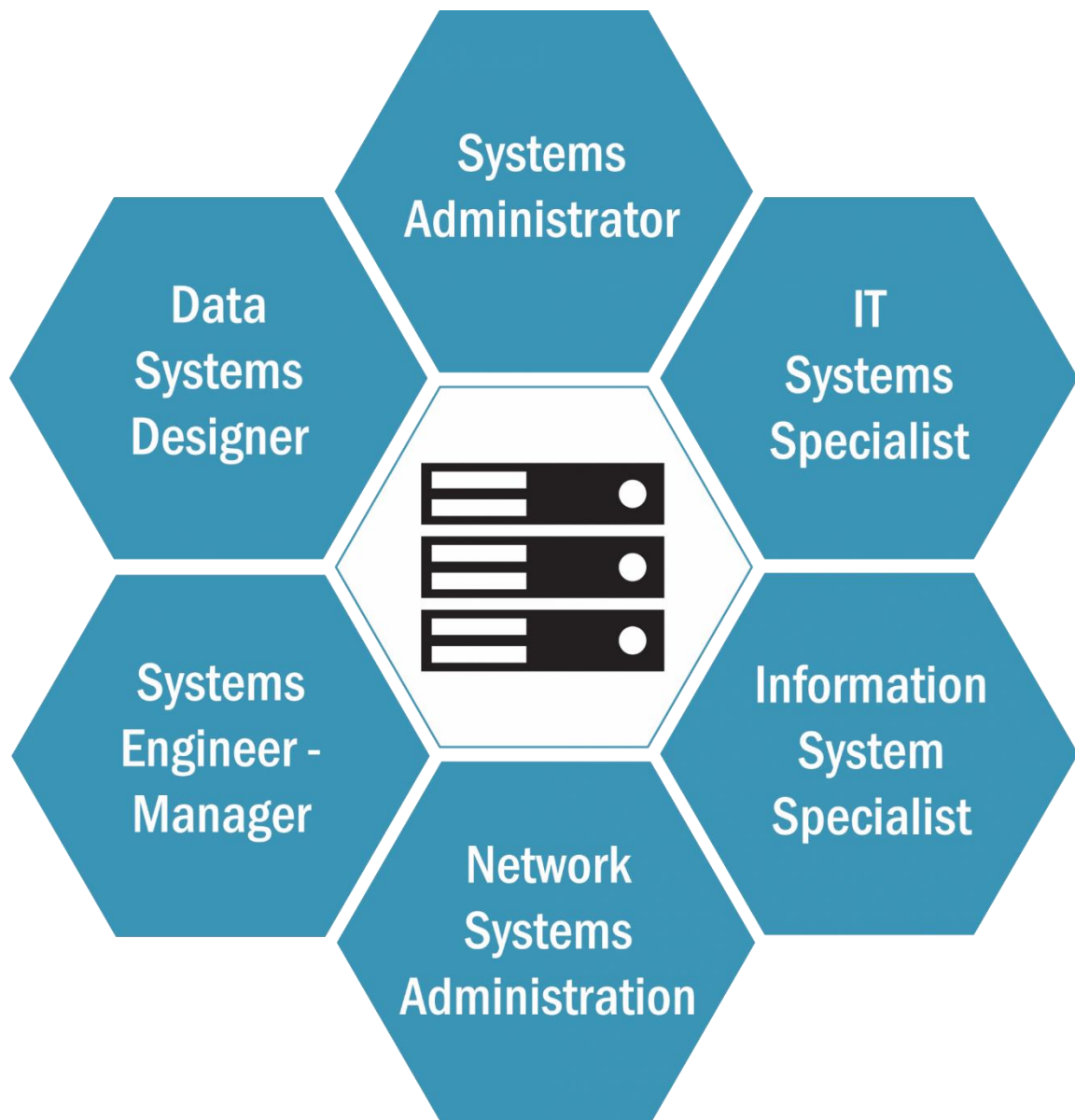


Servidor Startup - PIXIE



Héctor Chamorro Álvarez – XXXXXXXXX

M.A.M – XXXXXXXXX

Tabla de contenidos

Introducción.....	5
Descripción básica del Sistema	7
Configuración inicial del sistema	7
Grupos de trabajo.....	7
Reporte del login del administrador.....	8
Copias de seguridad.....	8
Copias de seguridad remotas	9
Tripwire.....	14
Ficheros CRON.....	19
Particiones y cuotas	20
Sistema de ficheros /mnt/home.....	20
Aplicar cuotas	20
Configurar CGI	22
Ficheros log	22
Descripción básica de los servidores	23
SSH.....	23
Apache2	24
Introducción	24
Instalación Apache2.....	24
Habilitar fichero .htaccess.....	27
Mover HTML.....	28
Protección anti-ataques DoS	29
Página segura HTTPS con SSL-RSA	30
AWStats.....	34
MariaDB	39
Creación de Blogs	44
Cacti	45
Servidor de correo electrónico (Roundcube)	46
Servidor FTP	48
Explicación scripts CGI	50
Problemas encontrados.....	51
Posibles mejoras.....	52
Conclusiones	53
Referencias	54

Tabla de figuras

Ilustración 1: Startup Imagen 1	5
Ilustración 2: Startup Imagen 2	5
Ilustración 3: Startup Imagen 3	6
Ilustración 4: Startup Imagen 4	6
Ilustración 5: ID grupos y usuarios	7
Ilustración 6: Backups	9
Ilustración 7: Tamaño backups	9
Ilustración 8: Crear app Dropbox	10
Ilustración 9: Configuración app Dropbox	10
Ilustración 10: Ejecutando backup remoto	11
Ilustración 11: link verificación Dropbox	11
Ilustración 12: Código verificación Dropbox	12
Ilustración 13: Prueba dropbox remoto 1	12
Ilustración 14: Prueba dropbox remoto 2	12
Ilustración 15: Prueba dropbox remoto 3	13
Ilustración 16: tripwire installer	14
Ilustración 17: tripwire dominio	14
Ilustración 18: Instalación tripwire 1	15
Ilustración 19: Instalación tripwire 2	16
Ilustración 20: Instalación tripwire 3	16
Ilustración 21: Instalación tripwire 4	17
Ilustración 22: Prueba tripwire report	18
Ilustración 23: Ficherto cron	19
Ilustración 24: Ejemplo repquota	21
Ilustración 25: Módulos Apache2	24
Ilustración 26: Página por defecto Apache2	25
Ilustración 27: Orden ip addr	25
Ilustración 28: Modificación de fichero hosts	26
Ilustración 29: configuración apache2.conf	27
Ilustración 30: Home Page	28
Ilustración 31: Creación clave server.key	30
Ilustración 32: Creación de certificado	31
Ilustración 33: Configuración de ficheros SSL	32
Ilustración 34: Conexión HTTPS	32
Ilustración 35: Certificado HTTPS	33
Ilustración 36: Prueba conexión cifrada	33
Ilustración 37: Ejemplo AWStats	34
Ilustración 38: Instalación AWStats	34
Ilustración 39: Configuración Awstats	36
Ilustración 40: AWStats.conf	37
Ilustración 41: Prueba-1 AWStats	38
Ilustración 42: Prueba-2 AWStats	38
Ilustración 43: Prueba-3 AWStats	38
Ilustración 44: Instalación MariaDB	39
Ilustración 45: Instalación PhpMyAdmin	40
Ilustración 46: Configuración phpmyadmin 1	40

Ilustración 47: Configuración phpmyadmin 2.....	40
Ilustración 48: Login PhpMyAdmin	41
Ilustración 49: Crear Base de datos users.....	41
Ilustración 50: Crear tablas	42
Ilustración 51: wordpress bienvenida	44
Ilustración 52: wordpress config	45
Ilustración 53: Ejemplo cacti	46
Ilustración 54: Configuración roundcube	46
Ilustración 55: Login de roundcube	47
Ilustración 56: proftpd.conf	48
Ilustración 57:login proftpd	49
Ilustración 58: proftps escritorio usuario	49

Introducción

La siguiente práctica consistirá en hacer un servidor web para una pequeña startup, como se indica en el enunciado que podremos encontrar en Studium. Nuestro enfoque ha sido hacia una tienda de E-commerce y en concreto de venta de ropa.

