaciones profesionales a bajo coste
- Promover el uso de Linux y software libre en institutos
- Aprovechar ordenadores viejos para la mejora del centro

MinServ y sus características:

- Que sea fácil de usar
- Que tenga un Servidor Web
- Que el servidor Web tenga soporte en PHP
- Que PHP interactúe con la base de datos MySQL
- Que muestre gráficamente desde una Web
- Que esté integrado con la red Windows
- Que acepte conexiones remotas
- Que esté protegido

1.d.Preparación

1.d.i. Estado inicial de los ordenadores

La red del instituto consta de de varias clases de informática, en ellas hay equipos que por su bajo rendimiento no se pueden utilizar para las aplicaciones que se necesitan para llevar a cabo los estudios adecuadamente.

Hay un ordenador que usa Windows 2003 Server y que gestiona la red de cada clase y todos estos se conectan directamente a varios Switch que van a dos Routers. Cuando integremos nuestro servidor en esta red, lo tendremos en cuenta y nuestro equipo lo conectaremos directamente a una salida de un Router o Switch.

Esta red es de clase C, y su dirección IP es 192.168.X.Z./24 donde la X es el número de la clase y la Z son los equipos dentro de la clase.

1.d.ii. **Método a seguir**

El trabajo práctico está dividido en tres bloques: instalación, servicios básicos y servicios secundarios.

. Instalación: 1º elegiremos la distribución mas adecuada para lo que queremos hacer, 2º instalamos el sistema operativo con la configuración básica para iniciar el SO desde 0 y poderlo modificar todo a nuestro gusto

. Servicios básicos: 3º pondremos el servidor SAMBA y el correspondiente cliente para poder comunicarnos con redes Linux y Windows indiferentemente de la plataforma con la que nos queramos interconectar, 4º instalaremos el servidor Web Apache ya que este servidor es bastante estable y configurable, 5º insertaremos un servidor MySQL en nuestro MinServ para poder insertar los datos de los alumnos y para poder interactuar con el Apache mediante PHP.

. Servicios secundarios: 6º configuraremos el apache para cargarle en modulo el PHP en nuestra Web, añadiremos un servidor ssh para podernos conectar desde cualquier parte, haremos una pagina Web y le introduciremos unos scripts en php y por último le pondremos un firewall y lo configuraremos a nuestras necesidades.

1.d.iii. **Elección del ordenador**

No hace falta un ordenador muy potente para que haga de MinServ ya que en nuestro instituto no va a haber una saturación ya que no van a acceder cientos de ordenadores a nuestros servicios, además, Linux es un sistema operativo que aprovecha perfectamente los recursos de nuestra máquina

Nuestro PC para la inserción de MinServ

- Procesador con tecnología x386
- 16 Mb de memoria RAM
- Disco duro de 4 GB
- CD-ROM y disquetera.
- 2 Tarjetas de red Realtek 8139D
- Pantalla a color, teclado y ratón estándar.

Este ordenador tiene todo lo que necesitamos y más para funcionar como servidor de un instituto, pondremos todos los componentes para la instalación y su configuración, pero una vez terminado, le quitaremos el CD-ROM, la disquetera, el ratón, el teclado y la pantalla, le dejaremos solo la CPU con lo mínimo ya que si queremos configurarlo más tarde, entraremos por ssh desde otro PC y así nos quitaremos más cosas de el medio y será más cómodo trabajar.

Y así lo convertiremos en un servidor óptimo.

2. Configuración del servidor

2.a. Instalación de Linux

2.a.i. Elección de la distribución

Las distribuciones se adaptan a las necesidades de los usuarios y de este modo se puede escoger, ¿para que tener una distribución con paquetes y programas para administración de redes, si el trabajo será para el hogar? En general, se considera que una distribución es mejor a otra cuando el reconocimiento de los usuarios a nivel mundial les otorgue tal distinción.

Como Linux es software libre que puede ser modificado y adaptado por todos, se han creado muchas versiones distintas del sistema operativo, tantas que ni se pueden contar.

En principio, para el uso que le daremos, tendrá que cumplir las siguientes condiciones:

- Seguridad: ¡Muy importante! Ningún sistema operativo es 100% seguro y hemos de estar seguros de que ningún hacker ni ninguna alumno traviesillo pueda acceder a nuestro servidor. Para mantener la seguridad habrá que hacer actualizaciones del sistema muy a menudo.

- Fácil de actualizar: tendremos que tener siempre las últimas versiones de los programas para corregir todos los posibles errores.

- Estabilidad: no queremos que se quede pillado por que es el ordenador central y dejara de dar los servicios que está dando.

- Simplicidad: queremos un sistema que solo tenga los paquetes necesarios para su mayor aprovechamiento del poco rendimiento que vamos a tener, poco pero suficiente, es decir, el entorno gráfico no lo pondremos ya que consume recursos sin necesidad alguna.

Como no tenemos ninguna necesidad extremadamente especial, compararemos sólo las 5 distribuciones más utilizadas. Las versiones analizadas quedan anticuadas en pocos meses, pero la filosofía de los programadores y el tipo de distribución de cada una seguirá siendo el mismo.

Distribución	Origen	CDs	Gestor defecto	Tipo de paquete
Mandrake 10.1	Francia	3	KDE	rpm
Red-Hat Enterprise	USA	7	Gnome	rpm
Suse 9.3	Alemania	5	KDE	rpm
Debian sarge	-	1	-	deb
Gentoo	Usa	1	-	src

Es curioso, pero, de estas cinco, sólo las tres primeras son las realmente conocidas por su facilidad de instalación y uso. Es bueno que usemos una distribución conocida porque así nos costará poco encontrar manuales, ayuda o soporte técnico.

No obstante, vamos a compararlas punto por punto:

- Precio: los precios los vamos a ignorar por que al ser software libre, no vamos a pagar, sino que lo descargamos de Internet, inconvenientes es que no vamos a tener documentación impresa ni soporte técnico.

- Número de CDs: en la mayoría de los casos solo necesitaremos los 3 primeros CDs o el único que haya dependiendo de la distribución

- Versión del kernel: la versión del núcleo del sistema operativo es importante sobre todo cuando tenemos problemas de hardware. Necesitaremos una 2.4.18 o superior.

- Tipo de instalación: como utilizaremos el Linux en modo texto la instalación también será así.

- Tipo de paquetes: para instalar los programas que nos bajemos, lo podemos hacer de diferentes formas:

tar.gz: estos ficheros comprimidos -llamados Tarball- contienen el código fuente del programa en lenguaje C o C++.

Para instalarlo lo tendremos que compilar para nuestro modelo de ordenador, proceso que puede tardar varios minutos dependiendo del tamaño del programa, y que también puede dar algunos problemas. Funciona en todas las distribuciones Linux.

RPM: formato de ficheros originariamente para Red Hat, pero que se ha adaptado a otras distribuciones. Es rápido y fácil de usar, pero tiene los inconvenientes de que hemos de encontrar los paquetes para nuestro modelo en concreto de ordenador. Además, suelen dar problemas de dependencias.

DEB: formato propio de Debian. Similar al RPM, pero más seguro. La gran ventaja que presenta es el llamado apt, que sirve para gestionar los paquetes instalados y añadir nuevos quitar otros sin prácticamente ningún esfuerzo.

Otras características: cada distribución está orientada a un tipo de usuario en concreto. Por ejemplo, Red Hat está orientada a grandes empresas, Mandrake a principiantes y Gentoo a profesionales. Esta última no es apropiada para nuestro caso porque hay que compilar cada uno de los programas que se instalan, y eso nos haría perder muchas horas.

Después de haber hecho esta comparación, creo que la distribución más apropiada para el instituto es Debian, porque, además de representar al software libre, es simple y muy estable.

Es una de las distribuciones más aptas para hacer de servidor. Aparte de ésta, también se usa mucho FreeBSD, que es un sistema basado en UNIX

No utilizamos Mandrake, Red Hat ni Suse por ser demasiado orientadas a usuarios domésticos y principiantes. Gentoo es demasiado complicada y difícil de configurar y queremos un sistema que podamos resolver los errores al instante y tener suficiente documentación en Internet para poder configurarlo.

Debian desarrolla a la vez tres ramas de su sistema operativo: la versión estable, la inestable y la 'en pruebas'. Las diferencias entre versiones son:

- Estable ("stable"): la más recomendada. Está muy probada y teóricamente no debería fallar nada.

- Inestable ("unstable"): la que van mejorando los programadores cada día. No es seguro que funcione perfectamente. Cuando pasa tiempo, se dedican a probarla a fondo (se convierte en 'en pruebas') hasta que llega a ser 'stable'.

- En pruebas ("testing"): la versión inestable bloqueada, a la cual no añaden nada más y sólo se dedican a probarla a fondo. Cuando, después de unos meses, ven que puede salir al público, la convierten en 'stable'.

Cada versión lleva un nombre clave (Potato, Buzz, Rex, Bo, Slink, etc.), que, por cierto, son los nombres de los personajes de la película Toy Story. En el momento de escribir esto (septiembre de 2002), la versión estable es la Woody (es la versión 3.0), la 'en pruebas' se llama Sarge (será la versión 3.1) y la inestable se llama Sid.

La que usaremos para el servidor será la versión testing; no podemos arriesgarnos probando una inestable pero tampoco pondremos la versión stable por que los paquetes ya son algo antiguos y de esta forma es más fácil actualizar el sistema.

Por lo tanto, nos decidimos por una Debian testing.

2.a.ii. Proceso de instalación

Podemos instalar Debian de diversas maneras, de las cuales las más comunes son los CDS con las imágenes o la instalación por Internet. También lo podemos comprar en alguna tienda por un precio muy reducido. Toda la información la encontraremos en http://www.debian.org,

Como no nos hace falta más que los programas básicos, es más que rentable hacer la instalación por Internet, ya que nuestro CD ocupa 110Mb

1. Bajamos la imagen del CD-ROM de un FTP. La imagen debe de ser la versión netinst para la versión testing y un procesador x86 las encontraremos en:

http://cdimage.debian.org/pub/cdimage-testing/sarge_d-i/i386/rc2/sarge-i386-netinst.iso

Esta imagen la grabamos en un CD-ROM y ya tenemos el Linux Debian Sarge listo para instalar en nuestro PC.

