

**Ingeniería de Servidores (2015-2016)**  
GRADO EN INGENIERÍA INFORMÁTICA  
UNIVERSIDAD DE GRANADA



---

## Práctica 2

---

José Carlos Martínez Velázquez

13 de noviembre de 2015

## Índice

1. Liste los argumentos de yum necesarios para instalar, buscar y eliminar paquetes. 6
2. ¿Qué ha de hacer para que yum pueda tener acceso a Internet? (Pistas: archivo de configuración en /etc, proxy: stargate.ugr.es:3128). ¿Cómo añadimos un nuevo repositorio? 6
3. Indique el comando para buscar un paquete en un repositorio y el correspondiente para instalarlo. (apt) 7
4. Indique qué ha modificado para que apt pueda acceder a los servidores de paquetes a través del proxy. ¿Cómo añadimos un nuevo repositorio? 7
5. ¿Qué diferencia hay entre telnet y ssh? 8
6. ¿Para qué sirve la opción -X? Ejecute remotamente, es decir, desde la máquina anfitriona (si tiene Linux) o desde la otra máquina virtual, el comando gedit en una sesión abierta con ssh. ¿Qué ocurre? 8
7. muestre la secuencia de comandos y las modificaciones a los archivos correspondientes para permitir acceder a la consola remota sin introducir la contraseña. (Pistas: ssh-keygen, ssh-copy-id). 9
8. ¿Qué archivo es el que contiene la configuración de sshd? ¿Qué parámetro hay que modificar para evitar que el usuario root acceda? Cambie el puerto por defecto y compruebe que puede acceder. 11
9. Indique si es necesario reiniciar el servicio ¿Cómo se reinicia un servicio en Ubuntu? ¿y en CentOS? Muestre la secuencia de comandos para hacerlo. 12
10. Muestre los comandos que ha utilizado en Ubuntu Server y en CentOS (aunque en este último puede utilizar la GUI, en tal caso, realice capturas de pantalla) 13
11. Enumere otros servidores web y las páginas de sus proyectos (mínimo 3 sin considerar Apache, IIS ni nginx). 17
12. Compruebe que el servicio está funcionando accediendo a la MV a través de la anfitriona. 17
13. Muestre un ejemplo de uso del comando patch (p.ej. <http://fedoraproject.org/wiki/VMWare>) 18

14. Realice la instalación de esta aplicación y pruebe a modificar algún parámetro de algún servicio. Muestre las capturas de pantalla pertinentes así como el proceso de instalación. 21
15. Instale phpMyAdmin, indique cómo lo ha realizado y muestre algunas capturas de pantalla. Configure PHP para poder importar BDs mayores de 8MiB (límite por defecto). Indique cómo ha realizado el proceso y muestre capturas de pantalla. 23
16. Viste al menos una de las webs de los software mencionados y pruebe las demos que ofrecen realizando capturas de pantalla y comentando qué está realizando. 26
17. Ejecute los ejemplos de find, grep y escriba el script que haga uso de sed para cambiar la configuración de ssh y reiniciar el servicio. 27
18. Escriba el script para cambiar el acceso a ssh usando PHP o Python. 29
19. Abra una consola de Powershell y pruebe a parar un programa en ejecución (p.ej), realice capturas de pantalla y comente lo que muestra. 29
20. Cuestión opcional 1: ¿Qué gestores utiliza OpenSuse? 31
21. Cuestión opcional 2: Instale y pruebe terminator. Con screen, pruebe su funcionamiento dejando sesiones ssh abiertas en el servidor y recuperándolas posteriormente. 31
22. Cuestión opcional 3: Instale el servicio y pruebe su funcionamiento (fail2ban). 35

## Índice de figuras

1.	Conectado al servidor remoto por ssh. . . . .	8
2.	Máquina anfitriona soportando la parte gráfica de gedit. . . . .	9
3.	Generación de public key en dispositivo cliente. . . . .	10
4.	Copia de la llave publica al dispositivo servidor. . . . .	10
5.	Entrando en el servidor sin introducir clave. . . . .	11
6.	Configurando sshd para evitar que root acceda. . . . .	11
7.	Cambiando el puerto por defecto. . . . .	11
8.	Accediendo remotamente a través del puerto 1001. . . . .	12
9.	Comprobación de que apache2 está ejecutándose (desde el servidor). . . . .	13
10.	Comprobación de que apache2 está ejecutándose (modo gráfico). . . . .	14
11.	Comprobación de que apache está ejecutándose (CentOS). . . . .	15
12.	Estableciendo la contraseña de root de MySQL. . . . .	15
13.	Prompt de MySQL. . . . .	16
14.	Comprobación de que se ha instalado MariaDB (CentOS). . . . .	16
15.	Comprobando versión de Python. . . . .	16
16.	Comprobando si se instaló IIS correctamente. . . . .	18
17.	Comprobando si se instaló IIS desde la máquina anfitriona. . . . .	18
18.	Programa con fallo en los límites del vector. . . . .	19
19.	Programa ejemplo arreglado y funcionando. . . . .	19
20.	Visualizando el parche creado. . . . .	20
21.	Aplicado el parche al archivo original y prueba. . . . .	20
22.	Pantalla de login de Webmin. . . . .	21
23.	Panel de Webmin. . . . .	22
24.	Reiniciando un servicio en remoto. . . . .	22
25.	Insalando PhpMyAdmin (I): Instalación para Apache. . . . .	23
26.	Insalando PhpMyAdmin (II): Configuracion dbconfig-common. . . . .	24
27.	Ingresando a PhpMyAdmin. . . . .	24
28.	Pantalla principal de PhpMyAdmin. . . . .	25
29.	Editando el tamaño máximo de BBDD de PhpMyAdmin. . . . .	25
30.	Panel principal de DirectAdmin. . . . .	26
31.	Monitorizando servicios en DirectAdmin. . . . .	26
32.	Panel principal de ISPConfig. . . . .	27
33.	Ausencia de servicios en ISPConfig. . . . .	27
34.	Ejecución del script de configuración de ssh. . . . .	29
35.	Parando procesos en PowerShell. . . . .	30
36.	Probando terminator. . . . .	31
37.	Abriendo sesiones screen en la máquina remota. . . . .	32
38.	Comprobando los procesos que se ejecutan en el servidor. . . . .	32
39.	Comprobando el estado de las sesiones. . . . .	33
40.	Los procesos de una sesión quedan vivos en estado deattached. . . . .	33
41.	Recuperación de una sesión de screen que estaba en estado deattached. . . . .	34
42.	Recuperación de una sesión de screen que estaba en estado attached. . . . .	34

43.	Instalando fail2ban en Ubuntu. . . . .	35
44.	Ausencia de la opción background en fail2ban.conf. . . . .	35
45.	Comprobación de que fail2ban está ejecutándose como servicio. . . . .	36
46.	Activando el baneo por 3 fallos a través de ssh. . . . .	36
47.	Rechazo de la conexión al tercer intento fallido. . . . .	37
48.	Línea de fallo generada por fail2ban. . . . .	37
49.	Error de la herramienta fail2ban-regex. . . . .	37

## **1. Liste los argumentos de yum necesarios para instalar, buscar y eliminar paquetes.**

La sintaxis general de yum es: yum [options] [command] [package ...]. Dejando a un lado las opciones, que se pueden consultar en [5], los comandos necesarios son:

- install paquete1 [paquete2] ... [paquete n]: Instala todos los paquetes pasados como parámetros.
- remove (o erase) paquete1 [paquete2] ... [paquete n]: Eliminará todos los paquetes pasados como parámetros.
- search nombre1 [nombre2] ... [nombre n]: Busca los paquetes que tengan los nombres pasados como parámetros.

(Fuentes: [5])

## **2. ¿Qué ha de hacer para que yum pueda tener acceso a Internet? (Pistas: archivo de configuración en /etc, proxy: stargate.ugr.es:3128). ¿Cómo añadimos un nuevo repositorio?**

Para permitir que yum se conecte a internet haciendo uso de un proxy, tendremos que editar el archivo de configuración yum.conf que se encuentra en el directorio /etc. Una vez encontrado debemos definir el proxy con la palabra clave "proxy" (cuidado, es sensible a mayúsculas/minúsculas) y el nombre de usuario y la contraseña de acceso con las palabras "proxy\_username" y "proxy\_password" respectivamente. Una vez editado, quedaría algo así:

```
# Definicion del proxy que usaremos
proxy=http://stargate.ugr.es:3128
# The account details for yum connections
proxy_username=nombre-usuario-que-se-va-a-usar
proxy_password=contrasenia-acceso-proxy .
```

Para añadir repositorios bastaría con ejecutar el siguiente comando como root:

```
~$ :yum-config-manager --add-repo [url_del_repositorio]
```

La url del repositorio es un link a un archivo con extensión .repo, por ejemplo [http://www.ugr.es/un\\_archivo.repo](http://www.ugr.es/un_archivo.repo)

(Fuentes: [4] [24])

### **3. Indique el comando para buscar un paquete en un repositorio y el correspondiente para instalarlo. (apt)**

Para buscar un paquete hay que utilizar el comando:

```
~$ apt-cache search nombre_del_paquete
```

Una vez que apt ha encontrado el paquete (o paquetes que se corresponden con la cadena nombre\_del\_paquete), se puede utilizar el archiconocido comando:

```
~$ apt-get install nombre_correcto_del_paquete
```

(Fuentes: [27])

### **4. Indique qué ha modificado para que apt pueda acceder a los servidores de paquetes a través del proxy. ¿Cómo añadimos un nuevo repositorio?**

Para permitir que apt pueda conectarse a un proxy, hay que editar el archivo apt.conf ubicado en el directorio /etc/apt. Por cada repositorio al que nos queramos conectar, hay que añadir dos líneas a dicho archivo con el siguiente formato:

```
Acquire :: http :: Proxy "http://usuario:contraseña@proxy.dominio:puerto/";  
Acquire :: https :: Proxy "https://usuario:contraseña@proxy.dominio:puerto/";
```

Por ejemplo, si quisiéramos que apt se conectara al proxy que nos sugiere la cuestión 2, habría que añadir:

```
Acquire :: http :: Proxy "http://usuario:contraseña@stargate.ugr.es:3128/";  
Acquire :: https :: Proxy "https://usuario:contraseña@stargate.ugr.es:3128/";
```

Para añadir repositorios, tenemos dos opciones. La primera es editar el archivo /etc/apt/sources.list, donde añadiremos los repositorios, que como nos dicen en [39], normalmente tienen un formato parecido a este (al menos, para Ubuntu):

```
deb http://PAIS.archive.ubuntu.com/ubuntu/ VersionDeDesarrollo main restricted  
#deb-src http://PAIS.archive.ubuntu.com/ubuntu/ VersionDeDesarrollo main restricted
```

La otra opción (desde la versión 9.10) es con un simple comando:

```
sudo add-apt-repository ppa:[ nombre del repositorio ]
```

(Fuentes: [3] [39] [32])

## 5. ¿Qué diferencia hay entre telnet y ssh?

Telnet es un protocolo de comunicación que permite el acceso y manejo de un dispositivo remoto a través del puerto 23 por defecto. Para usar telnet, es necesario instalar software adicional en los dispositivos que van a intervenir en la conexión. Es necesario instalar el servidor telnet en la máquina que va a ser accedida remotamente y el cliente telnet en la máquina que va a acceder. El gran inconveniente de telnet es que la información que se envía, se envía en texto plano, incluyendo nombres de usuario y contraseñas, lo cual, evidentemente, facilita la intercepción no deseada de estos datos, comprometiendo la seguridad.

SSH tiene el mismo cometido que telnet. Las diferencias son el puerto de operación, que por defecto para SSH es el 22 y que toda la información que se envía está cifrada por una llave sólo conocida por los equipos local y remoto.

(Fuentes: [2] [9])

## 6. ¿Para qué sirve la opción -X? Ejecute remotamente, es decir, desde la máquina anfitriona (si tiene Linux) o desde la otra máquina virtual, el comando gedit en una sesión abierta con ssh. ¿Qué ocurre?