Cabe destacar que la plantilla html la hemos descargado ya que era gratuita y la hemos modificado con nuestras necesidades.

Veamos algunas imágenes del sitio web que hemos creado:

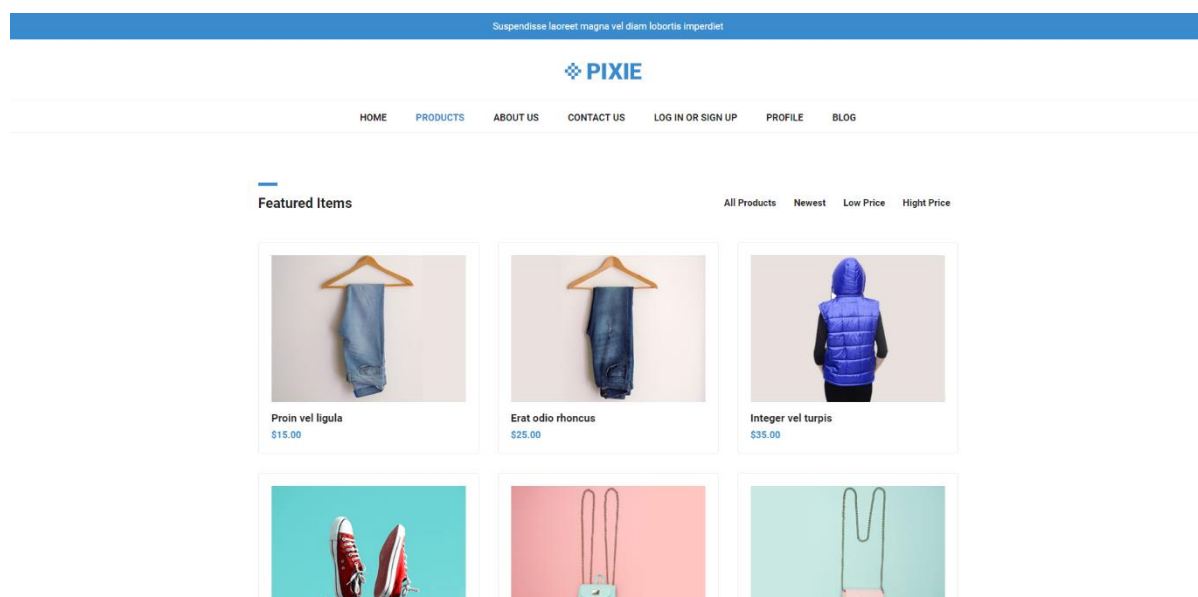


Ilustración 1: Startup Imagen 1

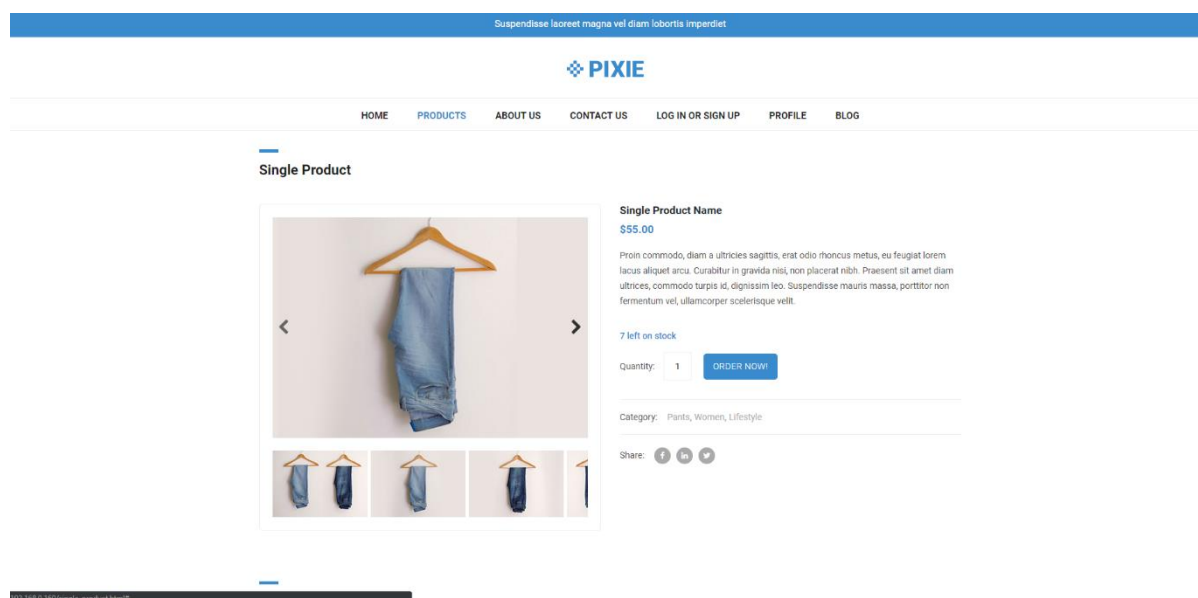


Ilustración 2: Startup Imagen 2

Descripción básica del Sistema

Lo primero que tenemos que conocer, antes de entrar en detalle de la descripción básica del sistema, es conocer el sistema operativo donde vamos a realizar la práctica.

Para la realización de la práctica se han usado tanto las distribuciones debian instaladas en el laboratorio de informática como Ubuntu para poder utilizarlo desde casa. En nuestro caso hemos usado *Ubuntu 18.04.2*.

Configuración inicial del sistema

Primero debemos actualizar el sistema e instalar algunos módulos que necesitaremos más adelante. Para actualizar el sistema:

1. `apt-get update`
2. `apt-get upgrade`

Después debemos instalar el siguiente paquete:

1. `apt-get install build-essential`

Grupos de trabajo

Lo primero que realizaremos para la configuración del sistema es la creación de los dos grupos de usuarios requeridos (CLIENTES y TECNICOS)

Para añadirlos lo hacemos con la siguiente orden:

1. `groupadd CLIENTES`
2. `groupadd TECNICOS`

Si queremos ver el id de los usuarios que se encuentran en nuestro sistema podemos usar la siguiente orden:

1. `cat /etc/group | awk -F ":" '{ print $1,$3 }'`

La salida que obtenemos es la siguiente:

```
geoclue 124
gdm 125
hector 1000
smbshare 126
mysql 127
quotasuser 1002
dovecot 130
dovenull 131
courier 132
postfix 128
postdrop 129
TECNICOS 1004
CLIENTES 1005
Debian-snmp 133
pruebasborrar 1001
root@hector-HUAWEI-MateBook-D:/home/hector#
```

Ilustración 5: ID grupos y usuarios

También debemos crear una carpeta para el grupo de técnicos en la cual solo puedan acceder ellos por lo que la crearemos en el directorio `/mnt/home/manuales`. Cabe destacar que `/mnt/home` en el momento de esta explicación no está creado ya que será la partición donde guardaremos los usuarios, sin embargo, en [esta](#) sección podemos ver como instalarla.