2. Procedemos a la instalación

1.A Insertamos el CD-ROM en el lector de nuestro PC y reiniciamos el ordenador.

NOTA: Mirad si tenemos bien configurado la BIOS para que arranque desde el CD-ROM antes que del HD.

1.B Nos aparecerá una pantalla de bienvenida y pulsaremos la tecla ENTER.

1.C Nos aparece un recuadro para elegir el idioma ponemos "Spanish" y pulsamos ENTER.

1.D Nos pide que pongamos el país donde estamos "España".

1.E Elegimos la distribución del teclado "Español".

1.F Ahora nos detecta que tenemos una tarjetas de red para Internet

Esta va a ser la tarjeta de red que va a ir a Internet, le pondremos la dirección IP: 192.168.1.100, mascara de red: 255.255.255.0, pasarela va a ser la dirección IP del Router: 192.168.1.1, direcciones de servidores de nombres ponemos unas DNS: 195.235.113.3, nombre de la máquina pondremos Debian-server, nombre de dominio: tranca.

1.G Método de particionado, le daremos a borrar editar manualmente la tabla de particiones, seleccionamos partición por partición borrándola, cuando no haya ninguna partición pondrá: pri/log 4.3GB ESPACIO LIBRE. Seleccionamos el espacio libre y le damos al ENTER, le damos a crear partición, ponemos 250 MB, la ponemos primaria y la ubicamos al principio del HD, entramos en la configuración de la partición:

```
Utilizar como: área de intercambio
Marca de arranque:   desactivada
Tamaño:      246.7 MB
```

Le damos a "Se ha terminado de definir la partición", ya tenemos la partición de área de intercambio, ahora hay que crear la partición del sistema operativo Linux.

Seleccionamos el espacio libre de nuevo, Crear una partición nueva, dejamos lo que ponga para usar el espacio máximo en esta partición, en mi caso 4G, Primaria, ahora la configuramos:

```
Utilizar como: Sistema ext3 con <<journaling>>
Punto de montaje     /
Opciones de montaje: defaults
Etiqueta:            /
Marca de arranque:   desactivada
Tamaño:              4.0 GB
```

Y seleccionamos "Se ha terminado de definir la partición", y volvemos al menú de particionado, le damos a la opción "Finalizar el particionado y escribir los cambios en el disco", por último ¿Desea escribir los cambios en los discos? => SI.

1.H Instalando el sistema base….

1.I Instalar el cargador de arranque GRUB en el HD: SI.

1.J La instalación se ha completado, extraemos el CD de instalación y le damos a continuar y reiniciamos el PC.

1.K Cuando arranquemos de nuevo el PC, nos sale el GRUB pulsamos ENTER para entrar en el SO.

1.L Nos sale la pantalla de “bienvenido a su nuevo sistema Debian!”.

1.M Fijamos el reloj del sistema UNIX a GMT => SI.

1.N Europe / Madrid (mainland) Aceptamos.

1.Ñ Ponemos contraseña de superusuario o root.
Esta contraseña es privada, no se la debes de mostrar a nadie, hay que intentar no poner contraseñas fáciles como: 123456, root, abcdef… hay que intentar poner una palabra que no este en el diccionario y que no tenga nada que ver con nosotros ni con el instituto, es mejor que sea una combinación de números y letras, no repetir los caracteres y no la anotes en ningún lado.

1.O Introducir el nombre completo

1.P Introducimos el nombre de usuario que vamos a acceder al sistema a partir de ahora, solo accederemos como root cuando sea estrictamente necesario.

1.Q Introducimos la contraseña de este usuario y siempre es bueno poner la contraseña con las mismas condiciones que la contraseña de root.

1.R Configuración de apt, esta es la aplicación con la que podremos descargar algunos paquetes e instalar sin mucho esfuerzo, en este apartado pondremos FTP, para que coja los archivos desde un servidor FTP y elegimos el servidor, en mi caso suelo poner:

```
Estados Unidos => ftp.es.debian.org,
España => ftp.rediris.es
```

1.S Información sobre Proxy HTTP, no ponemos nada y le damos al ENTER.

1.T Nos sale un menú donde podemos elegir algunas aplicaciones, bueno pues no elegimos ninguna por que los paquetes que queremos los vamos a instalar nosotros mas adelante, le damos al ENTER.

1.U Cuando termina de bajar, le damos a aceptar.

1.V Configuración de console-data => No tocar el mapa del teclado.

1.W Configuración de Exim version 4 (exim4-config) es para la configuración del correo, y como no queremos, le damos a “Sin configuración de momento”, y le damos a “Si estamos seguros”.

1.X Destinatario del correo de <<root>> y <<postmaster>> aceptar.

1.Y Reinicia el sistema.

1.Z Ya esta listo para su configuración.

2.a.iii. Configuraciones básicas

Ahora que tenemos el sistema operativo instalado hay que hacer unas configuraciones sencillas, que deben hacerse antes de empezar a poner programas y demonios. Lo primero que hace falta es identificarse como root.

Configuraremos apt que es el sistema de control de paquetes exclusivo de debian, esto lo haremos siendo usuario root y pondremos en la línea de comandos:

```
#apt-setup
FTP => Estados Unidos => ftp.es.debian.org
```

Con este comando pondremos líneas al archivo /etc/apt/sources.list, con esta aplicación se descarga, te instala y configura los programas que pongas, este también se descarga todas las dependencias que vayan a hacer falta para su correcto funcionamiento.

Actualizaremos el sistema
`#apt-get update`

Para actualizar la información sobre Linux los nuevos paquetes del servidor

`#apt-get upgrade -u` (el -u es para mostrar los nombres)

Para actualizar los paquetes instalados con las nuevas versiones disponibles

Después de mostrar la lista de paquetes a bajar, respondemos (Y/n) y sólo habrá que esperar a que acabe.

`#apt-get install PAKETE` Si queremos algún paquete adicional que este en los servidores que pusimos en el sources.list.

También es recomendable instalar ahora gpm para poder utilizar el ratón en la consola (se configura con gpmconfig).

Hagamos más cómoda la shell:

Entramos en el archivo /etc/profile y en el principio ponemos

```
alias ls="ls --color"
```
Para ver el listado de ficheros en colores

```
setleds +num
```
Para activar NumLock

```
alias "cd.."="cd .."
alias i="apt-get install"
alias u="apt-get update && apt-get upgrade -u"
```

Protección de algunos archivos importantes:

Quitaremos los permisos de lectura, escritura o ejecución

```
#chmod o-x /etc/passwd /etc/*netd.conf
```

/etc/passwd tiene información sobre los usuarios del sistema

/etc/inetd.conf o xinetd.conf dice que servicios se cargan cada vez que se enciende el PC.

Instalación de un kernel nuevo, si hay: no es necesario si no tenemos problemas con el actual, pero es recomendable porque soluciona problemas de seguridad que afectan a todo el sistema. , algunas cosas cambian, como por ejemplo la forma de implementar las reglas del cortafuego que montaremos. Podemos ver la versión del kernel actual con:

```
#uname -a
```

Los nuevos kernels se encuentran precompilados, o sea, que sólo hace falta bajarlos con apt-get. Para saber el nombre y tipo de kernel que necesitamos, podemos buscar en la base de datos local de paquetes con:

```
#apt-cache search kernel-image
```

Tenemos que fijarnos en todos los disponibles. La versión ha de ser:

- Superior al actual
- La plataforma ha de ser la de nuestro sistema
- No hay que bajar una versión SMP "Symmetric Multiprocessor”

Instalaremos otro kernel que es el 2.6.8 ya que nuestra versión era la 2.4

```
#apt-get install kernel-image-2.6.8-2-386
```

Directamente te configura el grub y te añade dos líneas nuevas en la configuración, las del nuevo kernel, si da error este kernel se elimina y no pierdes el sistema

Ya configuramos la tarjeta de red en la instalación, ahora algunos comandos para poder sabernos manejar con la tarjeta de red:

#ifdown eth0 Echar abajo la red de eth0
#ifup eth0 Levantar la red de eth0

/etc/resolv.conf => Están las DNS
/etc/network/interfaces => Esta la configuración de la red que interpreta cuando arrancas el PC.
/etc/fstab => Donde aparecen los dispositivos montados
/etc/motd => Modifícalo para tener un inicio de sesión más amigable
/etc/profile => Todo lo que pongamos es ese archivo se van a ejecutar cuando iniciemos sesión, no es conveniente cargarlo mucho.

2.b.Servicios básicos

2.b.i. Servidor Web para Internet

El servidor Web es un programa que corre sobre el servidor que escucha las peticiones HTTP que le llegan y las satisface. Dependiendo del tipo de la petición, el servidor Web buscará una página Web o bien ejecutará un programa en el servidor. De cualquier modo, siempre devolverá algún tipo de resultado HTML al cliente o navegador que realizó la petición. Este servidor escucha por el puerto 80 normalmente, pero nosotros lo configuraremos para que escuche por el puerto 8080.

En el mercado hay muchos, y en concreto que funcionen bajo Linux también (Jigsaw, GoAhead, Roxen, Stronghold, Zeus, Abyss, Apache,...). Incluso podemos programar uno sencillo con Netcat, haciendo que escuche en el puerto 80 y devuelva cada página pedida. Pero en Internet los servidores más usados son claramente dos: Apache y Microsoft IIS (Internet Information Server). Obviamente, IIS es sólo para Windows, así que para nuestro ordenador usaremos Apache para Linux.

Integrar Apache en Debian es tan sencillo como hacer un apt-get install apache (se supone que la base de datos de apt-get ya está actualizada mediante apt-get update) y automáticamente se bajarán todos los paquetes necesarios.

```
#apt-get install apache
```

Si todo ha salido bien, es decir, no a dado errores en la compilación ni en la instalación, ya tendremos instalado apache.