Según el manual del comando ssh la opción -X sirve para habilitar la tecnología X11 forwarding, que sirve para ejecutar aplicaciones en el dispositivo remoto pero la carga gráfica es para el dispositivo local, para ello se va a crear una "cookie mágica" con la ip de la máquina que se está conectando en el archivo /home/nombre\_usuario/.Xauthority. Cabe destacar que la información en éste archivo está cifrada también. Si nunca se han conectado remotamente, dicho archivo no existe y por tanto se crea la primera vez. Para establecer una conexión de este tipo:

Abriremos una conexión ssh mediante el comando

```
~$: ssh -X nombre_usuario_serv:contrasenia_serv@XXX.XXX.XXX.XXX
```

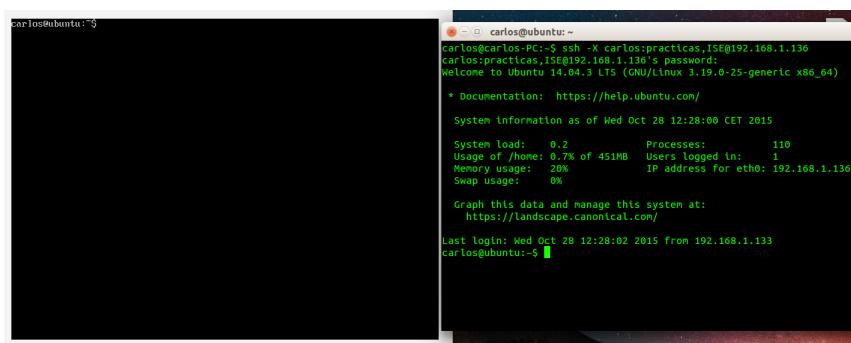


Figura 1: Conectado al servidor remoto por ssh.

La ip de la máquina servidor se debe consultar con el comando ifconfig. Vuelve a pedir la contraseña, pero como la sabemos, se le vuelve a pasar. Ya estamos en la máquina servidor a través de la máquina anfitriona. Si ahora hacemos:

```
~$: sudo gedit unaPrueba.txt
```

Nos damos cuenta de que la ventana de gedit está abierta en la máquina anfitriona, que soporta toda la carga gráfica, mientras las operaciones que se realizan en background se están ejecutando en el servidor (máquina virtual).

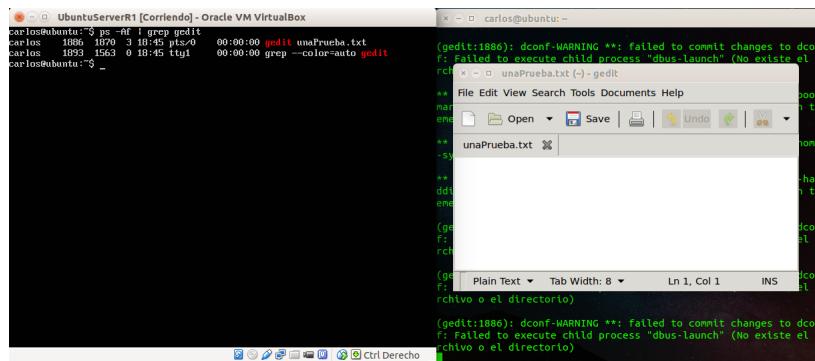


Figura 2: Máquina anfitriona soportando la parte gráfica de gedit.

(Fuentes: [1] [16])

## 7. muestre la secuencia de comandos y las modificaciones a los archivos correspondientes para permitir acceder a la consola remota sin introducir la contraseña. (Pistas: ssh-keygen, ssh-copy-id).

Hay que diferenciar entre cliente (máquina anfitriona) y servidor (máquina virtual). El cliente será el que se conecte al servidor. También hay que diferenciar entre llave pública y privada. La privada es aquélla que sólo tendrá el cliente y la pública será distribuida por todos los equipos a los que el cliente se va a conectar (servidores). A continuación vemos cómo hacerlo.

Paso 1: Vamos a generar la llave pública en el dispositivo cliente. Para ello utilizamos el comando:

```
~$: ssh-keygen -b 4096 -t rsa
```

Lo que estamos diciendo es que vamos a crear una llave de tipo RSA de 4096 bits. Habrá que introducir una frase-contraseña dos veces.

```

carlos@carlos-PC:~$ ssh-keygen -b 4096 -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/carlos/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/carlos/.ssh/id_rsa.
Your public key has been saved in /home/carlos/.ssh/id_rsa.pub.
The key fingerprint is:
37:20:45:4a:75:b3:9b:0b:b7:59:1e:f8:f0:8e:0b:5c carlos@carlos-PC
The key's randomart image is:
---[ RSA 4096]---
. o+ o
. o . o
o . .
. . +
S E o
. = @ .
o + +
. o
o..
-----
carlos@carlos-PC:~$
```

Figura 3: Generación de public key en dispositivo cliente.

Paso 2: Vamos a distribuir la llave pública por los servidores a los que nos vayamos a conectar. En este caso sólo hay que conectarse a uno, así que daremos la llave a éste servidor. Consultaremos la IP del servidor con el comando ifconfig en el mismo, y luego ejecutaremos el comando siguiente en el dispositivo cliente:

```
~$: ssh-copy-id nombre_usuario@XXX.XXX.XXX.XXX
```

Lo que nos daría el siguiente resultado:

```

carlos@carlos-PC:~$ ssh-copy-id carlos@192.168.1.136
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompt-
ed now it is to install the new keys
carlos@192.168.1.136's password:
Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'carlos@192.168.1.136'"
and check to make sure that only the key(s) you wanted were added.

carlos@carlos-PC:~$
```

Figura 4: Copia de la llave publica al dispositivo servidor.

Paso 3: Vamos a comprobar que se ha instalado correctamente. Para ello ejecutamos el comando que nos sugiere la consola al copiar la llave publica:

```
~$: ssh nombre_usuario@XXX.XXX.XXX.XXX
```

Entonces nos debería pedir la frase-contraseña que pusimos en el primer paso. Si volvemos a ejecutar dicho comando, vamos a acceder sin tener que introducir ningún tipo de clave:

```

carlos@carlos-PC:~$ ssh carlos@192.168.1.136
Welcome to Ubuntu 14.04.3 LTS (GNU/Linux 3.19.0-25-generic x86_64)

 * Documentation: https://help.ubuntu.com/
 
 System information as of Wed Oct 28 13:28:53 CET 2015

 System load: 0.0          Processes:      111
 Usage of /home: 0.7% of 451MB  Users logged in:   1
 Memory usage: 20%          IP address for eth0: 192.168.1.136
 Swap usage:  0%
 
 Graph this data and manage this system at:
 https://landscape.canonical.com/
 
 Last login: Wed Oct 28 13:28:53 2015 from 192.168.1.133
carlos@ubuntu:~$ 

```

Figura 5: Entrando en el servidor sin introducir clave.

(Fuentes: [6] [18])

## **8. ¿Qué archivo es el que contiene la configuración de sshd? ¿Qué parámetro hay que modificar para evitar que el usuario root acceda? Cambie el puerto por defecto y compruebe que puede acceder.**

El archivo que contiene la configuración de sshd es /etc/ssh/sshd\_config. Para evitar que el usuario root acceda, hay que poner la opción PermitRootLogin a no:

```

# Authentication:
LoginGraceTime 120
PermitRootLogin no
StrictModes yes

```

Figura 6: Configurando sshd para evitar que root acceda.

Para cambiar el puerto, hay que irse a la opción port y cambiarlo simplemente. En mi caso utilizaré el 1001:

```

# What ports, IPs and protocols we listen for
Port 1001

```

Figura 7: Cambiando el puerto por defecto.

Ahora, accediendo desde la máquina anfitriona, veré si puedo seguir accediendo y, en caso afirmativo, volveré a consultar el archivo /etc/ssh/sshd\_config para ver si está en el puerto 1001. Efectivamente, podemos comprobar que es así:

The screenshot shows two terminal windows. The left window displays the output of the command 'ifconfig' on an Ubuntu system. It lists two interfaces: 'eth0' (Link encap:Ethernet, dirección IP 192.168.1.136) and 'lo' (Link encap:Loopback local). The right window shows the contents of the '/etc/ssh/sshd\_config' file being edited with nano 2.2.6. The configuration includes settings for port 1001, host keys, and privilege separation.

```

carlos@ubuntu:~$ ifconfig
eth0      Link encap:Ethernet  dirección IP 192.168.1.136
          Direc. inet6: fe80::a00:27ff:fe22:9044/64 Alcance:Enlace
          Dirección inet6: ::1/128 Alcance:Anfitrón
          ACTIVO DIFUSIÓN FUNCIONANDO MTU:1500 Métrica:1
          Paquetes RX:35 errores:0 perdidos:0 overrunns:0 frame:0
          Paquetes TX:36 errores:0 perdidos:0 overrunns:0 carrier:0
          colisiones:0 long_colatX:0
          Bytes RX:4043 (4.0 KB) TX bytes:3574 (3.5 KB)

lo        Link encap:Loopback local
          Direc. inet6: ::1/128 Alcance:Anfitrón
          Dirección inet6: ::1/128 Alcance:Anfitrón
          ACTIVO BUCLE FUNCIONANDO MTU:65536 Métrica:1
          Paquetes RX:0 errores:0 perdidos:0 overrunns:0 frame:0
          Paquetes TX:0 errores:0 perdidos:0 overrunns:0 carrier:0
          colisiones:0 long_colatX:0
          Bytes RX:0 (0.0 B) TX bytes:0 (0.0 B)

carlos@ubuntu:~$


GNU nano 2.2.6   Archivo: /etc/ssh/sshd_config

# Package generated configuration file
# See the sshd_config(5) manpage for details

# What ports, IPs and protocols we listen for
Port 1001
# Use these options to restrict which interfaces/protocols ssh$ 
#listenAddress ::

#listenAddress 0.0.0.0
Protocol 2
# HostKeys for protocol version 2
HostKey /etc/ssh/ssh_host_rsa_key
HostKey /etc/ssh/ssh_host_dsa_key
HostKey /etc/ssh/ssh_host_ecdsa_key
HostKey /etc/ssh/ssh_host_ed25519_key
#Privilege Separation is turned on for security
UserPrivilegeSeparation yes

# Lifetime and size of ephemeral version 1 server key
KeyRegenerationInterval 3600

```

Figura 8: Accediendo remotamente a través del puerto 1001.

(Fuentes: [29])

## 9. Indique si es necesario reiniciar el servicio ¿Cómo se reinicia un servicio en Ubuntu? ¿y en CentOS? Muestre la secuencia de comandos para hacerlo.

En la cuestión anterior, se podía acceder sin reiniciar el servicio ssh después de cambiar el puerto y la opción de acceso root. Después de reiniciarlo, dejando el puerto en 1001, todo sigue igual, es decir, que se puede seguir accediendo sin problema, así que no parece ser necesario tener que reiniciar el servicio.