Para la creación de esta carpeta realizamos los siguientes comandos:

```
1. mkdir /mnt/home/manuales
2. chown root:1004 /mnt/home/manuales
```

Donde pone 1004 se debe poner el número que nos devuelve el comando que hemos explicado anteriormente. El comando `chown` sirve para especificar el o los propietarios de un archivo o carpeta.

Reporte del login del administrador

Para la realización de este apartado haremos uso de la carpeta `/etc/profile.d/` esta carpeta contiene con una serie de scripts que se ejecutan al iniciar sesión. Encontramos un problema ya que al guardar en esta carpeta nuestro código de perl para enviar un correo al administrador, no se ejecutaba el programa por lo que creamos un fichero `.sh` que ejecutara el `.pl`. El script `.sh` es el siguiente:

```
1. #!/usr/bin/bash
2. `perl /etc/profile.d/report_root_login.pl`;
```

El script `.pl` es el siguiente:

```
1. #!/usr/bin/perl
2. use strict;
3. use warnings;
4.
5. use Mail::Sender;
6. use Email::Send::SMTP::Gmail;
7.
8.
9. my $destination='xamo1998@gmail.com';
10. my ($mail,$error)=Email::Send::SMTP::Gmail->new( -smtp=>'smtp.gmail.com',
11.                                                    -login=>'xamo1998@gmail.com',
12.                                                    -pass=>'XXXXXXXXXXXXXXXXXXXX',
13.                                                    -layer=>'ssl');
14. print "Session error: $error" unless ($mail!=1);
15. $mail->send(-to=>$destination,-subject=>'Admin Login', -
    body=>'The admin just logged in the system! =) ', -attachments=>'');
16. $mail->bye;
```

Para el correcto funcionamiento del código anterior hemos tenido que instalar a través de CPAN los módulos `Mail::Sender` y `Email::Send::SMTP::Gmail`, para ello tan solo ejecutamos el comando `cpan` y una vez dentro `install Mail::Sender` y `Email::Send::SMTP::Gmail`.

Copias de seguridad

En este apartado vamos a explicar el método que hemos usado para la realización de las copias de seguridad.

Dicho esto, vamos a explicar como lo hemos hecho, hemos utilizado `rsync` que es una aplicación libre para sistemas de tipo Unix y Microsoft Windows que ofrece transmisión eficiente de datos incrementales, que opera también con datos comprimidos y cifrados.

Primero mostraremos el fichero que realiza las copias de seguridad:


```

1. #!/usr/bin/perl
2. use warnings;
3. use strict;
4.
5. system('rsync -av /mnt /var/backups/backup_mnt.rsync');
6. system('rsync -av /etc /var/backups/backup_etc.rsync');
7. system('rsync -av /home /var/backups/backup_home.rsync');
8. system('rsync -av /usr/local/sbin /var/backups/backup_usr.rsync');
9. my $perm=0644;
10. chmod($perm, "/var/backups/backup_mnt.rsync");
11. chmod($perm, "/var/backups/backup_etc.rsync");
12. chmod($perm, "/var/backups/backup_home.rsync");
13. chmod($perm, "/var/backups/backup_usr.rsync");

```

Este archivo se encuentra en la carpeta `/usr/local/sbin/backups.pl` y como vemos nos genera una copia de seguridad para las carpetas que nos interesa conservar si se pierden:

- **/mnt/home:** Aquí se encuentran todos los usuarios que se registran en el sistema.
- **/etc:** Archivos de configuración para el correcto funcionamiento del servidor
- **/home:** Para guardar las carpetas del usuario principal y de otros tipos de usuarios que se encuentren en este directorio.
- **/usr/local/sbin:** En esta carpeta guardamos los scripts que se ejecutan con el cron.

Si ejecutamos el archivo podemos ver las copias de seguridad que nos ha generado:

```

drw-r--r--  3 root root      4096 may 24 14:55 backup_etc.rsync
drw-r--r--  3 root root      4096 may 24 14:55 backup_home.rsync
drw-r--r--  3 root root      4096 may 24 14:55 backup_mnt.rsync
drw-r--r--  3 root root      4096 may 24 14:55 backup_usr.rsync

```

Ilustración 6: Backups

Si vemos el tamaño de estos ficheros con el comando `du` vemos que el más grande es el home ya que tenemos muchas cosas descargadas:

```

root@hector-HUAWEI-MateBook-D:/var/backups# du -m -s -h *.rsync
16M      backup_etc.rsync
1,4G     backup_home.rsync
24K      backup_mnt.rsync
44K      backup_usr.rsync

```

Ilustración 7: Tamaño backups

Tras haber hecho esto, vamos a ver como hemos hecho las copias remotas, las cuales enviamos por Dropbox utilizando el siguiente código.

Copias de seguridad remotas

Hemos escrito un script el cual permite al administrador mandar sus copias de seguridad a la nube, en este caso a Dropbox, para ello, debemos implementar una serie de cosas:

Primero entramos en Dropbox developers, iniciamos sesión y creamos una aplicación:



Ilustración 8: Crear app Dropbox

Después nos pide 3 datos:

Ilustración 9: Configuración app Dropbox

El código es el siguiente:

```

1. use Webservice::Dropbox;
2.
3. my $dropbox = Webservice::Dropbox->new({
4.     key => '1mrpx4in244hy2h', # App Key
5.     secret => 'cm20j76v7xfwnyf' # App Secret
6. });
7. # Authorization
8. if ($access_token) {
9.     $dropbox->access_token($access_token);
10. } else {
11.     my $url = $dropbox->authorize;
12.
13.     print "Please Access URL and press Enter: $url\n";
14.     print "Please Input Code: ";
15.
16.     chomp( my $code = <STDIN> );
17.
18.     unless ($dropbox->token($code)) {
19.         die $dropbox->error;
20.     }
21.
22.     print "Successfully authorized.\nYour AccessToken: ", $dropbox->access_token, "\n";
23. }
24.
25. my $info = $dropbox->get_current_account or die $dropbox->error;
26. my $to_compress="/var/backups/backup_mnt.rsnc /var/backups/backup_usr.rsnc /var/backups/backup_etc.rsnc";

```

```

27. my $compressed="/var/backups/backup_mnt.zip";
28. system("zip -r $compressed $to_compress");
29. # upload
30. # https://www.dropbox.com/developers/documentation/http/documentation#files-
    upload
31. my $fh_upload = IO::File->new("/var/backups/backup_mnt.zip");
32. $dropbox->upload('/make_test_folder/backup.zip', $fh_upload) or die $dropbox-
    >error;
33. $fh_upload->close;
34. unlink $compressed;

```

Veamos un ejemplo, todos los `.rsync` están dentro de `/var/backups/` por lo que los comprimiremos en un `.zip` y los enviaremos a la cuenta del administrador.