Comprobamos su instalación, para ello instalamos un cliente Web en consola y arrancamos el servidor

```
#apt-get install lynx          #instalar cliente Web
#lynx 127.0.0.1                #visitando nuestra Web
```

Y nos saldrá una página Web informándonos que el apache ha sido instalado correctamente

Antes de configurar el apache vamos a ver la estructura que tiene para sabernos manejar en el apache:

/etc/init.d/	Archivos para iniciar, parar, resetear servidor
/var/www	Alojamiento para la página Web
/etc/apache	Archivos de configuración
/var/log/apache	Accesos a la Web

Vamos a configurar apache:

Paramos el servidor
```
#cd /etc/init.d
#./apache stop
```

Entramos en el archivo de configuración:
```
#cd /etc/apache
#nano httpd.conf
```

Este archivo de configuración es bastante extenso, por lo tanto nos vamos a dirigir a las partes más importantes.

`ServerType Standalone`
Tipo de servidor, es mejor que inetd

`ServerRoot /etc/apache`
Donde esta instalado el apache

`Port 80`
Puerto donde escucha apache, cambiamos por 8080

```
User www-data
Group www-data
```
Usuario y grupo de los usuarios que visitan la Web por defecto

`ServerAdmin you@example.com`
Lo cambiamos por `antus.oto@gmail.com` que es el correo del administrador del servidor, así cuando salga una pagina de error salga ese correo para poder dar parte de ello

`DocumentRoot /var/www`
Directorio donde se aloja la pagina Web

`DirectoryIndex index.html index.htm index.shtml index.cgi index.php`
Esto es que cuando pongamos //127.0.0.1 sea igual a //127.0.0.1/index.html y así se auto arranca la pagina Web

```
AccessFileName .htaccess
```
Es el nombre del archivo que apache mira cuando se accede a una carpeta, sirve para poder restringir el acceso a usuario a una parte la página Web

```
ErrorLog /var/log/apache/error.log
```
Archivo donde se guardan los errores del apache que puedan ocurrir

Comprobamos que la configuración ha sido aplicada correctamente, arrancamos el servidor, y visitamos de nuevo la Web

```
#cd /etc/init.d/
#./apache start
#lynx 127.0.0.1
```

Ahora nos debe de decir lynx: Unable to connect to remote host, es decir, que no se puede conectar al equipo, esto pasa por que en el archivo de configuración le hemos dicho que el apache escuche desde el puerto 8080, no desde el 80 (puerto predeterminado de los navegadores), entonces deberíamos hacer:

```
#lynx 127.0.0.1:8080
```

Y ahora si debería de funcionar correctamente y mostrarnos la pagina que trae por defecto el servidor apache, esto se lo modificaremos mas adelante cuando hagamos la pagina Web, por ahora solo hemos instalado y configurado el servidor Web apache.

2.b.ii. Integración con la red Windows

En este instituto como en la mayoría hay implantados SO en Windows surge la necesidad de conectar estos equipos en Windows con este que esta en Linux y para ello esta SAMBA.

Servidor samba: Para que Windows pueda entrar los recursos compartidos de nuestro sistema Linux.

Cliente smbclient: Para poder acceder desde Linux a los recursos compartidos de Windows.

Esto nos va a servir para hacer nuestra Web en un ordenador mas potente que puede ser Windows o Linux y después poder montar la Web desde ese mismo ordenador o bien copiársela a directorios del servidor, esto ofrece una gran ventaja por que nuestro MinServ no va a llevar entorno gráfico y no seria muy cómodo de trabajar con él ya que nuestro MinServ no tiene recursos para ponerle el entorno gráfico y las herramientas necesarias.

Todo esto lo haremos fácilmente y de forma segura con las herramientas que nos ofrece Samba. Samba es una implementación para Linux del protocolo SMB (Server Message Block), creado por IBM en 1985, redefinido después por Microsoft, y presente en otros sistemas operativos. Con Samba podremos acceder (por TCP/IP) a servidores SMB como cliente, o montar un servidor SMB propio.

Samba3 va a un 50% mas rápido que el que utiliza Windows por ello es más rápido y eficiente.

Para instalar el Servidor samba haremos:

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

Una vez instalado nos hacen unas preguntas sobre la configuración del servidor

*Nombre del dominio o del grupo de trabajo
En este caso es tranca

*¿Cómo queremos que se ejecuten los procesos de Samba, como demonios o como una parte del demonio inetd?

Es mejor que se ejecuten como demonios, ya que así se podrán controlar mejor.

*¿Crear la base de datos de contraseñas /var/lib/samba/passdb.tdb? ¡Sí! Si no lo hacemos, aparecerán problemas extraños cuando intentemos entrar a recursos compartidos de un Windows NT. Además, es para crear contraseñas cifradas y es obvio que las contraseñas cifradas dan más seguridad que las estándar.

Después de esto ya tenemos los dos procesos de Samba funcionando: smbd y nmbd.

Si queremos que Debian nos vuelva a preguntar todo lo anterior podemos hacer un

```
#dpkg-reconfigure samba
```

Hay que tener claro que es lo que queremos hacer en nuestro servidor samba para poderlo configurar adecuadamente y funcione al máximo rendimiento, sobre las decisiones que tomemos son:

Hemos de compartir sólo una carpeta del servidor (con sus directorios) a todos los usuarios de Windows de la red con una contraseña para poder modificar el contenido de la página Web.

Hay que tener cuidado por que es muy probable que salga un problema si usamos tanto máquinas Windows NT/2000 como Windows 9x a la vez, debido a su forma de enviar las contraseñas y a los requisitos de cada implementación.

Para entender estas complicaciones hay que ver unas cuantas diferencias entre el SMB de Win98, WinNT y Samba:

- El Windows 98 envían contraseñas encriptadas por defecto, mientras que Samba las suele recibir en formato de texto normal. Esto tiene solución fácil añadiendo la directiva encrypt passwords = yes al fichero de configuración.

- Los Windows NT y Samba son muchísimo más configurables que el SMB de Windows 98. Entre otras cosas, con WinNT y Samba podemos especificar el nombre de usuario y contraseña con los que queremos acceder a un recurso compartido. Esto quiere decir que con Windows 98, la única forma de acceder con un nombre de usuario en concreto es ser usuario (de todas maneras, crear un nuevo usuario cuesta poco).

- Samba y Windows NT no aceptan 'invitados' por defecto. Los podemos activar, pero provocan problemas de seguridad. Este es un tema muy delicado en Windows NT, donde es muy fácil encontrar sistemas a los que se puede entrar usando Invitado como nombre de usuario y dejando en blanco la contraseña...

Bien, pues podemos empezar: lo primero de todo será crear un usuario en el servidor que sea el único que pueda tener acceso a los recursos compartidos.

Es el usuario que usarán todos los ordenadores que accedan, y, tal como hemos dicho antes, es necesario que este usuario esté creado en todos los Windows y tenga la misma contraseña o tendrán que poner nombre de usuario y contraseña cada vez que se acceda al servidor.

Podemos hacer una excepción con los Windows NT ya que, como hemos explicado antes, permiten escribir un nombre de usuario y contraseña al acceder, independientemente del nombre del usuario conectado.

> *NOTA*: no hace falta que vayamos ordenador por ordenador creando este usuario, ya que se creará automáticamente cuando alguien ponga su nombre y contraseña en la pantalla de inicio de sesión.

En nuestro caso a este usuario lo llamaremos **marismas** y le daremos una contraseña fácil de memorizar "**adminwinxp**" (¡pero no de adivinar!). La contraseña es opcional, pero el no poner representa un agujero de seguridad muy importante (sobre todo cuando hablamos de sistemas Microsoft).

Añadimos el usuario al servidor con

```
#adduser marismas
```

Nos pide la contraseña "adminwinxp", lo demás no hace falta que lo pongamos y lo añadimos a la lista de usuarios de Samba que se encuentra en

/etc/samba/smbpasswd con el comando

```
#smbpasswd -a marismas
```

Y poniendo la misma contraseña.

Nuestro propósito de instalar samba es para poder subir la página Web directamente desde otro PC de la red.
Ahora mismo el usuario marismas cuando accedemos a el desde otro pc nos dirige a /home/marismas, entonces le cambiamos su "domicilio" al de la página Web:

```
#cd /etc
#nano passwd
```

Abajo del todo en el usuario:
`Marismas:x:100:1001:,,,:/home/marismas:/bin/bash`
Lo sustituimos por:
`Marismas:x:100:1001:,,,:/var/www:/bin/bash`

Así cuando entremos en la carpeta de marismas directamente podremos modificar la página Web

Ahora hemos de editar el fichero de configuración
/etc/samba/smb.conf.

Al principio encontraremos las opciones que pusimos al instalar Samba; mejor comprobamos que haya un:

```
Workgroup = NOMBRE_DEL_GRUPO_DE_TRABAJO.
Server string = %h
```

Habrá una línea comentada con el carácter; que pone:

```
;    Guest account = nobody
```

Es bueno saber que si la cambiamos a guest account = marismas haremos que dejar el nombre de usuario y contraseña en blanco equivalga a entrar con el usuario marismas (sin contraseña). Como hemos decidido que había que poner contraseña, dejaremos la línea comentada, pero si vemos que es más fácil sin contraseña, sólo hay que modificar eso.

En esta sección también hay que añadir la siguiente línea para evitar uno de los métodos de intrusión más usados desde hace muchos años: el de los recursos compartidos.

```
Hosts allow = 192.168.1. localhost
```

Y también haremos que sólo permita el acceso al usuario marismas.
Si se crean más habrá que ponerlos aquí también.

```
Valid users = marismas
```

Ahora localizamos la sección titulada Share Definitions y, en concreto, este fragmento:

```
[homes]

   comment = Home Directories
   browseable = no
   writable = no
   create mask = 0700
   directory mask = 0700
```

Según nuestros intereses, lo cambiaremos a:

```
[homes]

   comment = Home Directories
   browseable = yes
   writable = yes
   create mask = 0777
   directory mask = 0777
```

Todo ese fragmento hace que se compartan las carpetas personales de los usuarios ([homes]). En nuestro caso sólo hay un usuario, que se llama marismas. Éste tiene permiso para escribir en su directorio (writable=yes) y todo lo que se cree podrá ser leído, modificado o ejecutado por cualquier usuario del sistema. El browseable=yes hace que el recurso se pueda ver al navegar por los recursos compartidos del servidor.