En ubuntu, la lista de servicios que hay corriendo se consulta en el directorio /etc/init.d. Para reiniciar un servicio en Ubuntu, basta con ejecutar los comandos:

```

~$: ls /etc/init.d/ #Vamos a ver que servicios hay en ejecucion
~$: sudo /etc/init.d/[servicio_a_reiniciar] restart #Reiniciamos el
servicio servicio_a_reiniciar

```

Para hacer ésto mismo en CentOS, se utiliza otro comando. El directorio /etc/init.d ya no nos será útil, sino que utilizaremos la herramienta systemctl. Hay que ejecutar los siguientes comandos:

```

~$: systemctl # (o bien systemctl list-units, es lo mismo) Nos va a ofrecer
un listado de todos los servicios, o units, como los llama CentOS, que
hay ejecutandose en el sistema.
~$: sudo systemctl restart [servicio_a_reiniciar] #Reiniciamos el servicio
servicio_a_reiniciar

```

(Fuentes: [33] [26])

**10. Muestre los comandos que ha utilizado en Ubuntu Server y en CentOS (aunque en este último puede utilizar la GUI, en tal caso, realice capturas de pantalla)**

LAMP es la unión de las siglas de Linux Apache MySQL PHP. Aunque parece un paquete, hay que instalar cada herramienta por separado. Para instalar LAMP completo:

- Paso 1. Instalación de Apache

**Ubuntu Server:**

El primer paso es instalar Apache. Como vamos a instalar a través del gestor de paquetes apt, actualizaremos con:

```
~$: sudo apt-get update
```

Una vez hecho, instalaremos apache2, ejecutando:

```
~$: sudo apt-get install apache2
```

Será necesario probar la instalación de apache, para lo que hay que abrir un navegador e ingresar a la IP de la máquina. Dado que Ubuntu server se ejecuta en modo texto, no podemos hacerlo, pero podemos ver si el servicio está ejecutándose. Para ello ejecutamos el siguiente comando:

```
~$: service --status-all | grep a #Buscamos todos los servicios que contienen una a
```

```
carlos@ubuntu:~$ service --status-all | grep a
[ + ] acpid
[ + ] apache2
[ - ] apparmor
[ ? ] apport
```

Figura 9: Comprobación de que apache2 está ejecutándose (desde el servidor).

Vemos que apache tiene un '+' y eso significa que el proceso está corriendo. Si queremos ver que apache está corriendo (en modo gráfico) en cualquier máquina que forme parte de la red, ingresamos desde un navegador a la IP de la máquina donde ha sido instalado apache y vemos que efectivamente está instalado:

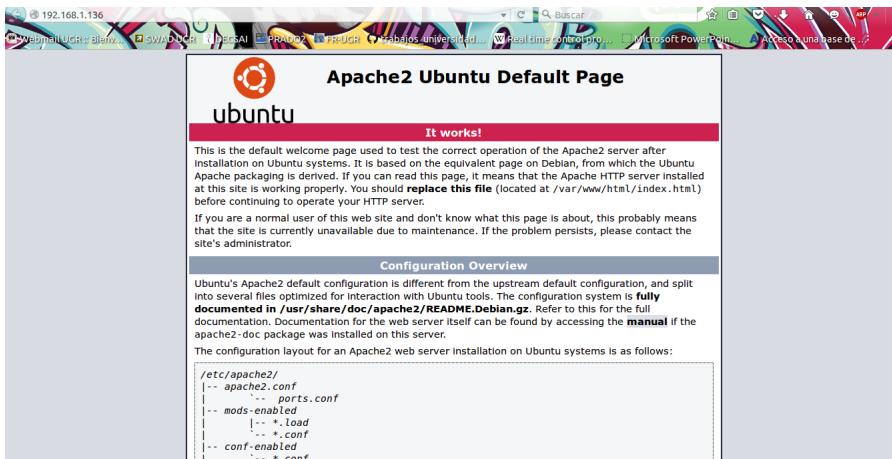


Figura 10: Comprobación de que apache2 está ejecutándose (modo gráfico).

Como posteriormente instalaremos python, es necesario instalar algunas herramientas para que apache y python se entiendan. Para ello utilizaremos apt e instalaremos los siguientes paquetes: python-setuptools y libapache2-mod-wsgi. Posteriormente reiniciaremos el servicio de apache para que los cambios surtan efecto.

```
~$: sudo apt-get install python-setuptools libapache2-mod-wsgi
~$: sudo service apache2 restart
```

### CentOS:

No voy a utilizar la interfaz gráfica para éste proceso. Lo primero es instalar apache. Para ello ejecutaremos los siguientes comandos:

```
~$: sudo yum install httpd

~$: sudo systemctl start httpd #Comenzar el servicio apache
~$: sudo systemctl enable httpd #Activar el servicio apache
~$: sudo firewall-cmd --zone=public --add-port=80/tcp --permanent #
    Establecer el puerto 80 por defecto
~$: sudo firewall-cmd --reload #Recargar la configuracion.
```

Para comprobar que el servicio está ejecutándose ejecutamos:

```
~$: sudo systemctl status httpd
```

Y obtendremos lo siguiente:

```
[carlos@localhost ~]$ sudo systemctl status httpd
httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled)
     Active: active (running) since dom 2015-11-08 10:23:05 CET; 3min 23s ago
       Main PID: 3471 (httpd)
          Status: "Total requests: 0; Current requests/sec: 0; Current traffic: 0 B/s"
             CGroup: /system.slice/httpd.service
                   ├─3471 /usr/sbin/httpd -DFOREGROUND
                   ├─3472 /usr/sbin/httpd -DFOREGROUND
                   ├─3473 /usr/sbin/httpd -DFOREGROUND
                   ├─3474 /usr/sbin/httpd -DFOREGROUND
                   ├─3475 /usr/sbin/httpd -DFOREGROUND
                   └─3476 /usr/sbin/httpd -DFOREGROUND

nov 08 10:23:05 localhost.localdomain systemd[1]: Starting The Apache HTTP Se...
nov 08 10:23:05 localhost.localdomain httpd[3471]: AH000558: httpd: Could not ...
nov 08 10:23:05 localhost.localdomain systemd[1]: Started The Apache HTTP Ser...
Hint: Some lines were ellipsized, use -l to show in full.
[carlos@localhost ~]$ _
```

Figura 11: Comprobación de que apache está ejecutándose (CentOS).

- Paso 2. Elegir entre MySQL, MariaDB o PostgreSQL

#### Ubuntu Server:

Aunque algunos puedan discrepar, personalmente pienso que MySQL es la base de datos más extendida en los paquetes de servidores web, por eso elegiremos éste. Para instalar MySQL ejecutaremos el siguiente comando:

```
~$: sudo apt-get install mysql-server
```

Durante la instalación, debemos establecer una contraseña para el usuario root de MySQL, repitiéndola dos veces.

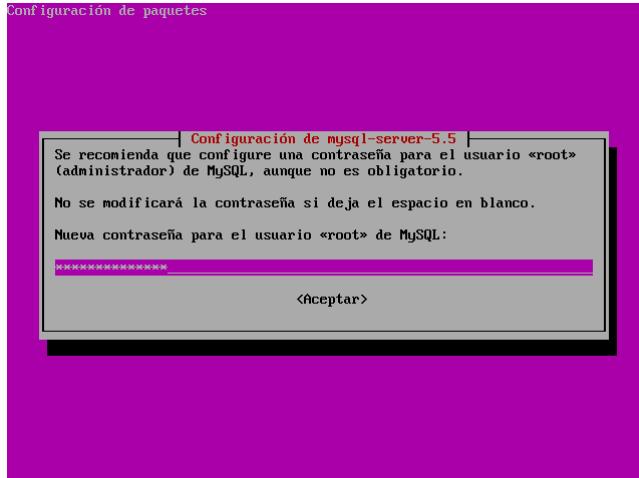


Figura 12: Estableciendo la contraseña de root de MySQL.

Para probar que MySQL está corriendo, ejecutaremos el siguiente comando:

```
~$: sudo mysql -u root -p
```

Deberíamos obtener el siguiente prompt:

```
carlos@ubuntu:~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 42
Server version: 5.5.46-0ubuntu0.14.04.2 (Ubuntu)

Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> _
```

Figura 13: Prompt de MySQL.

A partir de aquí podemos empezar a crear usuarios, tablas, registros...

**CentOS:**

Dado que yum no encuentra el paquete mysql-server, instalaremos MariaDB. Ejecutamos los siguientes comandos:

```
~$: sudo yum install mariadb-server
~$: sudo systemctl start mariadb
~$: sudo systemctl enable mariadb
~$: sudo mysql_secure_installation
```

```
Cleaning up...
All done! If you've completed all of the above steps, your MariaDB
installation should now be secure.

Thanks for using MariaDB!
[carlos@localhost ~]$ _
```

Figura 14: Comprobación de que se ha instalado MariaDB (CentOS).

■ Paso 3. Elegir entre PHP, Python y Perl

**Ubuntu Server:**

Aunque PHP es el lenguaje más extendido, personalmente me llevo mejor con Python, por eso lo elijo. Normalmente, Python es instalado por defecto con cualquier distribución Ubuntu, por lo que se puede comprobar la versión con el comando python, en nuestro caso la 2.7.6.

```
carlos@ubuntu:~$ python
Python 2.7.6 (default, Jun 22 2015, 17:58:13)
[GCC 4.8.2] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> _
```

Figura 15: Comprobando versión de Python.

### **CentOS:**

Al igual que en Ubuntu Server podemos comprobar que Python ha sido instalado. Para ello ejecutamos el comando python.

(Fuentes: [8] [34] [22] [42])

## **11. Enumere otros servidores web y las páginas de sus proyectos (mínimo 3 sin considerar Apache, IIS ni nginx).**

Varios de los servidores web más famosos sin considerar los anteriormente citados son:

- Cherokee: Servidor web openSource escrito completamente en C. La página de su proyecto es: <http://www.cherokee-project.com/>
- Tomcat: (o Jakarta Tomcat) Creado por Sun microsystems, es un servidor web que funciona como un contenedor de servlets. La página de su proyecto es: <http://tomcat.apache.org>
- Lighttpd: Servidor web openSource orientado sobre todo a la rapidez y la seguridad, consumiendo pocos recursos. Funciona en GNU/Linux y Unix de forma oficial. La página de su proyecto es: [lighttpd.net](http://lighttpd.net)
- Thhttpd: Al igual que el anterior, es liviano y rápido, orientado a cumplir los requisitos mínimos de HTTP. Es openSource. La página de su proyecto es <http://www.acme.com/software/thhttpd/>

(Fuentes: [38] [36] [41] [37] [40])

## **12. Compruebe que el servicio está funcionando accediendo a la MV a través de la anfitriona.**

Para instalar IIS haremos clic en "Aregar roles y características", posteriormente seleccionaremos "Instalación basada en características o en roles", seleccionamos nuestro servidor mediante la opción "seleccionar un servidor o un grupo de servidores" y siguiente. En la siguiente pantalla buscaremos "Servidor web (IIS)" y pulsamos el botón Agregar características. Dos pantallas después seleccionaremos el servicio de rol "Restricciones de IP y dominio", dejando lo demás tal cual está. En la siguiente pantalla pulsaremos Instalar.

El servicio está instalado. Ahora comprobaremos si está ejecutándose. Para ello abrimos un navegador y tecleamos <http://localhost/>. Deberíamos ver lo siguiente:



Figura 16: Comprobando si se instaló IIS correctamente.

Si consultamos la IP en Windows y la consultamos a través de un navegador en la máquina anfitriona:

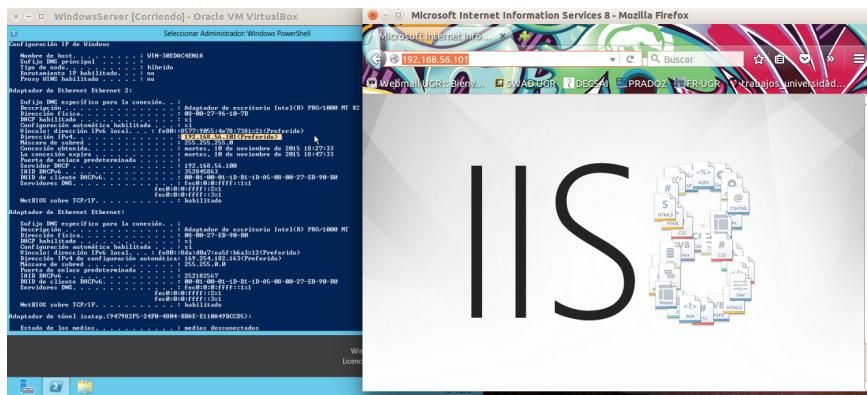


Figura 17: Comprobando si se instaló IIS desde la máquina anfitriona.