Al ejecutar nos proporciona un link el cual al ponerlo en el navegador nos da un código para poder subir los archivos:

```

root@hector-HUAWEI-MateBook-D: /var/backups
root@hector-HUAWEI-MateBook-D:/var/backups# pico /usr/local/sbin/rem.pl
root@hector-HUAWEI-MateBook-D:/var/backups# perl /usr/local/sbin/rem.pl
Please Access URL and press Enter: https://www.dropbox.com/oauth2/authorize?client_id=1mrpx4in244hy2h&response_type=code
Please Input Code: █

```

Ilustración 10: Ejecutando backup remoto

Si introducimos el link en el navegador nos aparecerá una ventana como esta:

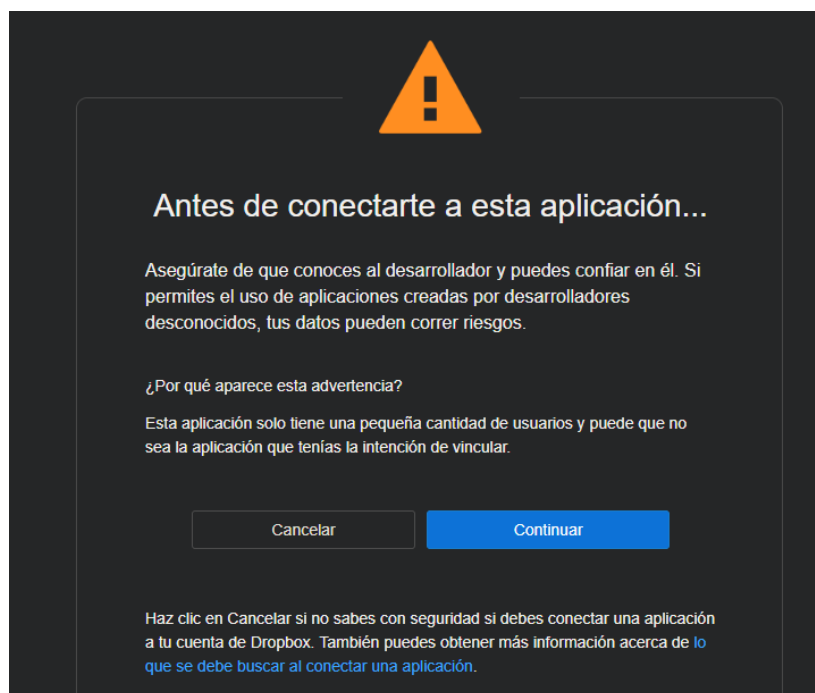


Ilustración 11: link verificación Dropbox

Le damos a continuar y nos mostrará un código de verificación el cual debemos copiar en el script que estábamos ejecutando anteriormente:

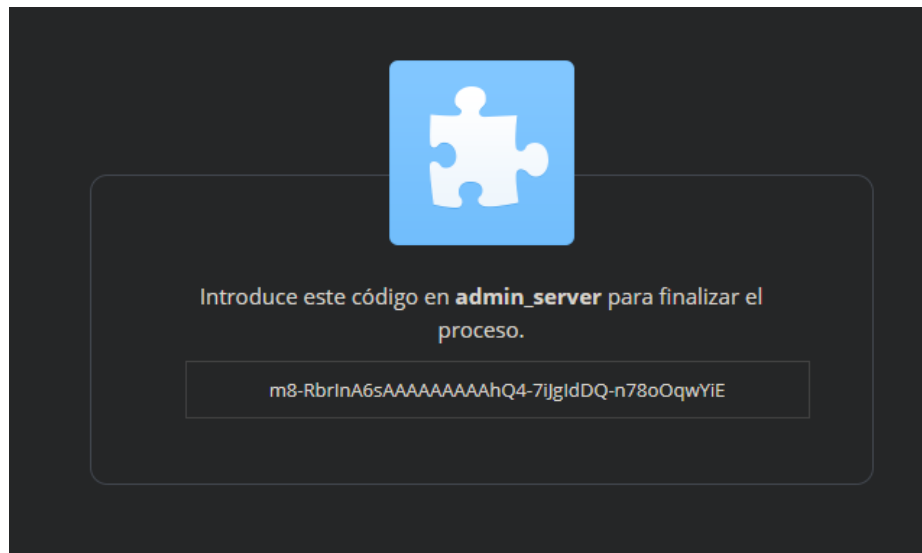


Ilustración 12: Código verificación Dropbox

Al copiar el código y dar enter se creará el fichero zip y al acabar se enviará a Dropbox como podemos ver en las siguientes imágenes:

```
adding: var/backups/backup_etc.rsync/etc/rcS.d/S01ufw (deflated 68%)
adding: var/backups/backup_etc.rsync/etc/rcS.d/S01pppd-dns (deflated 46%)
adding: var/backups/backup_etc.rsync/etc/rcS.d/S01udev (deflated 63%)
adding: var/backups/backup_etc.rsync/etc/rcS.d/S01apparmor (deflated 60%)
adding: var/backups/backup_etc.rsync/etc/rcS.d/S01alsa-utils (deflated 65%)
adding: var/backups/backup_etc.rsync/etc/rcS.d/S01networking (deflated 67%)
adding: var/backups/backup_etc.rsync/etc/rcS.d/S01kmod (deflated 58%)
adding: var/backups/backup_etc.rsync/etc/netplan/ (stored 0%)
adding: var/backups/backup_etc.rsync/etc/netplan/01-network-manager-all.yaml (
adding: var/backups/backup_etc.rsync/etc/services (deflated 61%)
adding: var/backups/backup_etc.rsync/etc/cron.hourly/ (stored 0%)
adding: var/backups/backup_etc.rsync/etc/cron.hourly/.placeholder (deflated 10
adding: var/backups/backup_etc.rsync/etc/init/ (stored 0%)
adding: var/backups/backup_etc.rsync/etc/init/whoopsie.conf (deflated 37%)
adding: var/backups/backup_etc.rsync/etc/init/anacron.conf (deflated 36%)
adding: var/backups/backup_etc.rsync/etc/magic.mime (deflated 15%)
adding: var/backups/backup_etc.rsync/etc/ftpusers (deflated 15%)
adding: var/backups/backup_etc.rsync/etc/courier/ (stored 0%)
adding: var/backups/backup_etc.rsync/etc/courier/authdaemonrc (deflated 53%)
adding: var/backups/backup_etc.rsync/etc/courier/dhparams.pem (deflated 35%)
adding: var/backups/backup_etc.rsync/etc/vtrgb (deflated 54%)
adding: var/backups/backup_etc.rsync/etc/subuid- (deflated 22%)
root@hector-HUAWEI-MateBook-D:/var/backups#
```