Insertamos estas líneas en esta parte:

```
[log]
    comment = Directorio de control
    path = /var/log/apace/
    browseable = yes
    writable = yes
    create mask = 0777
    directory mask = 0777
```

Con esto tendremos otra carpeta donde podremos ver los errores y los accesos que tiene nuestro Server desde la red local

El siguiente párrafo, que empieza por [printers], lo comentaremos entero con un carácter # en cada línea, porque no tenemos impresoras para compartir conectadas al servidor.

Reiniciamos el servicio

```
#cd /etc/init.d/
#./samba restart
```

Probamos a acceder desde un ordenador Windows explorando la red (en 'Entorno de red' o 'Mis sitios de red'). Veremos un ordenador llamado 'Marismas', y dentro, la carpeta compartida, donde podemos entrar y dejar archivos. También veremos el icono para gestionar las impresoras.

NOTA puede surgirnos un conflicto con los permisos, es decir, podremos acceder desde el ordenador en Windows al Linux, pero no podremos modificar, borrar, no crear ningún archivo, para solucionar este problema:

```
#cd /var/
#chown marismas www
#cd /var/log/
#chown marismas apache
#cd ../www
#chown marismas *.*
#cd ../log/apache/
#chown marismas *.*
```

Esto sirve para decirle al sistema que el propietario de esta carpeta y de sus archivos que tiene dentro es el usuario "**marismas**" que si recordamos, es el usuario con el que accedemos mediante el Windows con samba.

Con esto ya podremos modificar los archivos, crearlos y borrarlos desde el ordenador que queramos acceder y podremos ver los accesos y errores del Server.

Ya esta instalado el Server Samba.

Curiosidades -

Otras utilidades que nos Irán bien son testparm para comprobar la configuración y smbstatus para ver quien está conectado.

Nos queda explicar como acceder a la red de Windows desde Linux. No es difícil, pero hay muchas opciones de todo tipo. Por ejemplo, hay programas gráficos como komba y xfsamba, y también se puede hacer desde consola.

Vamos a explicar estos pasos sólo para curiosos que le guste este tema, pero estas aplicaciones no van a hacer falta para la implementación de MinServ.

Para instalar el cliente samba haremos:

```
#apt-get install smbclient
#apt-get install smbfs
```

smbclient -L host mostrará los recursos compartidos del equipo host. Podemos especificar el usuario con:

```
#smbclient -L host -U NombreUsuario
```

```
smbmount //host/nombredelrecurso /mnt/samba
```

Monta la carpeta o unidad compartida especificada en el directorio local que se le indique, como si fuese un disquete. Después podremos acceder de manera normal, y copiar archivos, borrar, crearlos, cambiar los permisos, etc.

*Nota: para especificar el nombre de usuario hay que usar
`#smbmount //host/nombrerecurso /mnt/samba -o username=NombreU`

smbumount /mnt/samba desmonta el recurso. Hay que hacerlo antes de que se apague el ordenador Windows porque sino saldrán mensajes de error.

nbtscan 192.168.0.0/24 escanea toda la red (de tipo C, con máscara 255.255.255.0) y muestra los equipos que comparten recursos.

2.b.iii. **Servidor de base de datos**

Esta base de datos va a contener los datos personales de los alumnos de la clase, algunos datos de los profesores, las notas de los alumnos y sus nombres y contraseñas con lo que pueden acceder a la base de datos.

Con esta base de datos, la característica que nos brinda es que mediante PHP que es un lenguaje de programación podremos tomar los datos de la base de datos y representarlos en el servidor Web Apache

Para la instalación de MySQL tendremos recurrir al famoso:

```
#apt-get install mysql-server
```

Una vez terminado, tendremos que empezar a configurar la base de datos, 1º lo que tenemos que hacer es diseñarla antes de hacerla:

BD: marismas

Director
Modifica datos personales de alumnos

Profesores
Modifican las notas solo de sus alumnos

Alumno
Visitan sus datos personales y sus notas

2º Esto es lo que queremos hacer, ahora vamos haber que campos insertamos:

Profesores
id_profesores, nombre, apellidos, asignatura

Alumnos
id_alumnos, nombre, apellidos, codigo_postal, Fecha_nacimiento, sexo, direccion, poblacion, email

Notas
id_notas, agc, fol, bd, mpi, ims

Contrasenia

id_contrasenia, usuario, contrasenia
Esta es la idea, vamos a comenzar dando una contraseña al usuario root para una mayor seguridad de los datos

```
#mysqladmin -u root password adminwinxp
```

Creamos la base de datos con el nombre 'marismas'

```
#mysqladmin -u root create marismas -p
```

Iniciamos el servidor mysql

```
#mysql -h localhost -u root -padminwinxp marismas
```

Ya hemos puesto la contraseña y hemos entrado, ahora vamos a dar una serie de comandos para que sea un poco mas fluida nuestra navegación por la base de datos:

```
mysql> SELECT USER();
"ver usuario"

mysql> SHOW DATABASES;
"muestra bases de datos"

mysql> SHOW TABLES;
"muestra tablas"

mysql> DESCRIBE profesores;
"muestra la estructura"

mysql> quit
"salir de la BD"

mysql> SELECT * FROM profesores;
"muestra el contenido"

mysql> update contrasenias set
usuario='root@localhost' where id_contrasenia=1
"modificar el campo usuario de la tabla
contrasenias donde el id_contrasenia sea 1"

mysql> delete from user where user='manolo';
"borrar usuario manolo"

mysql> drop database marismas;
"borrar base de datos"
```

```
#mysqldump -u root -p --opt marismas >
marismas.sql
"Crear copia de seguridad de la base de datos
marismas.sql"
```

Esto es una pequeña introducción a unos términos que vamos a ir detallando poco a poco en el paso de nuestro script.

#Creamos unas tablas que tiene los campos que dijimos antes en el diseño
#Un campo se llama KEY es la clave primaria
#INT(2) significa que se va a introducir un número entero entre el 0 y el 99
#VARCHAR(20) una línea de caracteres, como máximo 20
#DATE es para las fechas
#NOT NULL es que no puede ser un campo vacío
#Introducimos los registros de las tablas
#Con esto acabamos la base de datos.

Estamos fuera de la base de datos y creamos un archivo de texto convencional:

```
#cd /root
#nano scriptbd.sql
```

Entramos dentro de este nuevo archivo y empezaremos con la base de que nuestra BD "base de datos" marismas ya esta creada.

```
#-----------------------------------------------
#    Creando tablas
#-----------------------------------------------
CREATE TABLE profesores(
id_profesores INT(2) NOT NULL,
nombre VARCHAR(20) NOT NULL,
apellidos VARCHAR(20),
asignatura VARCHAR(20) NOT NULL,
KEY(id_profesores)
);

CREATE TABLE alumnos(
id_alumnos INT(2) NOT NULL,
nombre VARCHAR(20) NOT NULL,
apellidos VARCHAR(20) NOT NULL,
codigo_postal VARCHAR(20),
fecha_nacimiento DATE,
sexo CHAR(1),
direccion VARCHAR(60) NOT NULL,
poblacion VARCHAR(20),
ciudad VARCHAR(20),
email VARCHAR(30),
KEY(id_alumnos)
);

CREATE TABLE notas(
id_notas INT(2) NOT NULL,
agc INT(2),
fol INT(2),
bd INT(2),
mpi INT(2),
ims INT(2),
KEY (id_notas)
);


CREATE TABLE contrasenias(
id_contrasenia INT(2) NOT NULL,
usuario VARCHAR(20) NOT NULL,
contrasenia VARCHAR(20) NOT NULL,
KEY (id_contrasenia)
);
```

```
#-----------------------------------------------
#    Insertando registros en las tablas
#-----------------------------------------------
INSERT INTO profesores VALUES('1','Jesus','no lo
se','mpi');
INSERT INTO profesores VALUES('2','Eduardo','no lo
se','db');
INSERT INTO profesores VALUES('3','Encarni','no lo
se','ims');
INSERT INTO profesores VALUES('4','Julia','no lo
se','fol');
INSERT INTO profesores VALUES('5','Julia','no lo
se','agc');

INSERT INTO alumnos
VALUES('1','Isabel','Aguadel','21003','1986-02-
15','F','-','Huelva','Lepe','-');
INSERT INTO alumnos
VALUES('2','Angel','Baldera','2004','1985-01-
26','M','-','Huelva','Huelva','-');
INSERT INTO alumnos
VALUES('3','Fermin','Bermejo','21005','1986-06-
14','M','-','Huelva','Huelva','-');

INSERT INTO notas VALUES('1','9','5','7','7','8');
INSERT INTO notas VALUES('2','5','6','4','5','7');
INSERT INTO notas VALUES('3','6','2','5','6','5');

INSERT INTO contrasenias
VALUES('1','isabel','unisabel');
INSERT INTO contrasenias
VALUES('2','angel','unangel');
INSERT INTO contrasenias
VALUES('3','pelopua','unpelopua');

#-----------------------------------------------
#    FIN del script
#-----------------------------------------------
```

Terminamos de escribir los datos, salimos y guardamos los datos que hemos introducido en el script y con esta línea de comandos le decimos que ejecute el script en la base de datos llamada marismas.

```
#mysql –u root marismas <scriptbd.sql –padminwinxp
```

Si todo ha salido bien y no nos ha dado ningún error, cosa que no debería "este script ha sido probado" entramos en la base de datos para ver lo sucedido.

```
#mysql –h localhost –u root –padminwinxp marismas
```

Ya hemos terminado la base de datos.

2.c. Servicios secundarios

2.c.i. Soporte de PHP en la Web

Nuestro servidor apache lo podremos aprovechar mucho más el servidor si usamos páginas Web que incluyan scripts en PHP "paginas dinámicas".