(Fuentes: [15])

### 13. Muestre un ejemplo de uso del comando patch (p.ej. <http://fedoraproject.org/wiki/VMWare>)

Imaginemos un programa, en el que nos hemos equivocado en cualquier cosa y que falla, ya sea porque no se ejecuta o tiene un agujero de seguridad, etc. como el siguiente (éste programa podría ser algo tan complejo como el kernel de linux):

```

OPEN FILES
x ejemplo.cpp
x fix_ejemplo.cpp
x ejemplo.cpp

ejemplo.cpp
1 #include <iostream>
2 #include <vector>
3
4 using namespace std;
5
6 int main(){
7     vector<int> numeros(10);
8     for(int i=0;i<numeros.size();i++){
9         numeros[i]=i+1;
10        cout<<numeros[i];
11    }
12 }
13

*** Error in `/home/carlos/Escritorio/ejemplo': free(): invalid next size (fast):
0x4567891011/bin/bash: linea 1: 8820 Abortado ('core' generado) "/home/
carlos/Escritorio/ejemplo"
(Finished in 0.5s with exit code 134)
[shell cmd: g++ "/home/carlos/Escritorio/ejemplo.cpp" -o "/home/carlos/Escritorio/
ejemplo" && "/home/carlos/Escritorio/ejemplo"]
[dir: /home/carlos/Escritorio]
[path: /usr/local/sbin:/usr/local/bin:/usr/sbin:/bin:/usr/games:/usr/local/
games]

Line 9, Column 34
Tab Size: 4
C++

```

Figura 18: Programa con fallo en los límites del vector.

En éste caso, hemos excedido el límite 10 "sin querer". Para éste caso sería muy sencillo borrar el igual que sobra y recompilar, pero si el programa que falla es el kernel de linux, no sería tan accesible, ni tan sencillo, para eso existe patch. Vamos a arreglar éste problema mediante patch.

Para ello, lo primero que hay que hacer es un backup del programa que falla y arreglar el código.

```
~$: cp ejemplo.cpp fix_ejemplo.cpp
```

Se arregla el problema y se prueba, para comprobar que cuando al original se le apliquen éstos cambios, funcione:

```

OPEN FILES
x exemplo.cpp
x fix_ejemplo.cpp
x exemplo.cpp

ejemplo.cpp
1 #include <iostream>
2 #include <vector>
3
4 using namespace std;
5
6 int main(){
7     vector<int> numeros(10);
8     for(int i=0;i<numeros.size();i++){
9         numeros[i]=i+1;
10        cout<<numeros[i];
11    }
12 }
13

12345678910[Finished in 0.3s]

1 characters selected
Tab Size: 4
C++

```

Figura 19: Programa ejemplo arreglado y funcionando.

Ahora crearemos el parche (patch) con el comando diff, de la siguiente forma:

```
~$: diff -u ejemplo.cpp fix_ejemplo.cpp > ejemplo.patch
```

Con el que tendremos un archivo ejemplo.patch que contiene lo siguiente:

```

OPEN FILES
x ejemplo.patch
x ejemplo.patch
1 |-- ejemplo.cpp 2015-11-11 11:38:06.888781625 +0100
2 +++ fix_ejemplo.cpp 2015-11-11 11:47:31.196769291 +0100
3 @@ -6,7 +6,7 @@
4 int main(){
5     vector<int> numeros(10);
6
7     - for(int i=0;i<numeros.size();i++){
8     + for(int i=0;i<numeros.size();i++){
9         numeros[i]=i+1;
10        cout<<numeros[i];
11    }
12

```

Line 1, Column 1 Tab Size: 4 Diff

Figura 20: Visualizando el parche creado.

Donde la línea roja significa que se ha borrado (-) y la línea verde significa que se ha añadido (+). Dado que patch sabe que archivo es el original (—) y cual el arreglado (+++), si ejecutamos el comando:

```
~$: patch < ejemplo.patch
```

Estamos aplicando el parche al archivo `ejemplo.cpp` con lo que contiene `fix_ejemplo.cpp`. Así que, si volvemos a ver el archivo `ejemplo.cpp`, podremos ver como todo está arreglado y funciona como `fix_ejemplo.cpp`:

```

OPEN FILES
x ejemplo.cpp
x exemplo.cpp
1 #include <iostream>
2 #include <vector>
3
4 using namespace std;
5
6 int main(){
7     vector<int> numeros(10);
8
9     for(int i=0;i<numeros.size();i++){
10        numeros[i]=i+1;
11        cout<<numeros[i];
12    }
13

```

Line 1, Column 1 Tab Size: 4 C++

Figura 21: Aplicado el parche al archivo original y prueba.

(Fuentes: [28])

**14. Realice la instalación de esta aplicación y pruebe a modificar algún parámetro de algún servicio. Muestre las capturas de pantalla pertinentes así como el proceso de instalación.**

Para instalar Webmin en el Ubuntu Server, lo primero que debemos hacer es añadir los repositorios al archivo /etc/apt/sources.list, por lo que editaremos dicho archivo añadiendo las siguientes líneas (para poder copiar-pegar, haremos una conexión ssh con la opción -X y usar gedit en modo gráfico):

```
deb http://download.webmin.com/download/repository sarge contrib  
deb http://webmin.mirror.somersettechsolutions.co.uk/repository sarge contrib
```

Una vez añadido, habrá que decirle a apt que los repositorios que hemos añadido son de confianza, ésto se hace con el siguiente comando:

```
~$ wget -q http://www.webmin.com/jcameron-key.asc -O- | sudo apt-key add -
```

Una vez que se ha hecho ésto, actualizaremos los repositorios de apt para poder instalar con apt-get. Para ello haremos:

```
~$ sudo apt-get update  
~$ sudo apt-get install webmin
```

La instalación de webmin comienza y ocupará 157MB. Para comprobar que funciona, debemos ingresar a la IP del servidor desde el puerto 10000, en el navegador de la máquina anfitriona, así: <https://xxx.xxx.xxx.xxx:10000>. Puede que nos diga que el certificado no es de confianza y haya que añadir confirmar una excepción de seguridad, pero posteriormente nos llevará a la pantalla de login a la que habrá que acceder con el usuario y la contraseña del servidor:

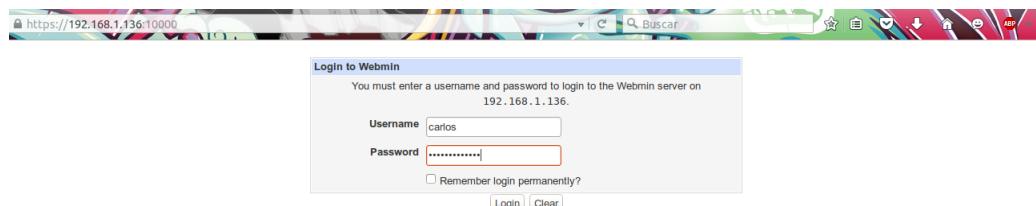


Figura 22: Pantalla de login de Webmin.

La página de configuración de Webmin es la siguiente:

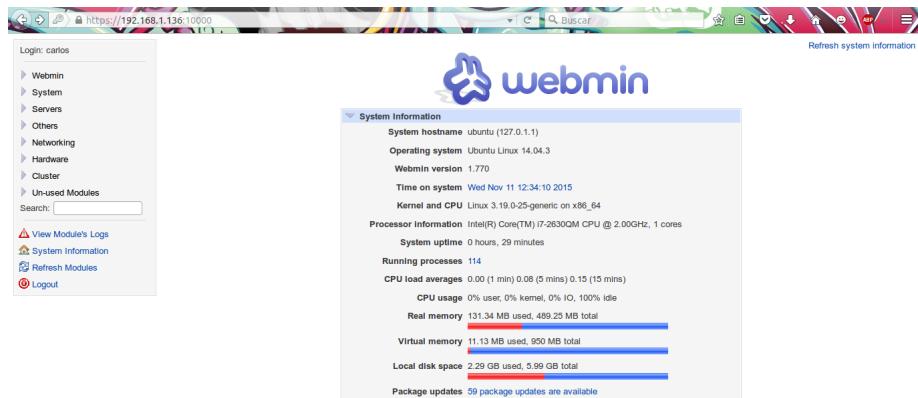


Figura 23: Panel de Webmin.

Por tocar algo, podemos reiniciar servicios que están corriendo en el servidor, por ejemplo, Apache:

	Setting	Value
save kernel messages	<input type="checkbox"/>	Yes
D-Bus system message bus	<input type="checkbox"/>	Yes
enable encrypted block devices	<input type="checkbox"/>	Yes
cryptisks	<input type="checkbox"/>	No
enable remaster boot-time encrypted block devices	<input type="checkbox"/>	Yes
regular root-group program processing daemon	<input type="checkbox"/>	Yes
cron	<input type="checkbox"/>	Yes
ctrl-alt-delete	<input type="checkbox"/>	No
emergency keypress handling	<input type="checkbox"/>	Yes
container-detect	<input type="checkbox"/>	No
Traces if update is running in a container	<input type="checkbox"/>	Yes
console-setup	<input type="checkbox"/>	No
set console keymap	<input type="checkbox"/>	Yes
console-font	<input type="checkbox"/>	No
set console font	<input type="checkbox"/>	No
console	<input type="checkbox"/>	No
checkroot.sh	<input type="checkbox"/>	Yes
Signal sysvinit that the rootfs is mounted	<input type="checkbox"/>	Yes
checkroot-bootclean.sh	<input type="checkbox"/>	Yes
checks.sh	<input type="checkbox"/>	Yes
bootmisc.sh	<input type="checkbox"/>	Yes
defsched	<input type="checkbox"/>	Yes
ad	<input type="checkbox"/>	Yes
automount	<input type="checkbox"/>	Yes
crash	<input type="checkbox"/>	Yes
egpmor	<input type="checkbox"/>	No
AppArmor init script. This script loads all AppArmor profiles	<input checked="" type="checkbox"/>	Unknown
apache2	<input checked="" type="checkbox"/>	Start the web server and associated helpers
acpid	<input type="checkbox"/>	ACPI daemon

Select all | Invert selection | Create a new update service.

Start | Stop | Restart | Start On Boot | Disable On Boot | Start Now and On Boot | Disable Now and On Boot | Reboot System | Shutdown System

(a) Seleccionando un servicio para reiniciar.

```

Restarting service apache2...
 * Starting web server apache2
[warn] [pid 24655] apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the 'ServerName' directive glob
...done.
Return to backup and shutdown actions

```

(b) Reiniciando apache.

Figura 24: Reiniciando un servicio en remoto.

(Fuentes: [20])

**15. Instale phpMyAdmin, indique cómo lo ha realizado y muestre algunas capturas de pantalla. Configure PHP para poder importar BDs mayores de 8MiB (límite por defecto). Indique cómo ha realizado el proceso y muestre capturas de pantalla.**

Para instalar PhpMyAdmin debemos ejecutar los siguientes comandos:

```
~$ sudo apt-get update  
~$ sudo apt-get install phpmyadmin
```

La instalación comienza y ocupará 49,5MB. La primera pantalla de la instalación nos pregunta qué servicio web queremos que ejecute PhpMyAdmin, seleccionaremos Apache.

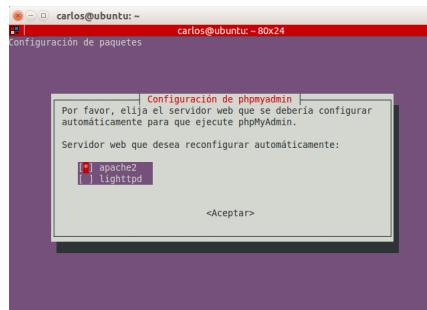
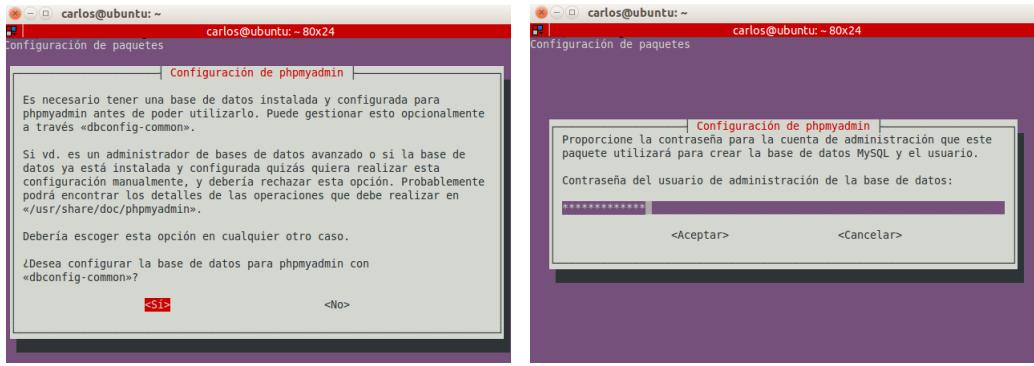


Figura 25: Instalando PhpMyAdmin (I): Instalación para Apache.