Ilustración 13: Prueba dropbox remoto 1

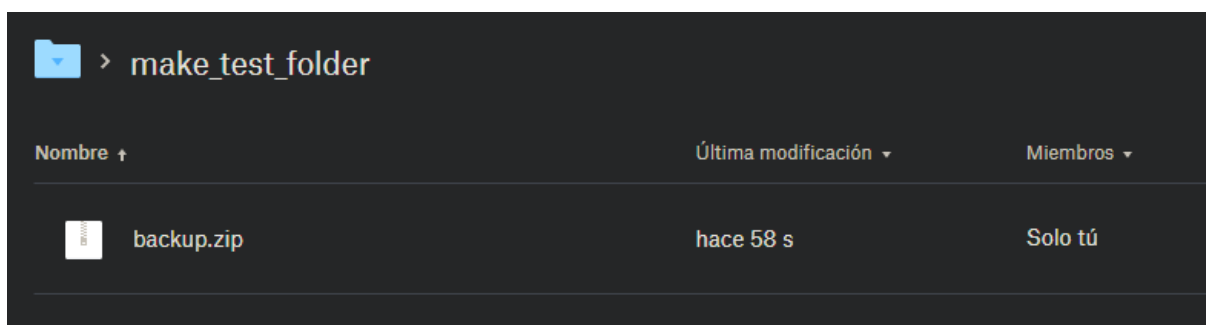


Ilustración 14: Prueba dropbox remoto 2

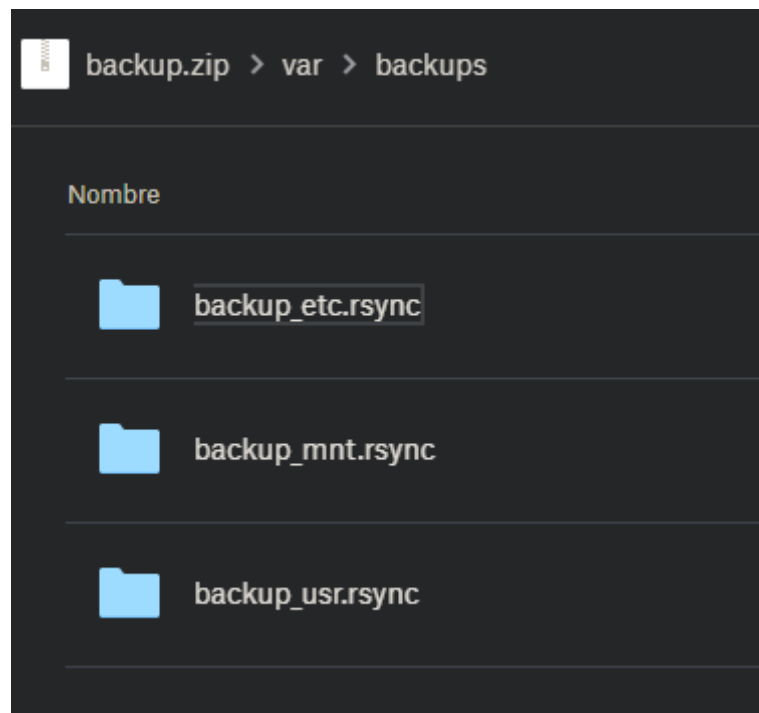


Ilustración 15: Prueba dropbox remoto 3

Otra opción para las copias de seguridad remotas es mediante rsync y ssh, dejamos el código para realizarlo con este método:

```
1. #!/usr/bin/perl
2.
3. use warnings;
4. use strict;
5.
6. system('rsync -av /mnt/home backup_mnt.rsync');
7. system('rsync -av /etc backup_etc.rsync');
8. system('rsync -av /var backup_var.rsync');
9. system('rsync -av /usr/local/sbin backup_usr.rsync');
10.
11. system('rsync -avz -P -
    e ssh /mnt/home/backup_mnt.rsync root@172.20.1.58:/home/backups');
12. system('rsync -avz -P -
    e ssh /etc/backup_etc.rsync root@172.20.1.58:/home/backups');
13. system('rsync -avz -P -
    e ssh /var/backup_var.rsync root@172.20.1.58:/home/backups');
14. system('rsync -avz -P -
    e ssh /usr/local/sbin/backup_usr.rsync root@172.20.1.58:/home/backups');
```

Tripwire

Para la monitorización local hemos usado Tripwire el cual, al ejecutarlo nos proporciona una gran cantidad de información muy útil para el administrador. Para instalar tripwire debemos realizar los siguientes pasos:

Instalamos tripwire con la siguiente orden:

1. `apt-get install tripwire`

lo configuramos como sitio de internet:

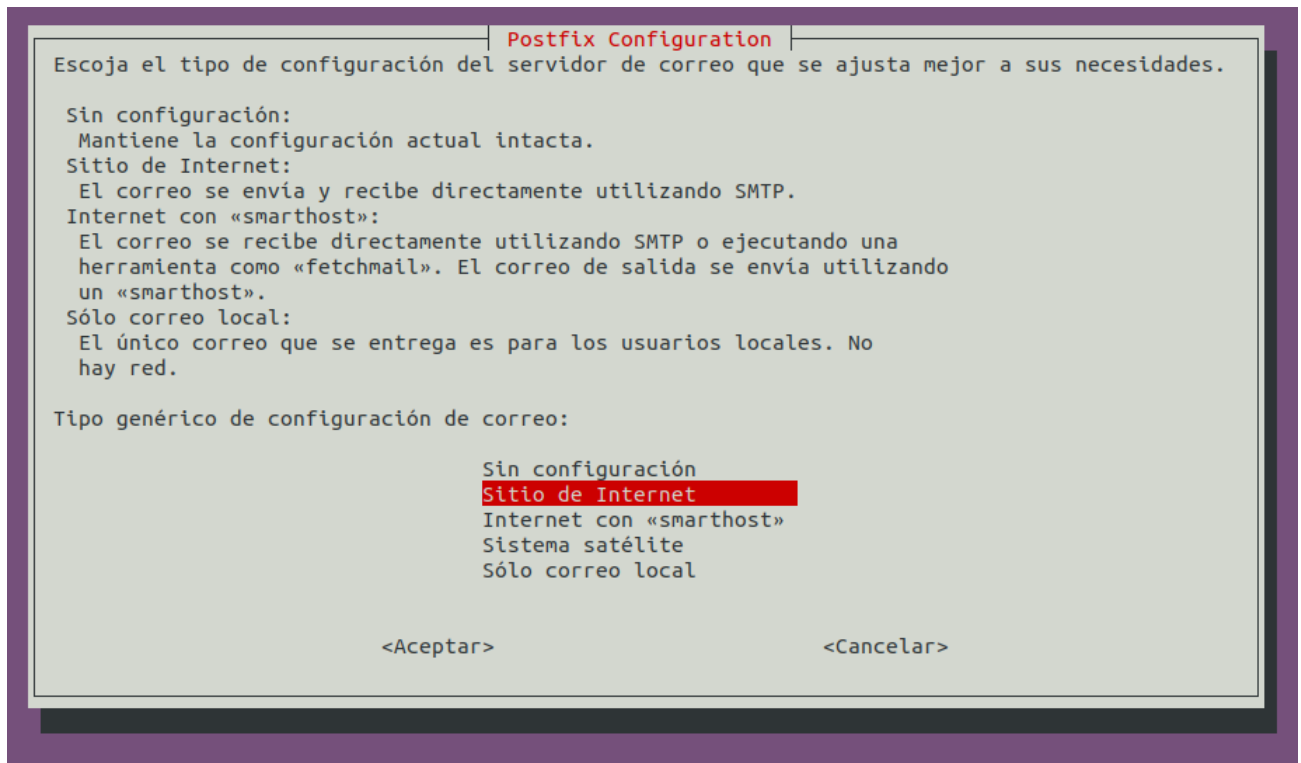


Ilustración 16: tripwire installer

Configuramos nuestro dominio:

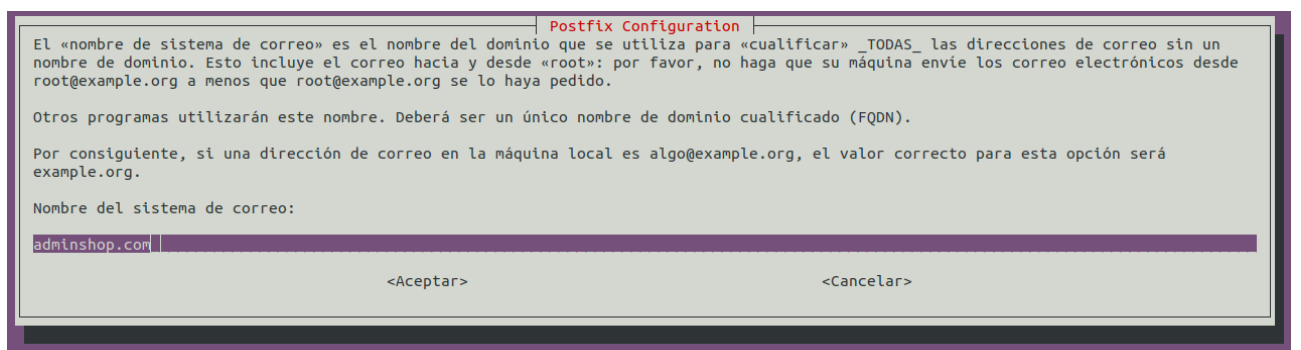


Ilustración 17: tripwire dominio

Damos a Si hasta que nos pida una contraseña y la instalación finalizará.