PHP es un lenguaje muy potente que se ejecuta en el servidor cuando alguien pide una página, y que genera un código HTML que incluye en la página de retorno.

Hay que destacar que nada del código PHP que se escriba lo podrá ver nunca el usuario (aunque intente descargar el .php), ya que el servidor genera y envía sólo código HTML.

Las posibilidades son inagotables: se pueden crear dinámicamente páginas que pide el usuario, se pueden usar cookies, contraseñas, sesiones que expiran al cabo de un tiempo, contadores, se puede acceder a bases de datos con el motor MySQL, y muchas cosas más. Hay muchas páginas importantes hechas con lenguaje PHP. También hay herramientas especiales, como PHPNuke, que don las bases para crear un portal sin tener que aprender a programar en PHP.

Para nuestro servidor, nos ocuparemos de instalar PHP, integrarlo en Apache, y comprobar que funcione con un ejemplo que haremos después.

Un apt-cache search php nos revela que el paquete más apropiado se llama php4 y php4-mysql, por lo tanto hacemos un apt-get install php4-mysql y apt-get install php4 y se bajará junto con sus dependencias.

Sólo te configura los archivos de configuración pero sino lo hace tendremos que hacerlos nosotros manualmente:

```
<IfModule mod_php4.c>
    AddType application/x-httpd-php .php .phtml .php3
    AddType application/x-httpd-php-source .phps
</IfModule>
```

También tenemos que decirle a Apache que la página a cargar por defecto puede ser index.php. Buscamos un fragmento donde pone:

```
<IfModule mod_dir.c>
    DirectoryIndex index.html index.htm index.shtml
index.cgi
</IfModule>
```

Y añadimos index.php al final, de forma que queda:

```
<IfModule mod_dir.c>
    DirectoryIndex index.html index.htm index.shtml
index.cgi index.php
</IfModule>
```

Así, intentará cargar primero index.html, si no lo encuentra index.htm, después index.shtml y así sucesivamente. Podemos ponerlo en el orden que queramos, aunque no debería haber más de un fichero index.* en el directorio.

Ahora reiniciamos Apache y vamos a crear un script PHP para comprobar su funcionamiento.

```
#cd /var/www
#nano info.php
```

Y escribimos sólo esto

```
<?
phpinfo ();
?>
```

Guardamos el archivo y ponemos:
`#lynx 127.0.0.1:8080/info.php`

Nos ha salido la pagina, eso quiere decir que esta correctamente instalado y configurado con el apache, ahora hay que probar con la base de datos.

Crearemos otro script que nos diga si se consigue una conexión con nuestra base de datos en MySQL.

En este script va a aparecer información que no nos interesa que la vean usuarios anónimos, pero no hay que olvidar que PHP no manda el script PHP solo manda código HTML, veamos:

```
<html>
<body>
<?php

function Conectarse()
{
     if
(!($link=mysql_connect("localhost","root","adminwinxp"
)))
     {
          echo "Error conectando a la base de datos.";
          exit();
     }
     if (!mysql_select_db("marismas",$link))
     {
          echo "Error seleccionando la base de datos.";
          exit();
     }
     return $link;
}

$link=Conectarse();
echo "Conexión con la base de datos conseguida.
";

mysql_close($link); //cierra la conexion
?>
</body>
</html>
```

Y lo guardamos con el nombre conexion.php. Esto si miramos por encima es muy fácil de adaptar este script a otras bases de datos, vemos localhost es por que este script se ejecuta en el PC local de la base de datos, root es el usuario y adminwinxp es la contraseña, un poco mas abajo vemos marismas que es el nombre de la base de datos a la que vamos a acceder desde este strip, para un correcto funcionamiento debe de aparecernos:

```
http://192.168.1.100:8080/conexion.php
```

Conexión con la base de datos conseguida.

Si nos aparece algún error de este estilo:

```
Fatal error: Call to undefined function:
mysql_connect() in /var/www/conexion.php on line 7
```

Significa que el archivo php.ini no esta bien configurado y le falta por descomentar una línea.

```
#cd /etc/php4/apache
#nano php.ini
```

En la línea 524 de este documento encontramos:

```
;   extensión=msql.dll          “la dejamos así:”
   extensión=msql.dll
```

Salimos del archivo, guardamos y reiniciamos el servidor apache.

```
#cd /etc/init.d
#./apache restart
```

Pues ya funciona nuestro PHP tanto en el Apache como en MySQL

3.c.ii. Conexiones por Secure Shell

Este servidor utiliza el protocolo SSH (siglas de Secure Shell) es un tipo de conexión idéntica a Telnet, pero encriptada con claves RSA, usa el puerto 22 de TCP y lo vamos a montar para poder actualizar y tener al día este MinServ desde cualquier parte del mundo, es decir, si surge algún conflicto, o algún pequeño fallo del sistema, solo hace falta un PC con Internet y desde ahí podemos conectarnos al MinServ y poder hacer los arreglos que veamos oportunos

La forma tradicional de hacer esto es montando un servidor de Telnet, y después conectando desde otra máquina con telnet IP. Este sistema sólo tiene un inconveniente: toda la información viaja por la red en formato texto, o sea, sin encriptar. Por tanto, cualquier persona que pueda interceptar paquetes podría ver todo lo que envía y recibe cada uno de los usuarios que hay conectados, incluso las contraseñas.

Sólo hay que hacer un `apt-get install ssh` y responder las preguntes que hará. Siempre podemos volver a hacer esta configuración guiada por preguntas con un dpkg-reconfigure ssh.

Para comprobar su funcionamiento hacemos:
```
#ssh 127.0.0.1
```

Nos aparecerá un mensaje diciendo que no somos un usuario autorizado. Los usuarios autorizados son los que no tienen que escribir su contraseña para acceder. Por seguridad, no crearemos ninguno de estos. Decimos yes para continuar con la conexión, y el mensaje no volverá a salir más en esta máquina.

Para entrar remotamente hay que especificar el usuario
```
#ssh 127.0.0.1 -l root
```

Una vez escrita la contraseña de ese usuario entramos en la consola y para salir de la shell pondremos el comando exit.

Remotamente tenemos el:
ssh para Linux
Putty para Windows
NiftyTelnet para Macintosh

Ya tenemos instalado el servidor de ssh

2.c.iii. Web para Internet

Ya que tenemos instalado el servidor apache con php y mysql, podremos empezar a diseñar la página Web, el diseño lo dejamos a la mano ustedes, yo utilicé un programa de diseño que se llama Xara Webstyle. Este programa nos cede el diseño entero, solo tenemos que combinarle el contenido y dicho contenido lo haremos mediante html y php.

El contenido de mi página Web es un poco lo que suelen tener todas las paginas, algo de historia del centro y directamente las cuentas de los usuarios, tanto alumnos, profesores y la cuenta del director.

Esta página Web una vez que la tengamos hecha, la colgaremos en el servidor para ver su estructura y para darnos cuenta de que el proyecto va por buen camino, esto lo haremos desde otro ordenador accediendo mediante la red, utilizando samba y comprobando su funcionamiento poniendo en el navegador:

http://192.168.1.100:8080

2.c.iv. Introducir Scripts en PHP

Contamos con el diseño que ya lo tenemos hecho, solo faltaría ingresarle el código en php para el control de usuarios, en este caso e decidido en poner contraseña cada vez que accedamos a la base de datos por mayor seguridad y no tener que preocuparnos de los últimos bug de seguridad que han surgido en php.

Para conectarnos a la base de datos vamos a utilizar un script de php lo llamaremos en los demás script para no tener que repetirlo todas las veces que nos hagan falta.

```
<?
$host = 'localhost';
$user = 'root';
$pass = '';
$dbname = 'marismas';
mysql_connect($host,$user,$pass) or die("No se pudo
establecer la conexión<br>".mysql_error());
mysql_select_db($dbname) or die("no se puede acceder a
la base de datos:<br>".mysql_error());
?>
```

Atención: para evitar problemas es aconsejable que el <? del principio este al principio de la pagina y que no halla ningún espacio en blanco, este script se va a llamar: conectarbd.php

El siguiente script sirve para poder mostrarnos una información de los usuarios y así poder ver el ID de los alumnos.

```
<?
include ("conectarbd.php"); //Conectamos a la bd
$result=mysql_query("select contrasenias.usuario,
alumnos.nombre, alumnos.apellidos, alumnos.id_alumno
from alumnos, contrasenias where
contrasenias.id_contrasenia=alumnos.id_alumno;");
//Esta es la consulta a la base de datos ?>

<table align="center">
<tr>
<th>Usuario</th>
<th>Nombre</th>
<th>Apellidos</th>
<th>ID_Alumno</th>
</tr>
```

```
<?
//Mostramos los registros
while ($row=mysql_fetch_array($result))
{
echo '<tr><td>'.$row["usuario"].'</td>';
echo '<td>'.$row["nombre"].'</td>';
echo '<td>'.$row["apellidos"].'</td>';
echo '<td>'.$row["id_alumno"].'</td></tr>';
}
mysql_free_result($result)
?>
</table>
```

Con esta parte ya veremos el resultado impreso en la pagina web, ahora vamos a poner un formulario para poder insertar su id de usuario y su contraseña, para poderlo contrastar con la base de datos y así poder ver sus notas.

```
<p><div align=center>
<FORM METHOD="POST" action=ver.php>
ID_Alumno<br><INPUT TYPE="TEXT" NAME="id"><br>
Contraseña<br><INPUT TYPE="PASSWORD"
NAME="contrasenia"><br>
<INPUT TYPE="SUBMIT" value="Ver mis notas">
</FORM></div>
```

Veremos que el formulario lo enviamos al archivo “ver.php”, veamos ese script:

```
<?
$id = $_POST['id']; //recordamos lo que nos a enviado
el formulario
$contrasenia = $_POST['contrasenia'];
include ("conectarbd.php");
$result=mysql_query("select contrasenias.usuario,
notas.agc, notas.fol, notas.bd, notas.mpi, notas.ims
from contrasenias, notas where
notas.id_nota=contrasenias.id_contrasenia and
contrasenias.id_contrasenia='$id' and
contrasenias.contrasenia='$contrasenia'");
```

```
?>
<table align="center" width=30%>
<tr>
<th>Usuario</th>
<th>AGC</th>
<th>FOL</th>
<th>BD</th>
<th>MPI</th>
<th>IMS</th>
</tr>
<?
//Mostramos los registros
while ($row=mysql_fetch_array($result))
{
echo '<tr><td>'.$row["usuario"].'</td>';
echo '<td>'.$row["agc"].'</td>';
echo '<td>'.$row["fol"].'</td>';
echo '<td>'.$row["bd"].'</td>';
echo '<td>'.$row["mpi"].'</td>';
echo '<td>'.$row["ims"].'</td></tr>';
}
mysql_free_result($result)
?>
```

Con este script veremos las notas solo del alumno que ha insertado su id y su contraseña, pues todos los script son parecidos, lo que cambia es la consulta a la base de datos que en vez de solo seleccionar, puede modificar, eliminar… todo lo posible con la base de datos.