El siguiente paso para instalar PhpMyAdmin es configurar la base de datos para PhpMyAdmin con dbconfig-common. Seleccionaremos Sí. Lo primero que habrá que hacer es establecer una contraseña y repetirla, después, la instalación continuará hasta terminar.



(a) Seleccionar configuración con dbconfig-common.

(b) Estableciendo contraseña.

Figura 26: Insalando PhpMyAdmin (II): Configuracion dbconfig-common.

Para utilizar PhpMyAdmin, hay que activar explícitamente la extensión `php5-mcrypt` y posteriormente reiniciar apache, lo que se hace con los siguientes comando:

```
~$ sudo php5enmod mcrypt
~$ sudo service apache2 restart
```

Ya está todo. Para loguearnos en PhpMyAdmin, basta con ingresar la dirección `http://xxx.xxx.xxx.xxx/phpmyadmin` e introducir el usuario root con la contraseña que definimos durante la instalación, con lo que obtendremos lo siguiente:

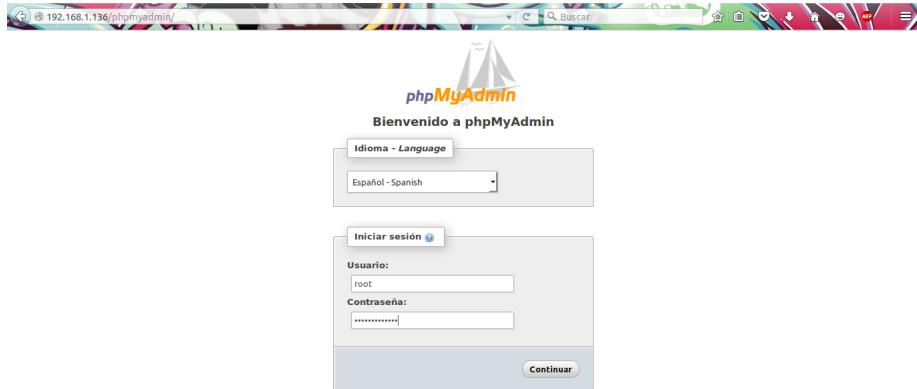


Figura 27: Ingresando a PhpMyAdmin.

Una vez logueados, la pantalla principal es la siguiente:

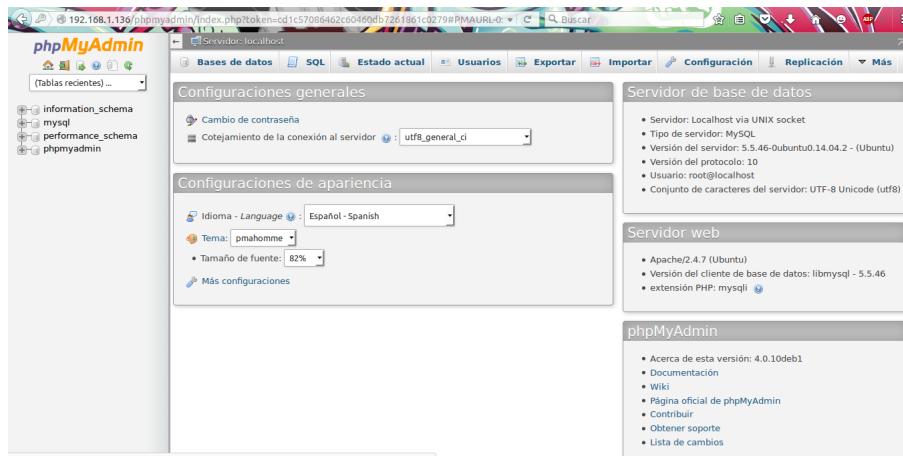


Figura 28: Pantalla principal de PhpMyAdmin.

Para permitir la importación de una base de datos de más de 8MB, hay que editar el archivo etc/php5/apache2/php.ini. Por defecto PhpMyAdmin permite importar bases de datos de 2MB máximo. Vamos a poner un máximo de 10MB, por lo que editaremos la línea max\_file\_size=10M. Una vez editado debe quedar así:

```

php.ini (/etc/php5/apache2) - gedit
File Edit View Search Tools Documents Help
Open Save Undo .ini Tab Width: 8 Ln 805, Col 25 INS
php.ini

; Temporary directory for HTTP uploaded files (will use system
; default if not
; specified).
; http://php.net/upload-tmp-dir
;upload_tmp_dir =

; Maximum allowed size for uploaded files.
; http://php.net/upload-max-filesize
upload_max_filesize = 10M

; Maximum number of files that can be uploaded via a single request
max_file_uploads = 20

;;;;;;
; Fopen wrappers ;;
;;;;;;

; Whether to allow the treatment of URLs (like http:// or ftp://)

```

Figura 29: Editando el tamaño máximo de BBDD de PhpMyAdmin.

Hecho esto, reiniciamos el servicio de apache y ya podríamos importar una base de datos de hasta 10MB.

(Fuentes: [19] [35])

## 16. Viste al menos una de las webs de los software mencionados y pruebe las demos que ofrecen realizando capturas de pantalla y comentando qué está realizando.

Para probar DirectAdmin nos loguaremos en la demo como administradores. Para ello sólo basta con introducir el nombre de usuario demo\_admin y la contraseña demo. Una vez dentro nos encontraremos con éste panel:

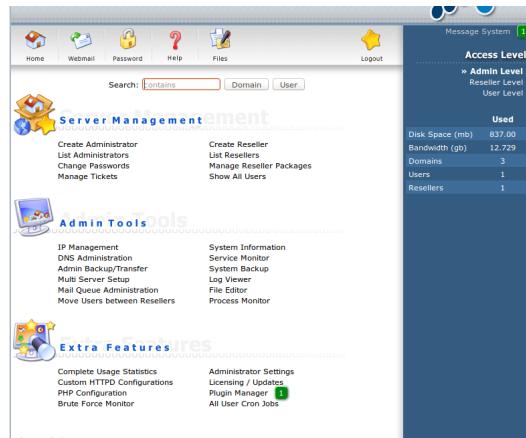


Figura 30: Panel principal de DirectAdmin.

Aunque en el panel podemos ver muchas cosas que se pueden hacer, la funcionalidad de DirectAdmin no me parece tan alta, dado que es una demo. Por tocar algo, he intentado monitorizar los servicios y reiniciar el server, pero lo segundo no fue posible porque dicha opción está capada.

Service	Status	Memory Usage	Start	Stop	Restart	Reload
directadmin	directadmin (pid 26222 15052.)	25.2 MB	Start	Stop	Restart	Reload
dovecot	dovecot (pid 13344)	90.8 MB	Start	Stop	Restart	
exim	exim (pid 6601 7721)	1.46 MB	Start	Stop	Restart	Reload
httpd	httpd (pid 26876)	229.0 MB	Start	Stop	Restart	Reload
mysqld	mysqld (pid 18401 22497 29354 19265 6354)	346.8 MB	Start	Stop	Restart	Reload
named	named (pid 4735)	222.9 MB	Start	Stop	Restart	Reload
proftpd	proftpd (pid 6199 6870 14412 17940 15460)	2.99 MB	Start	Stop	Restart	
sshd	sshd (pid 22783 28387 10343 17001 9769)	7.02 MB	Start	Stop	Restart	Reload

Your password:

Unable To Execute Your Command  
That feature has been disabled for the demo

(a) Servicios en ejecución.

(b) Intento fallido de reiniciar el servidor.

Figura 31: Monitorizando servicios en DirectAdmin.

ISP config es otra herramienta, a la que podemos ingresar de una manera parecida (admin/demo) y su panel principal es éste:

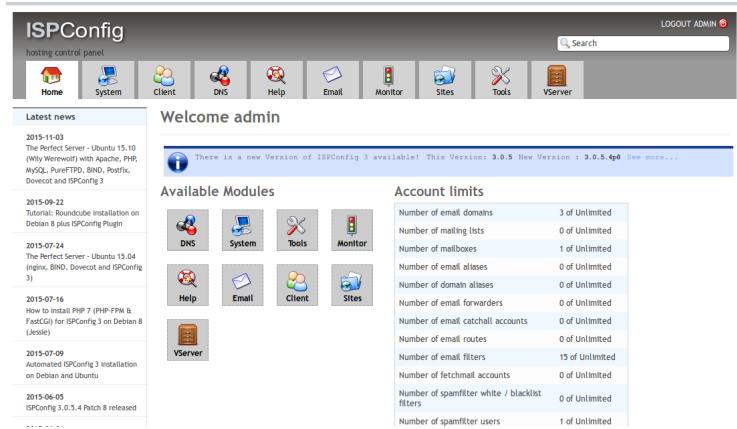


Figura 32: Panel principal de ISPConfig.

He intentado hacer lo mismo que con DirectAdmin, es decir, monitorizar servicios, ¡Pero ni siquiera aparecen!:



Figura 33: Ausencia de servicios en ISPConfig.

(Fuentes: [7] [14])

## 17. Ejecute los ejemplos de find, grep y escriba el script que haga uso de sed para cambiar la configuración de ssh y reiniciar el servicio.

Es posible reemplazar una linea de texto por otra en cualquier archivo, para ello utilizamos el comando sed, con la siguiente sintaxis:

```
~$ :sed -i 's/[linea_original]/[nueva_linea]/' [archivo_de_texto]
```

Para cambiar la configuración de ssh (parte de ella, ya que disponemos de muchas opciones) utilizaremos el siguiente script python:

```
import os
os.system("sed -i 's/[linea_original]/[nueva_linea]/' [archivo_de_texto]")
```

Por lo tanto, el script resultante será:

```
# -*- coding: utf-8 -*-
import os;
import sys;

if len(sys.argv) < 3:
    print("El script necesita 3 parametros")
    print("Uso: python scriptAcceso [puerto_ssh] [permitir_log_root(yes/no/without-password)] [permitir_password_vacias(yes/no)]")
else:

    #Copia de seguridad por si acaso...
    print("Creando copia de seguridad de la configuracion de ssh...")
    os.system("cp /etc/ssh/sshd_config /etc/ssh/sshd_config_backup")
    print("Hecho.")

    #Si peta en la siguiente linea, es porque el argumento puerto no
    #es un numero
    try:
        int(sys.argv[1])
        os.system("sed -i 's/Port[0-9][0-9]*$/Port"+sys.argv[1]+" /' /etc/ssh/sshd_config")
    except:
        print("El primer parametro no es un puerto valido")

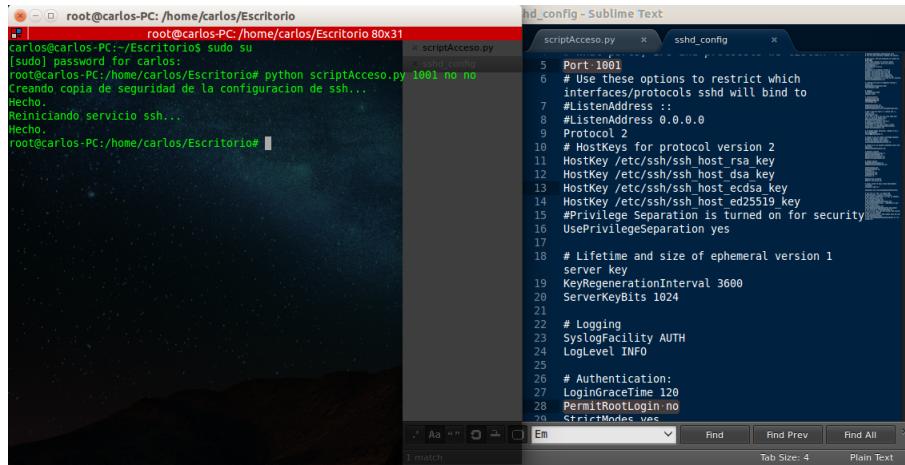
    #Hay que ir comprobando que los parametros sean correctos para no
    #destrozar el archivo
    if(sys.argv[2]=="yes" or sys.argv[2]=="no" or sys.argv[2]=="without-
        -password"):
        os.system("sed -i 's/PermitRootLogin [a-zA-Z][a-zA-Z]*-*[a-
            zA-Z]*$/PermitRootLogin "+sys.argv[2]+"/' /etc/ssh/
            sshd_config")
    else:
        print("El segundo parametro no es valido")

    if(sys.argv[3]=="yes" or sys.argv[3]=="no"):
        os.system("sed -i 's/PermitEmptyPasswords [a-zA-Z][a-zA-Z]*-*[a-
            zA-Z]*$/PermitEmptyPasswords "+sys.argv[3]+"/' /etc/ssh/
            sshd_config")
    else:
        print("El tercer parametro no es valido")

    #reiniciamos el servicio
    print("Reiniaciando servicio ssh...")
    os.system("/etc/init.d/ssh restart")
    print("Hecho.")
```