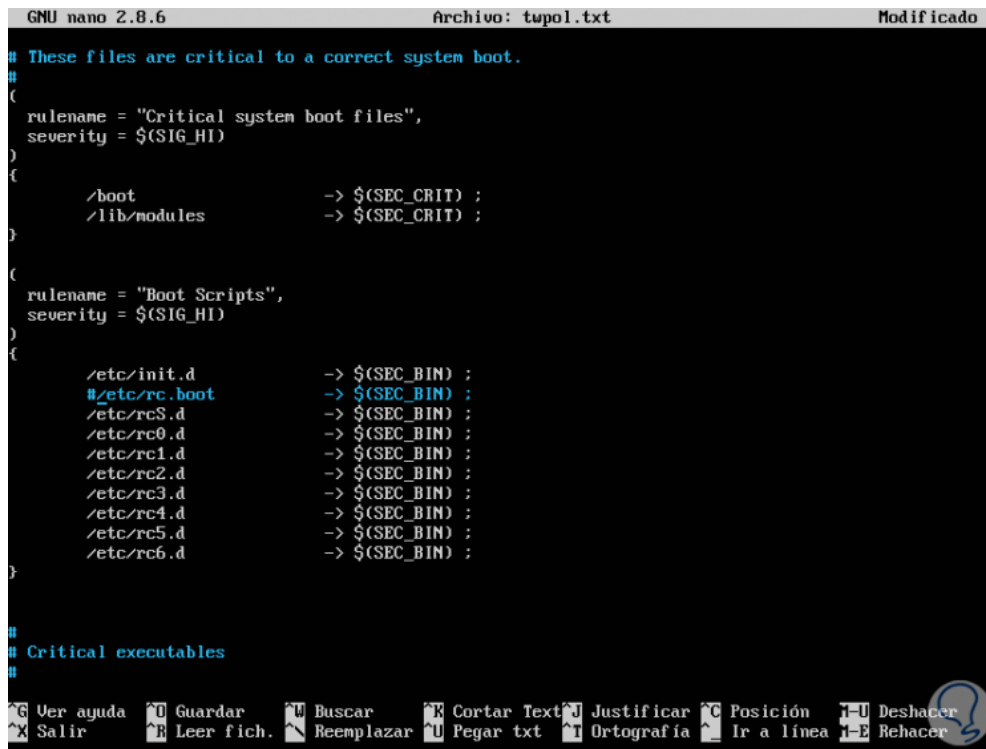
Una vez instalado, es necesario inicializar el sistema de la base de datos con el siguiente comando:

```
1. tripwire -init
```

Antes de editar la configuración de tripwire, debemos realizar el siguiente comando:

```
1. sh -c "tripwire --check | grep Filename > no-directory.txt"
```

En este paso debemos ir al directorio de configuración de tripwire y editar el archivo de configuración *rwpool.txt*



```
GNU nano 2.8.6 Archivo: twpol.txt Modificado
# These files are critical to a correct system boot.
#
(
  rulename = "Critical system boot files",
  severity = $(S16_HI)
)
(
  /boot                -> $(SEC_CRIT) ;
  /lib/modules          -> $(SEC_CRIT) ;
)

(
  rulename = "Boot Scripts",
  severity = $(S16_HI)
)
(
  /etc/init.d           -> $(SEC_BIN) ;
  #/etc/rc.boot         -> $(SEC_BIN) ;
  /etc/rcS.d            -> $(SEC_BIN) ;
  /etc/rc0.d            -> $(SEC_BIN) ;
  /etc/rc1.d            -> $(SEC_BIN) ;
  /etc/rc2.d            -> $(SEC_BIN) ;
  /etc/rc3.d            -> $(SEC_BIN) ;
  /etc/rc4.d            -> $(SEC_BIN) ;
  /etc/rc5.d            -> $(SEC_BIN) ;
  /etc/rc6.d            -> $(SEC_BIN) ;
)

#
# Critical executables
#

^G Ver ayuda  ^O Guardar  ^M Buscar   ^K Cortar Text ^J Justificar ^C Posición  ^U Deshacer
^X Salir      ^R Leer fich. ^N Reemplazar ^U Pegar txt  ^I Ortografía ^_ Ir a línea ^E Rehacer
```

Ilustración 18: Instalación tripwire 1

Comentamos la línea del *rc.boot*.

```

GNU nano 2.8.6                               Archivo: tupol.txt           Modificado
#
# Login and Privilege Raising Programs
#
(
    rulename = "Security Control",
    severity = $(SIG_MED)
)
(
    /etc/passwd          -> $(SEC_CONFIG) ;
    /etc/shadow          -> $(SEC_CONFIG) ;
)

#
# These files change every time the system boots
#
(
    rulename = "System boot changes",
    severity = $(SIG_HI)
)
(
    #/var/lock            -> $(SEC_CONFIG) ;
    #/var/run             -> $(SEC_CONFIG) ; # daemon PIDs
    /var/log              -> $(SEC_CONFIG) ;
)

# These files change the behavior of the root account
(
    rulename = "Root config files",
    severity = 100
)

```

Ilustración 19: Instalación tripwire 2

Comentamos las dos líneas dadas en la imagen.

```

GNU nano 2.8.6                               Archivo: tupol.txt           Modificado
    /root                -> $(SEC_CRIT) ; # Catch all additions to /root
    #/root/mail          -> $(SEC_CONFIG) ;
    #/root/Mail          -> $(SEC_CONFIG) ;
    #/root/.xsession-errors -> $(SEC_CONFIG) ;
    #/root/.xauth        -> $(SEC_CONFIG) ;
    #/root/.tcshrc       -> $(SEC_CONFIG) ;
    #/root/.sawfish      -> $(SEC_CONFIG) ;
    #/root/.pinerc       -> $(SEC_CONFIG) ;
    #/root/.mc           -> $(SEC_CONFIG) ;
    #/root/.gnome_private -> $(SEC_CONFIG) ;
    #/root/.gnome-desktop -> $(SEC_CONFIG) ;
    #/root/.gnome        -> $(SEC_CONFIG) ;
    #/root/.esd_auth     -> $(SEC_CONFIG) ;
    #/root/.elm          -> $(SEC_CONFIG) ;
    #/root/.cshrc        -> $(SEC_CONFIG) ;
    /root/.bashrc        -> $(SEC_CONFIG) ;
    #/root/.bash_profile -> $(SEC_CONFIG) ;
    #/root/.bash_logout  -> $(SEC_CONFIG) ;
    /root/.bash_history  -> $(SEC_CONFIG) ;
    #/root/.amandahosts  -> $(SEC_CONFIG) ;
    #/root/.addressbook.lu -> $(SEC_CONFIG) ;
    #/root/.addressbook  -> $(SEC_CONFIG) ;
    #/root/.Xresources    -> $(SEC_CONFIG) ;
    #/root/.Xauthority    -> $(SEC_CONFIG) -i ; # Changes Inode number on login
    #/root/.ICEauthority  -> $(SEC_CONFIG) ;
)

#
# Critical devices
#
(
    rulename = "Devices & Kernel information",
)

```

Ilustración 20: Instalación tripwire 3

Comentamos todas las líneas de la imagen.


```

GNU nano 2.8.6 Archivo: twpol.txt
)
(
/dev -> $(Device) ;
/dev/pts -> $(Device) ;
/dev/shm -> $(Device) ;
/dev/hugepages -> $(Device) ;
/dev/nqueue -> $(Device) ;
#/proc -> $(Device) ;
/proc/devices -> $(Device) ;
/proc/net -> $(Device) ;
/proc/tty -> $(Device) ;
/proc/cpuinfo -> $(Device) ;
/proc/modules -> $(Device) ;
/proc/mounts -> $(Device) ;
/proc/dma -> $(Device) ;
/proc/filesystems -> $(Device) ;
/proc/interrupts -> $(Device) ;
/proc/ioports -> $(Device) ;
/proc/scsi -> $(Device) ;
/proc/kcore -> $(Device) ;
/proc/self -> $(Device) ;
/proc/kmsg -> $(Device) ;
/proc/stat -> $(Device) ;
/proc/loadavg -> $(Device) ;
/proc/uptime -> $(Device) ;
/proc/locks -> $(Device) ;
/proc/meminfo -> $(Device) ;
/proc/misc -> $(Device) ;

# Other configuration files
[ 303 líneas escritas ]
Ver ayuda Guardar Buscar Cortar Text Justificar Posición Deshacer
Salir Leer fich. Reemplazar Pegar txt Ortografía Ir a línea Rehacer

```

Ilustración 21: Instalación tripwire 4

Comentamos y escribimos todas las líneas que vemos en la imagen.