2.c.v. Firewall

La seguridad en este MinServ es muy importante ya que al tener que dar varios servicios tiene varios puertos abiertos, para poderle dar seguridad le instalaremos un escaneador de puertos para escanearnos a nosotros mismos y comprobar nuestros puertos y también tendremos que ponernos el firewall IPTABLES que es el mas potente de Linux ya que viene incorporado en el kernel, este firewall o cortafuegos lo usaremos para aceptar, denegar o ignorar los paquetes que pasen por nuestro ordenador, IPTABLES tiene diferentes nombres en las diferentes versiones del kernel:

ipfwadm para kernel 2.0
ipchains para kernel 2.2
iptables para kernel 2.4

Nuestro kernel es el 2.4, por eso nuestra utilidad es IPTABLES y en nuestro equipo ya lo tenemos instalado automáticamente.

IPTABLES

Tenemos tres tablas, INPUT, FORWARD y OUTPUT, para catalogar los paquetes:

INPUT: los paquetes que llegan al servidor desde el exterior o desde la misma máquina pasan por la tabla de INPUT ('entrada'). Es de donde vienen todos los ataques.

FORWARD: los paquetes que llegan al servidor para ser enrutados hacia otro sitio pasan por la regla FORWARD ('hacer que continúe'). Hay que activar una opción antes para permitir el forwarding.

OUTPUT: los paquetes generados en el mismo ordenador y listos para ser enviados hacia el exterior pasan por la tabla OUTPUT ('salida').

Cada tabla consta de una serie lógica de reglas (ordenadas) que deciden el destino del paquete. El paquete acabará básicamente de una de estas formas:

ACCEPT: se deja pasar el paquete; el firewall no actúa para nada.
REJECT: se deniega el paquete. El firewall contesta diciendo que no quiere hacer la conexión, por tanto el emisor del paquete se entera.
DROP: ignora el paquete. Es como si el ordenador estuviera apagado y el paquete se perdiera.

Para entender mejor la diferencia entre REJECT y DROP (llamado 'DENY' en versiones anteriores), nos irán bien unos conceptos básicos de TCP/IP.

Para establecer una conexión a una máquina y un puerto determinados, hay que hacer el conocido como 'three way handshake' (saludo de las tres vías). Consiste en enviarle un paquete con el BIT SYN (SYNCHRONIZE) activado. Nos contestará con un paquete con el BIT ACK (ACKNOWLEDGEMENT) activado y hará lo mismo que hemos hecho nosotros: enviar un paquete con SYN y esperar un ACK nuestro; entonces ya puede empezar a enviar y recibir datos. Como enviar un ACK y luego un SYN se puede simplificar en sólo un paquete con los bits SYN y ACK activados, el 'three way handshake' queda así:

Nosotros		Destino	Descripción
SYN	--------->		¿Me recibes? Quiero conectar
	<---------	SYN/ACK	Sí, te recibo. ¿Comenzamos ya?
ACK	--------->		De acuerdo, comencemos

Pero esto sólo pasa si el puerto está abierto. Si un puerto está cerrado, al enviar el SYN responde con un RST/ACK y finaliza la conexión. Iptables por defecto contesta de otra forma, con un mensaje ICMP Port Unreachable, pero lo podemos especificar con el parámetro --reject-with

Por tanto, volvamos a definir las tres opciones ACCEPT, REJECT y DROP, pero de una manera más técnica. Si enviamos un SYN al ordenador destino, la respuesta que recibiremos dependerá de la regla utilizada:

ACCEPT: recibiremos un SYN/ACK y se establecerá la conexión.
REJECT: recibiremos un ICMP Port Unreachable o un RST/ACK (se puede escoger en la configuración). Por tanto, el servidor corta la conexión.
DROP: ¡no recibiremos nada! Al final será nuestro ordenador el que, cansado de enviar SYNs, acabe la conexión. Esto es muy interesante porque podemos hacer ver que un ordenador está apagado si le decimos que ignore todos los paquetes no deseados.

Nuestro trabajo si queremos crear un firewall es definir estas reglas para las tres tablas (INPUT, FORWARD y OUTPUT), y para hacerlo lo primero es saber lo que queremos hacer, es decir plantearnos como va a ser nuestro firewall:

Lo primero es borrar todas las reglas que traiga nuestro iptables por defecto, que suele ser "aceptar todo".

Nuestro firewall se va a basar en la metodología más segura que hay:

Hay 2 metodologías:
1º Deja pasar todo y restringe lo que no necesita.
Ventaja, es mas sencillo,
Inconveniente, si te abren un puerto y no lo sabes, lo tienes desprotegido

2º Ignoras todo y solo dejas pasar lo que te interesa:
Ventaja, es mas seguro y fiable
Inconveniente es más difícil

Vamos a usar la 2º metodología, lo primero que tenemos que hacer es plantearnos cual es la función de nuestro firewall. Tenemos:
Aceptamos:
Server apache => puerto 8080
Server ssh => puerto 22
Server samba => puerto 139

Denegamos:
Todo lo demás y lo protegemos

Interpretamos el script entero y lo guardamos con el nombre iptables dentro de la carpeta /root:

```
#cd /root
#nano iptables


# Borramos las reglas por defecto
iptables -F INPUT
iptables -F OUTPUT
iptables -F FORWARD

# Aceptamos la entrada y salida de los 3 puertos.
iptables -A INPUT -i eth0 -p tcp --dport 8080 -j ACCEPT
iptables -A OUTPUT -o eth0 -p tcp --sport 8080 -j ACCEPT
iptables -A INPUT -i eth0 -p tcp --dport 22 -j ACCEPT
iptables -A OUTPUT -o eth0 -p tcp --sport 22 -j ACCEPT
iptables -A INPUT -i eth0 -p tcp --dport 139 -j ACCEPT
iptables -A OUTPUT -o eth0 -p tcp --sport 139 -j ACCEPT

# Protegiendo de Anti-flooding o inundación de tramas SYN.
iptables -N syn-flood
iptables -A INPUT -i eth0 -p tcp --syn -j syn-flood
```

```
iptables -A syn-flood -m limit --limit 1/s --limit-burst 4
-j RETURN


# Protegiendo contra ataques por ICMP como ping masivo.
iptables -A INPUT -i eth0 -p icmp --icmp-type 8 -j DROP
iptables -A FORWARD -i eth0 -p icmp --icmp-type 8 -j DROP

# Evitando ataques del tipo “Tiny Fragment Attack”.
iptables -A INPUT -i eth0 -f -m length --length 0:40 -j
DROP

# Deshabilitando broadcast.
/bin/echo "1" >
/proc/sys/net/ipv4/icmp_echo_ignore_broadcasts

# Evitar ataques de spoofing..."
echo "1" > /proc/sys/net/ipv4/conf/all/rp_filter

# Deshabilitar la redirección del ping..."
/bin/echo "0" >
/proc/sys/net/ipv4/conf/all/accept_redirects

# Registrar los accesos extraños, paquetes falseados...
/bin/echo "1" > /proc/sys/net/ipv4/conf/all/log_martians

# Ignorando todas las demás conexiones posibles.
iptables -A INPUT -i eth0 -p tcp -j DROP
iptables -A OUTPUT -o eth0 -p tcp -j DROP
iptables -A FORWARD -i eth0 -p tcp -j DROP
```

Para comprender cada comando hay que saber los parámetros del programa. Podemos verlos todos con:

```
#man iptables
```

Los más importantes los veremos aquí:

```
iptables -A <tabla>
-i <interfaz de entrada> -o <interfaz de salida>
-s <IP de origen>
-d <IP de destino>
-p <protocolo>
--sport <puerto origen> --dport <puerto destino>
-j <acción>
```

Ejemplos:

```
iptables -A INPUT -i eth0 -p tcp --dport 80 -j ACCEPT
iptables -A OUTPUT -o eth0 -p tcp --sport 80 -j ACCEPT
```

Los nombres de las opciones comentadas son las iniciales de las siguientes palabras inglesas:

i='input'
o='output'
s='source'
d='destination'
p='protocol'
j='jump'
A='add'
F='flush'
P='policy'
m='module'

Bueno ya tenemos hecho nuestro script, lo movemos dentro de la carpeta:

```
#mv iptables /etc/init.d
```

Ahora lo hacemos ejecutable

```
#chmod a+x iptables
```

Y le creamos los enlaces para que se auto-ejecute al iniciar Linux

```
#ln -s /etc/init.d/iptables /etc/rc2.d/S92iptables
#ln -s /etc/init.d/iptables /etc/rc3.d/S92iptables
#ln -s /etc/init.d/iptables /etc/rc5.d/S92iptables
```

Y al reiniciar ya estará funcionando nuestro firewall IPTABLES.

3. Conclusiones

3.a. Evaluación general

Después de este proyecto hemos conseguido un ordenador mucho más potente que cualquier otro de la red, incluso cuando sólo funciona a 100 Mhz. y tiene 16 Mb. de RAM. Por lo tanto, hemos aprovechado muy bien los recursos, sobre todo gracias a Linux. Si hubiésemos instalado otro sistema operativo (como Windows) no se habría podido hacer casi nada en este ordenador.