El script debe ejecutarse en modo superusuario y los argumentos pueden ser correctos. En este script vamos a cambiar el puerto de acceso, el login de root y permitir o no contraseñas vacías. Como va por bloques, se podrían introducir más, para cambiar otras cosas. El script comprueba que haya tres argumentos y se asegura de que estén en el rango de valores válidos para no destrozar el archivo, de todos modos hace una copia de

seguridad de la última configuración que funcionaba. Una muestra de la ejecución del script es la siguiente:



```
root@carlos-PC: /home/carlos/Escritorio
[...]
root@carlos-PC: /home/carlos/Escritorio# python scriptAcceso.py 1001 no no
[sudo] password for carlos:
root@carlos-PC: /home/carlos/Escritorio# python scriptAcceso.py 1001 no no
Creando copia de seguridad de la configuración de ssh...
Hecho.
Reiniciando servicio ssh...
Hecho.
root@carlos-PC: /home/carlos/Escritorio#
```

```
scriptAcceso.py      sshd_config - Sublime Text
5  Port 1001
6  # Use these options to restrict which
7  # protocols sshd will bind to
8  #ListenAddress :: 
9  #ListenAddress 0.0.0.0
10 # Protocol 2
11 HostKey /etc/ssh/ssh_host_rsa_key
12 HostKey /etc/ssh/ssh_host_dsa_key
13 HostKey /etc/ssh/ssh_host_ecdsa_key
14 HostKey /etc/ssh/ssh_host_ed25519_key
15 #Privilege Separation is turned on for security
16 UsePrivilegeSeparation yes
17
18 # Lifetime and size of ephemeral version 1
19 server key
20 KeyRegenerationInterval 3600
21 ServerKeyBits 1024
22
23 # Logging
24 SyslogFacility AUTH
25 LogLevel INFO
26
27 # Authentication:
28 LoginGraceTime 120
29 PermitRootLogin no
30 StrictModes yes
```

Figura 34: Ejecución del script de configuración de ssh.

Para usar el script con el comando find, podríamos ejecutar el siguiente comando:

```
~$:find /etc/ssh -name 'sshd_config' -exec python /home/carlos/Escritorio/
scriptAcceso.py 22 without-password no \;
```

(Fuentes: [13] [23] [25])

## 18. Escriba el script para cambiar el acceso a ssh usando PHP o Python.

Dada la versatilidad que ofrece el script de la cuestión anterior y la facilidad de modificación, concluyo que puede responder también a ésta cuestión.

## 19. Abra una consola de Powershell y pruebe a parar un programa en ejecución (p.ej), realice capturas de pantalla y comente lo que muestra.

Los procesos que se están ejecutando en la consola PowerShell de Windows pueden ser consultados mediante el comando tasklist. Para éste ejercicio abriremos la ventana de administracion de servidores (ServerManager) y pararemos el proceso con ésta herramienta. Para ver la lista de procesos que están ejecutandose ejecutamos el comando:

```
> tasklist .exe
```

Una vez que obtengamos la tabla de procesos ejecutándose, pararemos el servicio ServerManager con el comando:

```
> stop-process -processname ServerManager
```

Obtenemos los siguientes resultados:

Nombre de imagen	PID	Número de sesión	Mín. de ses.	Uso de memoria
System Idle Process	0	0	0	28 KB
System	216	Services	0	276 KB
corsrs.exe	380	Services	0	3.268 KB
lsm.exe	104	Services	1	1.012 KB
wininit.exe	360	Services	0	3.448 KB
Uninstall.exe	792	Services	0	1.120 KB
services.exe	456	Services	0	6.308 KB
lsass.exe	456	Services	0	1.048 KB
svchost.exe	532	Services	0	7.208 KB
lsass.exe	104	Services	0	1.072 KB
svchost.exe	656	Services	0	15.044 KB
lsass.exe	104	Services	0	1.072 KB
svchost.exe	724	Services	0	25.496 KB
lsass.exe	704	Services	0	1.072 KB
svchost.exe	876	Services	0	1.072 KB
lsass.exe	252	Services	0	1.072 KB
spoolsv.exe	808	Services	0	8.196 KB
lsass.exe	808	Services	0	1.072 KB
svchost.exe	1092	Services	0	2.544 KB
lsass.exe	1092	Services	0	1.072 KB
taskhost.exe	824	Console	1	5.592 KB
explorer.exe	1148	Console	0	50.892 KB
lsass.exe	1148	Console	0	1.072 KB
taskhost.exe	2772	Console	1	6.788 KB
svchost.exe	5772	Console	0	1.072 KB
ServerManager.exe	1644	Console	0	1.072 KB
taskhost.exe	1644	Console	0	1.072 KB
multitestSE.exe	1352	Services	0	6.704 KB
tasklist.exe	1352	Services	0	1.072 KB

Nombre de imagen	PID	Número de sesión	Mín. de ses.	Uso de memoria
System Idle Process	0	0	0	28 KB
System	216	Services	0	276 KB
corsrs.exe	380	Services	0	3.264 KB
lsm.exe	104	Services	1	1.120 KB
wininit.exe	360	Services	0	3.448 KB
Uninstall.exe	792	Services	0	1.120 KB
services.exe	456	Services	0	6.308 KB
lsass.exe	456	Services	0	1.072 KB
svchost.exe	532	Services	0	7.276 KB
lsass.exe	104	Services	0	1.072 KB
svchost.exe	656	Services	0	14.932 KB
lsass.exe	704	Services	0	1.072 KB
svchost.exe	724	Services	0	24.452 KB
lsass.exe	252	Services	0	1.072 KB
spoolsv.exe	876	Services	0	15.064 KB
lsass.exe	808	Services	0	1.072 KB
svchost.exe	1092	Services	0	2.544 KB
lsass.exe	1092	Services	0	1.072 KB
taskhost.exe	1096	Services	0	8.592 KB
explorer.exe	1096	Services	0	7.208 KB
lsass.exe	1096	Services	0	1.072 KB
taskhost.exe	2772	Console	1	6.788 KB
explorer.exe	944	Console	0	50.896 KB
lsass.exe	944	Console	0	1.072 KB
powershell.exe	2798	Console	1	6.494 KB
taskhost.exe	2798	Console	0	1.072 KB
tasklist.exe	1798	Console	1	5.192 KB
lsass.exe	2198	Services	0	5.752 KB

(a) Procesos en ejecución en PowerShell.

(b) Parando el proceso ServerManager y consultando si se ha parado.

Figura 35: Parando procesos en PowerShell.

(Fuentes: [31] [30])

## 20. Cuestión opcional 1: ¿Qué gestores utiliza OpenSuse?

Para gestionar paquetes en OpenSuse tenemos dos opciones: Hacerlo en modo gráfico o bien en linea de comandos. Para ello tenemos YaST y Zypper respectivamente.

(Fuentes: [21])

## 21. Cuestión opcional 2: Instale y pruebe terminator. Con screen, pruebe su funcionamiento dejando sesiones ssh abiertas en el servidor y recuperándolas posteriormente.

Hay que diferenciar terminator y screen. Terminator es una herramienta que permite abrir varias terminales en una sola ventana. Screen permite tener varias sesiones "virtuales" de manera que si se desconecta (desattach) alguna, ya sea de manera voluntaria o no, las tareas lanzadas no mueren, sino que siguen ejecutándose en segundo plano.

Muestra de terminator:



Figura 36: Probando terminator.

Abriremos una sesión ssh en cada uno, desconectaremos sesiones y veremos como algunos de los procesos lanzados siguen vivos en segundo plano y cómo se pueden recuperar. En la figura 37, podemos comprobar cómo abriendo dos procesos en clientes distintos, en el servidor se refleja ésta actividad. Lo primero que hay que hacer es abrir sesiones screen con el siguiente comando:

```
~$: screen -S [nombre_sesion]
```

Una vez que hemos abierto varias sesiones, tendremos algo así:

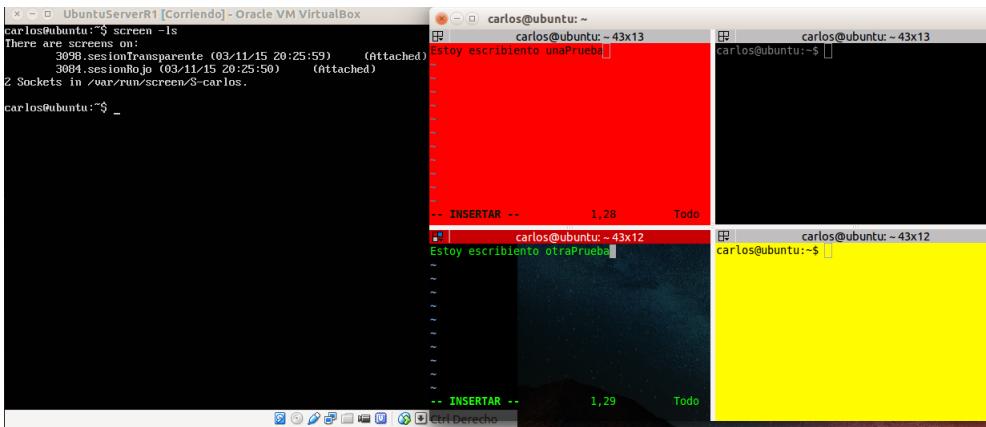


Figura 37: Abriendo sesiones screen en la máquina remota.