Una vez hechos estos pasos, ejecutamos el siguiente comando:

1. `tripwire -update-policy -secure-mode low /etc/tripwire/twpol.txt`

Para regenerar el archivo de configuración de tripwire ejecutaremos la siguiente línea:

1. `twadmin -m P /etc/tripwire/twpol.txt`

Una vez hayamos hecho todo lo anterior tripwire estará funcionando correctamente.

Para ello hemos realizado el siguiente script que se ejecutará cada día por la noche:

```

1. #!/usr/bin/perl
2. use strict;
3. use warnings;
4.
5. use Mail::Sender;
6. use Email::Send::SMTP::Gmail;
7.
8. system('tripwire --check >datos.txt');
9. my $destination='xamo1998@gmail.com';
10. my ($mail,$error)=Email::Send::SMTP::Gmail->new( -smtp=>'smtp.gmail.com',
11.                                                    -login=>'xamo1998@gmail.com',
12.                                                    -pass=>'XXXXXXXXXXXXXXXXXXXX',
13.                                                    -layer=>'ssl');
14. print "Session error: $error" unless ($mail!=1);
15. $mail->send(-to=>$destination,-subject=>'Security report', -body=>'Here is your
    daily report!', -attachments=>'/usr/local/sbin/datos.txt');
16. $mail->bye;
17. unlink 'datos.txt';

```

Al ejecutar el código si vamos a nuestro correo podremos ver como se ha enviado correctamente:

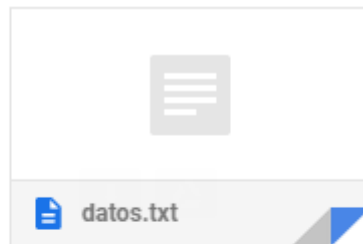
Security report Inbox x



xamo1998@gmail.com

to me ▾

Here is your daily report!



 Reply

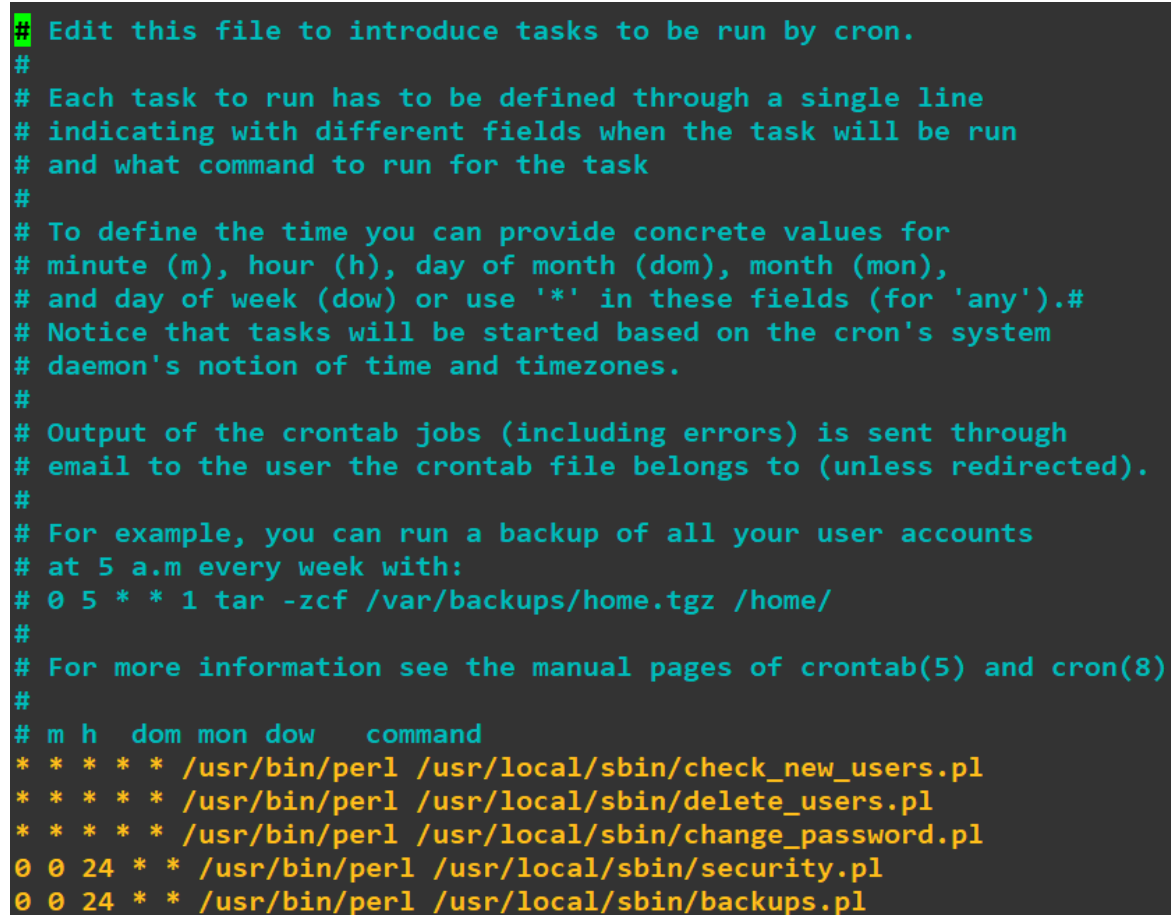
 Forward

Ilustración 22: Prueba tripwire report

Ficheros CRON

Aquí explicaremos todos los cron que tenemos en el servidor, para añadir un script al cron tenemos varias opciones, una de ellas es mediante el comando `crontab -e`, otra es modificando el archivo `/etc/crontab`.

Mostraremos una captura con los scripts que tenemos en el cron y cuál es su funcionamiento:



```

# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
* * * * * /usr/bin/perl /usr/local/sbin/check_new_users.pl
* * * * * /usr/bin/perl /usr/local/sbin/delete_users.pl
* * * * * /usr/bin/perl /usr/local/sbin/change_password.pl
0 0 24 * * /usr/bin/perl /usr/local/sbin/security.pl
0 0 24 * * /usr/bin/perl /usr/local/sbin/backups.pl
```

Ilustración 23: Ficherto cron

- **check_new_user.pl:** Script que se ejecuta cada 1 minuto y busca en la base de datos de usuarios verificados todos los usuarios y uno a uno los borra de esa base de datos y los mete en la base de datos de usuarios finales, también crea el usuario en Linux
- **delete_users.pl:** Busca en la base de datos de users_to_delete y por cada usuario que haya en esa tabla lo elimina de Linux.
- **change_password.pl:** Busca en la base de datos de users_to_change y por cada usuario que haya en esa tabla lo elimina de Linux y vuelve a crearlo con los nuevos datos.
- **security.pl:** Realiza un check de todo el sistema y le manda la salida al correo del administrador.
- **backups.pl:** Realiza las copias de seguridad.