Hemos elegido siempre la opción correcta al no dejar las configuraciones por defecto y revisarlas, aunque para hacerlo bien hecho sería necesario leerse toda la documentación hasta llegar a entender cada fichero de configuración del sistema. Algunos son muy complicados y ni aún haciendo esto conseguiríamos los resultados que queremos a la primera.

En resumen, el ordenador ha quedado bien, presentable, ya que todo lo que hace, lo hace bien: SSH; PHP, MYSQL, SAMBA y servidor web funcionan perfectamente;

Tenemos un firewall bastante seguro contra la posible intrusión de alumnos u otras personas y ha quedado con los puertos innecesarios cerrados

Pueden aparecer pequeños problemas con el hardware y con los otros sistemas operativos los podemos resolver leyendo toda la documentación incluida y buscando más por Internet. Después de probar cosas durante un tiempo, conseguiremos dejar el servidor 'perfecto'. Funcionará casi igual que ahora, pero ya sabremos si es normal que pase o no.

También hay que destacar que lo hemos hecho todo con software libre y que no nos hemos gastado nada de dinero, aunque todo es completamente legal. Si lo hubiésemos hecho con otros sistemas operativos propietarios tendríamos que haber pagado cantidades enormes de dólares o euros y los resultados no serían mejores que los que hemos conseguido con Linux.

3.b.**Otras utilidades**

Podemos aprovechar mucho un ordenador con Linux instalado, por muy viejo que sea. Como va a estar mucho tiempo encendido sin parar, es ideal para hacer tareas como:

. Descargar archivos grandes y páginas web enteras, dejándolos en un FTP para poder recuperarlos después desde cualquier ordenador de la red. Lo podemos hacer con wget.

. Participar en redes de intercambio de archivos; siempre que sean legales, claro. mldonkey es una buena opción.

. Hacer cálculos matemáticos complejos. Hay muchos programas matemáticos muy avanzados para Linux. Un ejemplo es Scilab. El problema es que 100 Mhz. no es mucha potencia...

. Trabajar en paralelo con otros ordenadores; o sea montar un clúster Beowulf. Muchas empresas y centros de investigación hacen exactamente esto. Cogen componentes viejos que la gente no quiere, los juntan para montar PCs (u otras plataformas), les instalan Linux u otro sistema operativo abierto con soporte de red, y los ponen a trabajar en tareas como el renderizado de imágenes 3D, fractales (matemáticas), cálculo de órbitas de planetas (astrodinámica), flujos turbulentos (hidráulica), y muchas cosas más.

. Programar: en Linux están las mejores herramientas para hacer código estándar, y están soportados la mayoría de los lenguajes: C/C++, Perl, Python, Tcl/Tk, assembler, etc.

. Programar tareas como, por ejemplo, enviar e-mails el primer día de cada mes a los profesores con información que les interese, modificar datos de alguna página externa periódicamente, etc.

También podemos decidirnos por instalar el modo gráfico y utilizar programas profesionales como blender (diseño 3D), gimp (retoque fotográfico), mozilla (navegador web), openoffice (suite de aplicaciones), mplayer (visualizador de vídeos), etc., todo funcionando en un ordenador sencillo pero consiguiendo resultados decentes.

4. Anexos

4.a. Comandos importantes

Éstas son algunas de las órdenes necesarias para utilizar Linux. Podemos ver la ayuda completa con la orden man seguida del nombre del programa.

- ls: hace un listado de los archivos.
Es conveniente utilizar ls -l para ver más propiedades.

- cd directorio: entra en un directorio. Sin parámetros, va al directorio $HOME

- mkdir/rmdir directorio: crear y borrar directorios vacíos

- cp origen destino: copia los archivos de origen al directorio de destino

- mv origen destino: mueve los archivos, o renombra (de hecho es lo mismo)

- rm archivo: borra un archivo.
rm -r para directorios (ir con cuidado)

- cat archivo: muestra el contenido del archivo

- chmod permisos archivo: cambia los permisos de un archivo

- ln origen destino: crea un enlace simbólico. Puede ser 'enlace duro' o normal

- alias orden="orden equivalente": nos ahorra el escribir lo mismo

- date: muestra la hora del sistema. También sirve para cambiarla

- uptime: muestra el tiempo que lleva encendido el PC y la carga de la CPU

- ps: muestra los procesos activos.
Es interesante usar ps aux para verlos todos

- top: monitoriza los procesos

- kill proceso: matar un proceso
- su usuario: permite identificarse como otro usuario (por defecto root)

- reset: si vemos códigos ASCII extraños en la consola se arreglará ejecutándolo

- reboot: reinicia el ordenador

- halt: apaga el ordenador

Otros programas útiles (que puede que haya que instalar) son:

- vim: versión mejorada de vi, editor que está en todos los Linux

- elinks: versión mejorada de lynx, navegador de Internet en modo texto

- wget: para bajar cosas de Internet

- ping: para comprobar si un equipo está activo

- ssh: para conectar a un puerto de un ordenador

- ftp: para conectar a un FTP

- nmap: escaneador de puertos

- nc: netcat, para hacer conexiones de forma más avanzada

- netstat: muestra las conexiones que mantiene el ordenador

- gcc: compilador de C/C++, utilizado para compilar el código fuente de los programas

- tar y gzip: empaquetador y compresor/descompresor de archivos.

- aview: para ver imágenes en códigos ASCII mediante la librería AAlib

- mpg123: para escuchar MP3 desde consola

- mplayer: reproductor de vídeo y DVD.
En consola los podemos ver en ASCII

- fsck: escanea un sistema de archivos y busca y corrige los errores

Es muy importante saber combinar estos programas con 'pipes' (traducido por 'tuberías') y redireccionadores.
Ejemplos (del shell bash):

apt-cache search php | less
Pipe que envía la salida del primer comando al less para poder ver el listado de paquetes por páginas

ps axu | grep ssh
Pipe que muestra los procesos activos que contienen la palabra ssh

nmap pc > puertos_abiertos
Graba el listado de puertos abiertos del host pc en un fichero

nc -l -p 6000 <fichero # Escucha en el puerto 6000 TCP y responde con el contenido del fichero cuando alguien se conecta (útil para transmitir ficheros entre ordenadores sin FTP)

mv b.bak b; cp b b2 # Hace una acción y después la otra “;”

apt-get update && apt-get upgrade
Baja la nueva lista de paquetes y, si no ha habido ningún problema, actualiza los nuevos paquetes

4.b.**Seguridad**

Nuestro ordenador ya es muy seguro; es poco probable que algún alumno lo pueda estropear 'por error' o 'sin querer'. Si queremos ser más paranoicos (nunca es mala idea) podemos hacer muchísimas cosas para mejorar la seguridad del servidor y de todos los datos que hay dentro (tanto pensando en intrusos como en accidentes). Por ejemplo:

Poner contraseña a la BIOS para que el ordenador no se pueda encender sin escribirla.

Hacer que la BIOS no arranque desde disquete.

La caja del ordenador se puede cerrar con llave. (Pero de calidad)

No trabajar como root para las tareas diarias.

No irse y dejar el ordenador con sesiones abiertas en ninguna terminal. Podemos hacer exit para salir o instalar el programa vlock, que la bloquea y pide contraseña.

Tener mucho cuidado con los programas 'setuid root' (con privilegios del usuario root).

Poner buenas contraseñas (sobre todo la de root).

Si queremos que un usuario se pueda conectar habitualmente por SSH o Telnet pero tenemos miedo de que tenga conocimientos suficientes para hacer cosas que no queremos; lo primero será crear un entorno idéntico a un sistema de Linux en su directorio y hacer que al conectarse se haga un chroot a ese directorio para que se convierta en su directorio raíz ('/'). Así, aunque crackease /etc/passwd o consiguiese convertirse en root de otra forma, sólo lo habría hecho dentro de su árbol de ficheros, ya que el fichero /etc/passwd que ha usado está situado realmente en /home/carpeta/etc/passwd

De tanto en tanto hemos de comprobar si realmente es necesario todo lo que hay instalado. Un dpkg -l muestra todos los paquetes que hay en el sistema. Hay que evitar tener cosas que no sepamos qué son, y borrar las que no usamos.

Nos podemos apuntar a las listas de correo de Debian para saber si sale alguna vulnerabilidad importante.

4.c. **Mantenimiento**

El servidor probablemente continuará funcionando a la perfección durante mucho tiempo, y sin tener que apagarlo, pero con el tiempo se irá quedando anticuado. Algunas de las cosas que hemos de hacer, ordenadas por importancia, son:

. Actualizar el sistema con apt-get update; apt-get upgrade -u
Esto lo podemos hacer cada semana o cuando sepamos que han salido versiones nuevas de algún programa que corrigen vulnerabilidades.

. Mirar los logs para ver errores de funcionamiento en algún programa.

. Cada vez que salga una nueva versión estable de Debian, hacer un apt-get dist-upgrade -u para actualizar toda la distribución.

. Ver si hay suficiente espacio en el disco duro, y si no hay, qué es lo que lo ocupa para borrarlo. Si son logs, podemos hacer que se roten.

. Comprobar de vez en cuando el estado del disco duro para ver si hay sectores erróneos con la utilidad fsck. ¡Cuidado!: la unidad tiene que estar desmontada antes de corregir errores (lo podemos hacer arrancando desde disquet).

. Actualizar a IPv6 cuando sea necesario. Poco a poco, todos los programas se van adaptando y muchos ya aceptan direcciones en formato IPv6 (la versión 6 del protocolo IP, ampliado para abastecer a más usuarios de todo el mundo).

. El software puede mantener activo un servidor durante mucho tiempo, pero el hardware se va estropeando. Típicamente, los equipos se apagan como mínimo cada nueve meses para dejar descansar la memoria RAM.

5.d.Glosario de términos

Como para usar Linux hay que tener muchos conocimientos teóricos y prácticos, listaremos aquí las palabras que pueden crear confusión. No salen nombres de marcas comerciales ni de programas; sólo de los más importantes.

- Administrador: persona encargada de llevar una red y sus elementos, entre ellos el ordenador central. En Linux el administrador suele ser el usuario root.