Aunque parezca evidente, me gustaría aclarar que en la figura, sesionRojo es abierta por la ventana de fondo rojo y la sesionTransparente es abierta por la de fondo semitransparente. Si vemos los procesos que están corriendo, veremos que sesionRojo ha lanzado un proceso vi y está escribiendo unaPrueba.txt y sesionTransparente ha lanzado otro proceso vi y está escribiendo otraPrueba.txt

```

1445 ? Ss 0:00 cron
1498 ttys1 Ss 0:00 /bin/login --
1563 ttys1 S 0:00 \_ bash
3116 ttys1 R+ 0:00 \_ ps aux
1602 ? Ss 0:00 dhclient -1 -v -pf /run/dhclient.eth0.pid -lf /var/li
1674 ? Ss 0:00 /usr/sbin/sshd -D
2465 ? Ss 0:00 \_ sshd: carlos [priv]
2514 ? S 0:00 \_ \_ sshd: carlos@pts/9
2515 pts/9 Ss+ 0:00 \_ \_ bash
2653 ? Ss 0:00 \_ sshd: carlos [priv]
2702 ? S 0:00 \_ \_ sshd: carlos@pts/1
2703 pts/1 Ss 0:00 \_ \_ bash
3097 pts/1 S+ 0:00 \_ \_ screen -S sesionTransparente
3098 ? Ss 0:00 \_ \_ SCREEN -S sesionTransparente
3099 pts/3 Ss 0:00 \_ \_ \_ /bin/bash
3112 pts/3 S+ 0:00 \_ \_ vi otraPrueba.txt
2748 ? Ss 0:00 \_ sshd: carlos [priv]
2797 ? S 0:00 \_ \_ sshd: carlos@pts/5
2798 pts/5 Ss+ 0:00 \_ \_ bash
2813 ? Ss 0:00 \_ sshd: carlos [priv]
2862 ? S 0:00 \_ \_ sshd: carlos@pts/6
2863 pts/6 Ss+ 0:00 \_ \_ bash
2956 ? Ss 0:00 \_ sshd: carlos [priv]
3005 ? S 0:00 \_ \_ sshd: carlos@pts/0
3006 pts/0 Ss 0:00 \_ \_ bash
3083 pts/0 S+ 0:00 \_ \_ screen -S sesionRojo
3084 ? Ss 0:00 \_ \_ SCREEN -S sesionRojo
3085 pts/2 Ss 0:00 \_ \_ /bin/bash
3111 pts/2 S+ 0:00 \_ \_ vi unaPrueba.txt
carlos@ubuntu:~$
```

Figura 38: Comprobando los procesos que se ejecutan en el servidor.

Ahora vamos a matar una de esas dos ventanas. A ver si la sesión sigue viva. Vemos cómo al cerrar la ventana roja, sesionRojo pasa a estado deattached (desligado), es decir, que esa sesión no está siendo ejecutada por ninguna conexión remota.

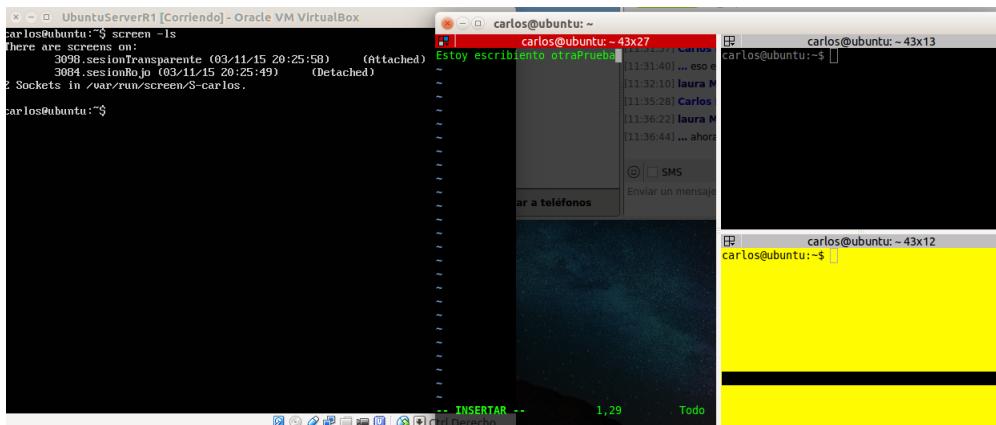


Figura 39: Comprobando el estado de las sesiones.

Si ahora miramos de nuevo los procesos, podemos ver que aunque la sesionRojo no está siendo ejecutada por nadie, sus procesos (vi - unaPrueba.txt) quedan en "stand by" (se puede comprobar en la parte inferior de la imagen):

```

1342 ttys0 Ss+ 0:00 /sbin/getty -8 38400 ttys0
1399 ? Ss 0:00 acpid -c /etc/acpi/events -s /var/run/acpid.socket
1402 ? Ss 0:00 /sbin/mdadm --monitor --pid-file /run/mdadm/monitor.pid
1444 ? Ss 0:00 atd
1445 ? Ss 0:00 cron
1498 ttys1 Ss 0:00 /bin/login --
1563 ttys1 S 0:00 \_ -bash
3125 ttys1 R+ 0:00 \_ ps aux
1602 ? Ss 0:00 dhclient -1 -v -pf /run/dhclient.eth0.pid -lf /var/lib/dhcp
1674 ? Ss 0:00 /usr/sbin/sshd -D
2465 ? Ss 0:00 \_ sshd: carlos [priv]
2514 ? S 0:00 \_ \_ sshd: carlos@pts/9
2515 pts/9 Ss+ 0:00 \_ \_ -bash
2653 ? Ss 0:00 \_ sshd: carlos [priv]
2702 ? S 0:00 \_ \_ sshd: carlos@pts/1
2703 pts/1 Ss 0:00 \_ \_ -bash
3097 pts/1 S+ 0:00 \_ \_ screen -S sesionTransparente
3098 ? Ss 0:00 \_ \_ SCREEN -S sesionTransparente
3099 pts/3 Ss 0:00 \_ \_ \_ /bin/bash
3112 pts/3 S+ 0:00 \_ \_ \_ vi otraPrueba.txt
2748 ? Ss 0:00 \_ sshd: carlos [priv]
2797 ? S 0:00 \_ \_ sshd: carlos@pts/5
2798 pts/5 Ss+ 0:00 \_ \_ -bash
2813 ? Ss 0:00 \_ sshd: carlos [priv]
2862 ? S 0:00 \_ \_ sshd: carlos@pts/6
2863 pts/6 Ss+ 0:00 \_ \_ -bash
3084 ? Ss 0:00 SCREEN -S sesionRojo
3095 pts/2 Ss 0:00 \_ /bin/bash
3111 pts/2 S+ 0:00 \_ \_ vi unaPrueba.txt
carlos@ubuntu:~$ 

```

Figura 40: Los procesos de una sesión quedan vivos en estado deattached.

Vamos a recuperar la sesionRojo en la ventana amarilla de nuestro terminator. para ello ejecutamos el comando:

```
~$: screen -d -R [ nombre_sesion ]
```

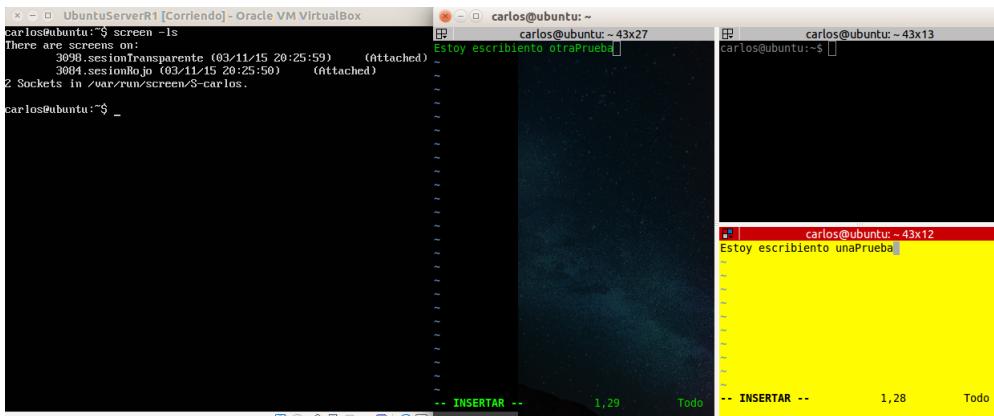


Figura 41: Recuperación de una sesión de screen que estaba en estado deattached.

Como podemos comprobar, hemos recuperado la sesión y todos los procesos que colgaban de ella y, por ende, sesionRojo vuelve a estar en estado attached. Cabe destacar que, con éste comando tambien se puede, dada una sesión en estado attached, llamarla desde una ventana, con lo que dejará de ejecutarse donde se estaba ejecutando para ejecutarse donde hemos ejecutado el comando. Con un ejemplo práctico, ahora salimos de sesionRojo en la ventana amarilla y desde ahí vamos a tomar el control de sesionTransparente. Basta con ejecutar el comando y *voilà*:

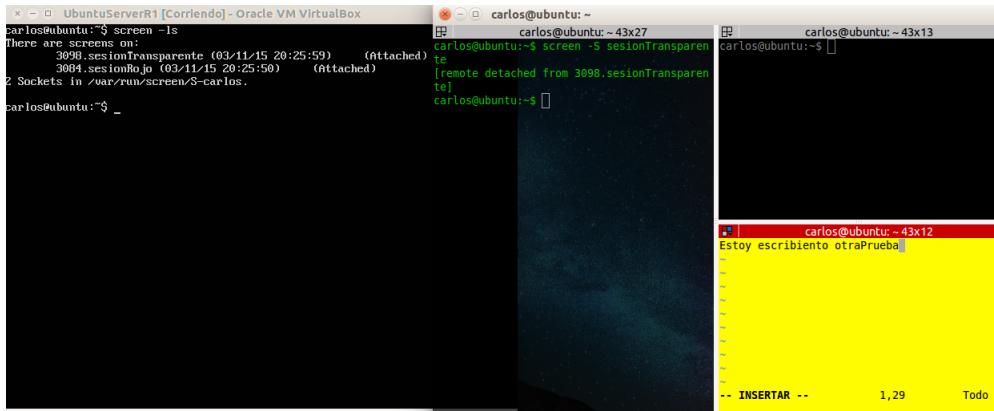


Figura 42: Recuperación de una sesión de screen que estaba en estado attached.

Como vemos, en la ventana transparente aparece un aviso de que la sesión ha sido ligada a otra ventana y por lo tanto, desligada a la suya, además aparece un identificador, siempre útil para saber quién obtuvo la sesión.

(Fuentes: [12] [17])

## 22. Cuestión opcional 3: Instale el servicio y pruebe su funcionamiento (fail2ban).

Vamos a instalar fail2ban a través de apt en Ubuntu server, para ello, lo primero es:

```
carlos@ubuntu:/etc/init.d$ sudo apt-get install fail2ban
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes:
  python-pyinotify whois
Paquetes sugeridos:
  python-gamin mailx python-pyinotify-doc
Se instalarán los siguientes paquetes NUEVOS:
  fail2ban python-pyinotify whois
0 actualizados, 3 se instalarán, 0 para eliminar y 53 no actualizados.
Necesito descargar 184 kB de archivos.
Se utilizarán 927 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] S
Des:1 http://es.archive.ubuntu.com/ubuntu/ trusty/universe fail2ban all 0.8.11-1
[129 kB]
Des:2 http://es.archive.ubuntu.com/ubuntu/ trusty/main python-pyinotify all 0.9.
4-1build1 [24.5 kB]
Des:3 http://es.archive.ubuntu.com/ubuntu/ trusty/main whois amd64 5.1.1-130.2 k
B]
Descargados 184 kB en 1seg. (118 kB/s)
Seleccionando el paquete fail2ban previamente no seleccionado.
(Leyendo la base de datos ... 70%
```

Figura 43: Instalando fail2ban en Ubuntu.

En algunas versiones era necesario editar el archivo de configuración de fail2ban, que se encuentra en /etc/fail2ban/fail2ban.conf con el objetivo de que se ejecute en segundo plano (background), poniendo la opción background a true. Si abrimos dicho archivo, nos damos cuenta que dicha opción no está:

```
GNU nano 2.2.6          Archivo: fail2ban.conf
# Fail2Ban main configuration file
#
# Comments: use '#' for comment lines and ':' (following a space) for inline comments
# Changes: in most of the cases you should not modify this
#           file, but provide customizations in fail2ban.local file, e.g.:
#
#[Definition]
# loglevel = 4
#
#[Definition]
# Option: loglevel
# Notes.: Set the log level output.
#          1 = ERROR
#          2 = WARN
#          3 = INFO
#          4 = DEBUG
# Values: [ NUM ] Default: 1
#
loglevel = 3

# Option: logtarget
^G Ver ayuda ^O Guardar ^R Leer Fich ^W RePág. ^X Cortar Tex^C Pos actual
^X Salir ^J Justificar ^U Buscar ^V Pág. Sig. ^U PegarTxt ^T Ortografía
```

Figura 44: Ausencia de la opción background en fail2ban.conf.