- Buffer overflow (desbordamiento de buffer): bug muy común que consiste en poner más caracteres de los que caben cuando un programa pregunta un dato. Estos caracteres sobrantes pueden escribir partes de la memoria y hacer que se ejecute algún código en concreto (por ejemplo, el programa /bin/sh, para conseguir una cuenta en el sistema). Se pueden explotar de cualquier forma, incluso desde programas sencillos como navegadores web.

- Bug: problema en un programa que hace que no funcione bien. Frecuentemente son amenazas para la seguridad del sistema, y han de ser corregidos rápidamente.

- Demonios (daemons): son programas que se ejecutan en segundo plano al iniciar el ordenador, y que hacen de servidores. Esta es la razón por la que llevan la letra d al final del nombre: telnetd, sshd, ftpd, httpd...

- DoS: Denial of Service (denegación de servicio): resultado de un ataque a un ordenador que hace que deje de ofrecer los servicios habituales. Un ejemplo de DDoS (Distributed DoS) es cuando centenares de ordenadores se conectan a la vez a uno solo hasta que consiguen que se sobrecargue y no trabaje de forma normal.

- IDS: Intrusión Detection System; programa que detecta las acciones no permitidas en una red y actúa en consecuencia. Por ejemplo, si ve que se intenta hacer un ataque por un puerto, lo cierra durante un rato a la IP del atacante y envía un mensaje al móvil del administrador. Son programas muy potentes.

- DNS: Domain Name Server, o servidor de nombres de dominio. Es el ordenador encargado de traducir las direcciones de Internet a direcciones IP. A esta operación se le llama 'resolver' un nombre de host. Por ejemplo, google.com se resuelve a 216.239.51.100. Actualmente hay 14 'root servers' en todo el mundo; el último que se creó está en Madrid.

- Exploit (o xploit): programa sencillo que aprovecha un bug para provocar un error en un programa, haciendo que el usuario o hacker aumente sus privilegios (normalmente la finalidad que se quiere conseguir es el acceso como root). Se pueden encontrar muchos por Internet, todos en lenguaje C.

- FAQ: Frequently Asked Questions (preguntas más frecuentes). Lo encontramos sobre todo en la documentación de programas.

- Firewall (cortafuegos): programa que bloquea el acceso a algunos puertos del ordenador para evitar las intrusiones no deseadas.

- FSF: Free Software Foundation. Fundación iniciada por Richard Stallman creadora del movimiento del software libre, y que comenzó el sistema operativo GNU.

- gcc: GNU C Compiler, el compilador de lenguaje C de la GNU. Aunque parezca que hay muchos compiladores de C y C++, en Linux por defecto está el gcc (o cc) y el g++ (o c++) para C++. cc es un link a gcc, y c++ es un link a g++. Por tanto, sólo está el gcc y el g++

- GID: Group ID, parecido al UID pero aplicado a grupos.

- GNU: siglas de GNU's Not Unix. Es un proyecto de la Free Software Foundation para crear un UNIX libre.

- GPL: General Public License. Una licencia que creó la GNU para proteger el software libre. Debian incorpora únicamente software GPL.

- Hacker: un hacker es una persona que disfruta con su trabajo. Los hackers informáticos son expertos en su materia, y les interesan los temas avanzados y difíciles (como por ejemplo demostrar que pueden entrar en un sitio donde teóricamente nadie puede entrar).

- Host (o hostname): nombre de un equipo que lo identifica en la red. Hay algunos predefinidos, como localhost, pero en /etc/hosts podemos definir más.

- HOWTO: manual de instrucciones sobre cómo hacer algo.

- ISP: Internet Service Provider. Empresa que nos da la conexión a Internet.

- Kernel: núcleo de un sistema operativo, que hace de intermediario entre el usuario y el hardware. En este trabajo se usa el kernel Linux, pero hay otros como el Hurd (aún en una fase muy primitiva).

- Keylogger: tal como dice la palabra, es un programa que queda residente en segundo plano y graba todo lo que se escribe con el teclado. El mejor método para interceptar contraseñas; el inconveniente es que el hacker ha de tener acceso a la máquina porque tiene que volver para revisar el log.

- Lilo: gestor de arranque de los diferentes sistemas operativos instalados. Podemos hacer, por ejemplo, que al encender el ordenador nos pregunte si queremos entrar en Linux 2.4.18, Linux 2.2.20, BSD, BeOS, o cualquier otro kernel o sistema. Todo esto se configura en /etc/lilo.conf y ejecutando después lilo -v para aplicar los cambios.

- Linux: nombre del kernel creado originariamente por Linus Torvalds. "Linux" es el nombre del kernel, y no del sistema operativo, ya que éste se llama GNU. Por eso habría que hablar de GNU/Linux en vez de Linux.

- Montar y desmontar: proceso por el cual un sistema de archivos se hace o se deja de hacer accesible a través de un directorio del sistema de archivos actual. Por ejemplo, podemos montar un disquet en /mnt/floppy. Se usan las órdenes mount y umount.

- Log: registro. En Linux se registran muchas acciones e incidentes, normalmente en el directorio /var/log. Se guarda información sobre los accesos a cada servidor y las operaciones realizadas, los ingresos y salidas de cada usuario (especialmente root), y los problemas que ha habido. Un hacker tiene que conseguir borrar los logs si quiere que nadie sospeche.

- Loopback (lo): interfaz de red que representa nuestro ordenador. Se le asigna la IP 127.0.0.1 (también la 0.0.0.0) y se utiliza en servicios internos.

- PATH: la variable $PATH contiene los directorios en los que se buscará cada programa que ejecutemos antes de comprobar si está en el directorio actual. Por ejemplo, cuando hacemos un ls lo que realmente estamos ejecutando es /bin/ls, porque el directorio /bin está en el PATH. Un hacker puede cambiar el PATH para que programas mal hechos ejecuten comandos falsificados.

- PID: Process ID, o número que tiene cada proceso en ejecución. Lo podemos ver con la orden ps. Es necesario para poder matar (finalizar) un proceso.

- Root: el administrador de un sistema Linux. Se le conoce como superusuario, tiene UID 0, y lo puede hacer absolutamente todo (nadie tiene más privilegios).

- Setuid: cuando se dice que un programa tiene el bit setuid activado, se ejecuta con los privilegios de su propietario. Por ejemplo, el programa passwd, que un usuario normal debe poder ejecutar para cambiar su contraseña, necesariamente ha de acceder a /etc/passwd o /etc/shadow, donde sólo puede escribir root. Por lo tanto, la única solución es el 'setuid root'. Hay que tener mucho cuidado con estos programas; si se encuentra un buffer overflow se pueden conseguir privilegios de root. Para más información: el propietario lo podemos cambiar con chown usuario archivo, el setuid lo podemos poner y quitar con chmod a+s archivo, y podemos ver si un programa lo está con un ls –l (veremos una s en donde normalmente aparecen las x).

- Shell: intérprete de comandos de Linux. Hay muchas shells, la mayoría muy parecidas, pero la más utilizada es bash. Otras son sh, csh, tcsh, bsh, ash, ksh, rbash ... Cada usuario del sistema puede escoger su shell preferida; sólo hay que modificar la entrada apropiada del /etc/passwd

- UID: User ID, o número de identificación de cada usuario del sistema. Se especifica en el tercer parámetro de cada registro de /etc/passwd NOTA: quien tiene UID 0 es superusuario (normalmente llamado root).

- X: el X Windows System (implementado por el programa Xfree86) es un servidor gráfico: un servidor como todos los otros al que los programas se conectan de forma local (también puede ser remota) y muestran una interfaz con botones, menús, ratón, etc.

5. Bibliografía

En las siguientes páginas es donde he encontrado información para rellenar este proyecto, aquí hay partes que he cogido de manuales que lo incluyo en el cd con el proyecto y también tenía algunos conocimientos previos para el desarrollo de MinServ

IP fijas
http://www.openforyou.com/acceso/acceso_tarifaplana.php
http://hosting.lomejordeinternet.net/ip/
http://www.eanet.es/cas/productos/conecta/tarifapl/

Anteproyecto
http://www.fibranet.com/~mikaku/espanol/inventos.html

Debian sarge
http://cdimage.debian.org/pub/cdimage-testing/sarge_d-i/i386/rc2/sarge-i386-netinst.iso

Elegir distribución
http://www.terra.es/tecnologia/articulo/html/tec8837.htm

Configuración servidor WEB, DNS, FTP, POP3 y SMTP
http://www.sorgonet.com/collaborations/servidor-ies/

CONFIGURACIÓN DE UN SERVIDOR GNU/LINUX
http://www.danielclemente.com/servidor/TRcast.txt

Servidor WEB
http://www.infor.uva.es/~jvegas/cursos/buendia/pordocente/node20.html

Instalar apache
http://www.tlm.unavarra.es/asignaturas/pol/2004-05/pol_prac4.html

Iniciar dominios automáticamente
http://wiki.adslayuda.com/linux/index.php/Basico/Demonios

Contraseña root
http://www.mujeresdeempresa.com/tecnologia/tecnologia030201.shtml

Administración de usuarios y grupos
http://www.gulic.org/node/65

Permisos de archivos
http://www.linuxchile.cl/docs.php?op=verVersion&doc=21&id=1

Base de datos -instalación
http://suburbia.sindominio.net/article.php3?id_article=76

Crear la base de datos y ejecutar el script
http://www.webestilo.com/php/php07a.phtml

Problema PHP con MySQL
http://www.todo-linux.com/modules.php?name=Forums&file=viewtopic&t=188

Tablas para el samba Script iptables
http://www.fabio.com.ar/verpost.php?id_noticia=1405

Regla para el ssh de iptables
http://usenet.jyxo.cz/cz.comp.linux.mandrake/0308/iptables-samba.html

Dibujos del proyecto
http://www.google.es

La demás información, los archivos de configuración más importantes, la pagina Web, la base de datos, las fotos utilizadas, los paquetes más actualizados y el kernel que hemos usado lo podéis encontrar en el CD-ROM que tiene adjunto el proyecto.