Si aplicamos lo aprendido en la cuestión 9, podemos listar los servicios en ejecución con:

```
~$: ls /etc/init.d
```

Al que le aplicaremos el comando grep f, que buscará todos los servicios que contengan la letra f (de fail2ban):

```
carlos@ubuntu:~$ ls /etc/init.d |grep f
fail2ban
friendly-recovery
resolvconf
umountfs
umountnfs.sh
carlos@ubuntu:~$ _
```

Figura 45: Comprobación de que fail2ban está ejecutándose como servicio.

Ahora configuraremos el archivo /etc/fail2ban/jail.conf que contiene la configuración de los baneos. Cuanto tiempo se banea una determinada ip, a través de qué protocolo, qué numero de intentos son necesarios para banear... Nosotros activaremos el baneo por fallo a través de ssh para probar la funcionalidad de la herramienta, el bloque correspondiente debe quedar así:

```
[ssh]
enabled  = true
port      = ssh
filter    = sshd
logpath   = /var/log/auth.log
maxretry = 3
```

Figura 46: Activando el baneo por 3 fallos a través de ssh.

Dado que mi máquina anfitriona tiene hecho todo el proceso de automatizar el login con la clave pública en el servidor, lo que haré para probar será ingresar por ssh a la máquina CentOS desde mi máquina anfitriona y, una vez establecida la conexión, ingresar por ssh a Ubuntu server desde la máquina CentOS remotamente. Un poco lioso pero tambien sirve para probar las conexiones ssh anidadas. Introduciré la contraseña incorrectamente 3 veces para copar el máximo de intentos y ver si la ip ha sido incluida en algún tipo de log. Como podemos observar, el servidor rechaza la conexión al tercer intento.

```
[carlos@localhost ~]$ ssh carlos@192.168.1.136
The authenticity of host '192.168.1.136 (192.168.1.136)' can't be established.
ECDSA key fingerprint is 36:cd:98:44:d2:2c:f7:cd:0d:69:92:a4:d6:0d:ca:54.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.1.136' (ECDSA) to the list of known hosts.
carlos@192.168.1.136's password:
Permission denied, please try again.
carlos@192.168.1.136's password:
Permission denied, please try again.
carlos@192.168.1.136's password:
Permission denied (publickey,password).
[carlos@localhost ~]$
```

Figura 47: Rechazo de la conexión al tercer intento fallido.

Si ahora consultamos el archivo `/var/log/auth.log`, que es donde hemos dicho que queremos que nos escriba los logs de fail2ban, buscamos la fecha en la que se han tenido los accesos y obtenemos líneas como éstas:

```
Nov 3 11:47:31 ubuntu sshd[12480]: pam_unix(sshd:auth): authentication failure: user carlos
Nov 3 11:47:37 ubuntu sshd[12480]: Failed password for carlos from 192.168.1.136 port 54442 ssh2
Nov 3 11:47:44 ubuntu sshd[12480]: password rejected 3 times; user failed password
```

Figura 48: Línea de fallo generada por fail2ban.

Se supone que la búsqueda se puede automatizar con la herramienta fail2ban-regex, pasando el archivo de logs y el archivo de configuraciones de las conexiones que quieras consultar pero yo no he sido capaz de conseguirlo. Da errores por cualquier cosa:

```
carlos@ubuntu:~$ fail2ban-regex /var/log/auth.log /etc/fail2ban/filter.d/ssh.conf
f
=====
Running tests
=====
Use failregex line : /etc/fail2ban/filter.d/ssh.conf
Traceback (most recent call last):
  File "/usr/bin/fail2ban-regex", line 415, in <module>
    fail2banRegex.readRegexCmd_regex('fail') or sys.exit(-1)
  File "/usr/bin/fail2ban-regex", line 226, in readRegex
    'add:shRegex' % regetype.title())(regex.getFailRegex())
  File "/usr/share/fail2ban/server/filter.py", line 92, in addFailRegex
    raise e
server.failregex.RegexException: No 'host' group in '/etc/fail2ban/filter.d/ssh.conf'
carlos@ubuntu:~$ _
```

Figura 49: Error de la herramienta fail2ban-regex.

(Fuentes: [10] [11])

## Referencias

- [1] CCM. Conectarse de manera remota por ssh linux. <http://es.ccm.net/faq/3561-conectarse-de-manera-remota-por-ssh-linux>.
- [2] Study CCNA. Telnet & ssh. <http://study-ccna.com/telnet-ssh>.
- [3] danjared wordpress. Configurar el proxy en ubuntu. <https://dunjared.wordpress.com/2011/03/09/configurar-el-proxy-en-ubuntu>.
- [4] Documentacion de Fedora. Usando yum con un servidor proxy. [https://docs.fedoraproject.org/es-ES/Fedora\\_Core/4/html/Software\\_Management\\_Guide/sn-yum-proxy-server.html](https://docs.fedoraproject.org/es-ES/Fedora_Core/4/html/Software_Management_Guide/sn-yum-proxy-server.html).
- [5] Páginas de manual. manual de yum. [http://www.linuxcommand.org/man\\_pages/yum8.html](http://www.linuxcommand.org/man_pages/yum8.html).
- [6] Blog desde linux. Configura conexiones ssh sin password en solo 3 pasos. <http://blog.desdelinux.net/ssh-sin-password-solo-3-pasos/>.
- [7] DirectAdmin. Demo de directadmin. <http://www.directadmin.com/demo.html>.
- [8] e ticpc. Como instalar lamp en un servidor ubuntu. <http://www.eticpc.es/e-tic-blog/9-software/30-como-instalar-lamp-en-un-servidor-ubuntu.html>.
- [9] Red Hat enterprise linux. Protocolo ssh. <http://www.gb.nrao.edu/pubcomputing/redhatELWS4/RH-DOCS/rhel-rg-es-4/ch-ssh.html>.
- [10] Fail2ban. Howto fail2ban spanish. [http://www.fail2ban.org/wiki/index.php/HOWTO\\_fail2ban\\_spanish](http://www.fail2ban.org/wiki/index.php/HOWTO_fail2ban_spanish).
- [11] Fail2ban. Manual 0 8. [http://www.fail2ban.org/wiki/index.php/MANUAL\\_0\\_8](http://www.fail2ban.org/wiki/index.php/MANUAL_0_8).
- [12] Matt Cutts: Gadgets Google and SEO. A quick tutorial on screen. <https://www.mattcutts.com/blog/a-quick-tutorial-on-screen/>.
- [13] HolaMundo. Como acceder a los argumentos de python. <http://www.holamundo.es/lenguaje/python/articulos/acceder-argumentos-pasados-parametro-python.html>.
- [14] ISPConfig. Demo de ispconfig. [http://www.ispconfig.org/page/en/ispconfig\\_online-demo.html](http://www.ispconfig.org/page/en/ispconfig_online-demo.html).
- [15] JGA ITpro. Windows server 2012 - instalar iis 8. <https://www.youtube.com/watch?v=zEr10VzvTpE>.
- [16] Ubuntu Life. Tip: Ssh x11 forwarding. <https://ubuntulife.wordpress.com/2008/09/23/tip-ssh-x11-forwarding/>.

- [17] Arch Linux. Gnu screen. [https://wiki.archlinux.org/index.php/GNU\\_Screen](https://wiki.archlinux.org/index.php/GNU_Screen).
- [18] Linux man page. ssh-keygen(1). <http://linux.die.net/man/1/ssh-keygen>.
- [19] Digital Ocean. How to install and secure phpmyadmin on ubuntu 14.04. <https://www.digitalocean.com/community/tutorials/how-to-install-and-secure-phpmyadmin-on-ubuntu-14-04>.
- [20] Digital Ocean. How to install webmin with ssl on ubuntu 14.04. <https://www.digitalocean.com/community/tutorials/how-to-install-webmin-with-ssl-on-ubuntu-14-04/>.
- [21] OpenSuse.org. Gestión de paquetes. [https://es.opensuse.org/Gesti%C3%B3n\\_de\\_paquetes#Gestor\\_de\\_paquetes](https://es.opensuse.org/Gesti%C3%B3n_de_paquetes#Gestor_de_paquetes).
- [22] Stack Overflow. How to check service is running or not in ubuntu? <http://stackoverflow.com/questions/18721149/how-to-check-service-is-running-or-not-in-ubuntu>.
- [23] Persoal. Expresiones regulares. [http://persoal.citius.usc.es/tf.pena/ASR/Tema\\_2html/node22.html](http://persoal.citius.usc.es/tf.pena/ASR/Tema_2html/node22.html).
- [24] Red Hat Customer Portal. Adding, enabling, and disabling a yum repository. [https://access.redhat.com/documentation/en-US/Red\\_Hat\\_Enterprise\\_Linux/6/html/Deployment\\_Guide/sec-Managing\\_Yum.Repositories.html](https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Deployment_Guide/sec-Managing_Yum.Repositories.html).
- [25] Python.org. Tutorial de python: 9. errores y excepciones. <http://docs.python.org.ar/tutorial/3/errors.html>.
- [26] rm rf.es. Arrancar / parar / reiniciar servicios en rhel 7 y centos 7. <http://rm-rf.es/arrancar-parar-reiniciar-servicios-en-rhel-7-y-centos-7/>.
- [27] rm rf.es. Guía de comandos apt para debian / ubuntu (apt-get, apt-cache). <http://rm-rf.es/guia-de-comandos-apt-para-debian-ubuntu-apt-get-apt-cache>.
- [28] The Geek Stuff. 7 patch command examples to apply diff patch files in linux. <http://www.thegeekstuff.com/2014/12/patch-command-examples/>.
- [29] TDLP. Configure the /etc/ssh/sshd\_config file. <http://www.tldp.org/LDP/solrhe/Securing-Optimizing-Linux-RH-Edition-v1.3/chap15sec122.html>.
- [30] Microsoft Technet. Tasklist. <https://technet.microsoft.com/en-us/library/cc730909.aspx>.
- [31] Microsoft Technet. Using the stop-process cmdlet. <https://technet.microsoft.com/en-us/library/ee177004.aspx>.
- [32] Guia Ubuntu. Añadir repositorios externos. [http://www.guia-ubuntu.com/index.php/A%C3%B1adir\\_repositorios\\_externos](http://www.guia-ubuntu.com/index.php/A%C3%B1adir_repositorios_externos).

- [33] Ubuntu-es. Cómo se detienen o arrancan los servicios ahora? <http://www.ubuntu-es.org/node/134485#.Vjh0opdVK1E>.
- [34] Udacity. A step by step guide to install lamp (linux, apache, mysql, python) on ubuntu. <http://blog.udacity.com/2015/03/step-by-step-guide-install-lamp-linux-apache-mysql-python-ubuntu.html>.
- [35] Biolucas web & mobile. Cómo aumentar la capacidad de phpmyadmin importando archivos. <http://biolucas.com/como-aumentar-la-capacidad-de-phpmyadmin-importando-archivos/>.
- [36] Wikipedia. Cherokee. [https://es.wikipedia.org/wiki/Cherokee\\_%28servidor\\_web%29](https://es.wikipedia.org/wiki/Cherokee_%28servidor_web%29).
- [37] Wikipedia. Lighttpd. <https://es.wikipedia.org/wiki/Lighttpd>.
- [38] Wikipedia. Servidor web. [https://es.wikipedia.org/wiki/Servidor\\_web#Software](https://es.wikipedia.org/wiki/Servidor_web#Software).
- [39] Wikipedia. sources.list. [https://es.wikipedia.org/wiki/Sources.list#sources.list\\_en\\_Ubuntu\\_.28y\\_derivadas.29](https://es.wikipedia.org/wiki/Sources.list#sources.list_en_Ubuntu_.28y_derivadas.29).
- [40] Wikipedia. Thttpd. <https://es.wikipedia.org/wiki/Thhttpd>.
- [41] Wikipedia. Tomcat. <https://es.wikipedia.org/wiki/Tomcat>.
- [42] Xmodulo. How to install lamp stack (apache, mariadb/mysql and php) on centos. <http://xmodulo.com/install-lamp-stack-centos.html>.