Particiones y cuotas

Los usuarios deben tener asignadas unas cuotas por lo que hemos optado por crear un sistema de fichero donde guardar los `/home` de los usuarios y aplicar ahí las cuotas. Para ello debemos realizar los siguientes pasos.

Sistema de ficheros `/mnt/home`

Primero debemos crear un sistema de ficheros en una carpeta, por ejemplo `/home/user/SistemaFich`, para hacer esto realizamos la siguiente orden:

```
1. dd if=/dev/zero of=/home/hector/SistemaFich count=10240 bs=10240
```

Esta orden nos creara un archivo de aproximadamente 100 Mb lleno de ceros. El siguiente paso es dar formato a este archivo, en nuestro caso hemos elegido ext4 por lo que la orden sería la siguiente:

```
1. mkfs.ext4 /home/hector/SistemaFich
```

El siguiente paso es montar el sistema de ficheros en el directorio que deseemos, en nuestro caso `/mnt/home` por lo que las órdenes a utilizar serían:

```
1. mkdir /mnt/home
2. mount -o loop /home/hector/SistemaFich /mnt/home
```

Añadimos una línea al fichero `/etc/fstab` el cual se encarga de cargar las particiones del sistema cuando se arranca, para ello escribimos la siguiente línea, con cuidado ya que cada campo se debe separar con un tabulador:

```
1. /home/hector/SistemaFich /mnt/home ext4 defaults,usrquota 0 0
```

Una vez hecho esto realizamos la siguiente orden:

```
1. mount -a
```

La partición estará configurada correctamente, si queremos usarla debemos reiniciar el sistema con la orden `reboot`.

Aplicar cuotas

Para aplicar las cuotas al sistema de ficheros debemos seguir los siguientes pasos, primero escribimos la siguiente línea:

```
1. mount -o remount /mnt/home
```

Una vez hecho esto realizamos las siguientes ordenes:

```
1. quotacheck -cugm /mnt/home
2. quotaon -ugv /mnt/home/
```

Con esto ya tendríamos las cuotas activadas en `/mnt/home` para poder ver un resumen de las cuotas en ese directorio escribimos la siguiente orden:

```
1. repquota /mnt/home
```

Obtenemos una salida como esta:

```
root@hector-HUAWEI-MateBook-D:/var/www/html# repquota /mnt/home/
*** Report for user quotas on device /dev/loop16
Block grace time: 7days; Inode grace time: 7days

```

User		used	Block limits				File limits		
			soft	hard	grace		used	soft	hard
www-data	--	1	0	0			1	0	0

```
root@hector-HUAWEI-MateBook-D:/var/www/html#
```

Ilustración 24: Ejemplo repquota

Para que cada usuario tenga una cuota en concreto lo veremos más adelante en este informe, en concreto en [esta](#) sección:

Configurar CGI

Para la comunicación entre el servidor y las páginas web hemos usado CGI, en este apartado vamos a ver como configurarlo para la correcta ejecución de los ficheros .pl.

NOTA: Este paso se debe realizar **después** de la [instalación de apache](#) pero por motivos de presentación en esta memoria lo haremos antes.

Primero debemos activar cgi, para ello realizamos los siguientes comandos:

```
1. a2dismod mpm_event
2. a2enmod mpm_prefork
3. a2enmod cgi
```

Después reiniciamos apache con la siguiente orden:

```
1. systemctl restart apache2
```

Ahora debemos configurar el fichero `/etc/apache2/conf-available/cgi-enabled` y copiar lo siguiente:

```
1. <Directory "/var/www/html/cgi-enabled">
2.     Options +ExecCGI
3.     AddHandler cgi-script .cgi .pl .py
4. </Directory>
```

Después activamos la configuración y reiniciamos apache con las siguientes instrucciones:

```
1. sudo a2enconf cgi-enabled
2. sudo service apache2 restart
```

Una vez hecho esto podremos usar ficheros perl para la ejecución de tareas junto con apache.

Ficheros log

En nuestra práctica utilizamos un fichero log que almacena la hora y el tipo de acceso cada vez que un usuario intenta hacer login en nuestro servidor.

No hemos utilizado ningún tipo de comando ya que desde perl nos ha parecido más fácil abrir un fichero y volcar los datos añadiéndolos al final.

Descripción básica de los servidores

En este apartado veremos cómo hemos instalado y configurado todos los servidores que usamos en nuestro servidor.

SSH

Primero debemos instalar SSH, para ello escribimos:

1. `apt-get install ssh`

Después debemos cambiar la configuración del ssh que se encuentra en `/etc/ssh/sshd_config`, debemos escribir las siguientes líneas:

1. `Port 1060`
2. `Protocol 2`
3. `LoginGraceTime 2m`
4. `PermitRootLogin no`
5. `MaxStartups 5`

Una vez configurado ssh tan solo tenemos que reiniciar el servicio de la siguiente manera:

1. `systemctl restart ssh`

Apache2

En este apartado veremos cómo instalar y configurar Apache.

Introducción

El servidor HTTP Apache es un servidor web HTTP de código abierto, para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras. Es una de las herramientas más usadas para la gestión de servidores que requieren HTTP.

Instalación Apache2

A continuación, procedemos a ver los pasos seguidos en la instalación y configuración de apache. Lo primero de todo antes de instalar nada, siempre es actualizar el sistema:

Una vez actualizado, instalamos el paquete de apache:

```
1. apt-get install apache2 apache2-doc apache2-utils
```

La salida que obtenemos es la siguiente:

```
root@mariaM-Aspire-A515-51G:/home/mariam/Escritorio# apt-get install apache2 apache2-doc apache2-utils
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  apache2-bin apache2-data libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.2-0
Paquetes sugeridos:
  apache2-suexec-pristine | apache2-suexec-custom
Se instalarán los siguientes paquetes NUEVOS:
  apache2 apache2-bin apache2-data apache2-doc apache2-utils libapr1
  libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.2-0
0 actualizados, 10 nuevos se instalarán, 0 para eliminar y 6 no actualizados.
Se necesita descargar 5.410 kB de archivos.
Se utilizarán 30,9 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] ☐
```

Ilustración 25: Módulos Apache2

Podemos destacar que los paquetes para apache tan solo ocupan 30.9 MB.

Si ahora abrimos nuestro navegador y escribimos en la url: *localhost* obtenemos la siguiente página web:



Ilustración 26: Página por defecto Apache2

Como hemos visto en el punto anterior si escribimos *localhost* en la url del navegador nos devuelve la página web por defecto, también podemos acceder a esta página mediante la ip de nuestro servidor, para obtener la ip nos basta con usar la orden *ip addr* y obtendremos una salida como esta:

```
root@hector-HUAWEI-MateBook-D:/home/hector# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: wlp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether f8:59:71:50:5f:b9 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.160/24 brd 192.168.0.255 scope global dynamic noprefixroute wlp1s0
        valid_lft 77218sec preferred_lft 77218sec
    inet6 fe80::964d:6d2c:fd70:9114/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
root@hector-HUAWEI-MateBook-D:/home/hector#
```

Ilustración 27: Orden ip addr

Como vemos, esta orden nos devuelve la ip privada, al igual que con *localhost*, podemos usar la ip. Si queremos acceder a nuestro servidor por un nombre de dominio, por ejemplo, dominio.com debemos modificar el fichero `/etc/hosts` para que al intentar resolver el nombre del dominio no sea necesario hacer una petición de DNS.

El archivo, debe quedar así:

```
root@hector-HUAWEI-MateBook-D:/home/hector# cat /etc/hosts
127.0.0.1      www.adminshop.com adminshop.com
127.0.0.1      localhost
127.0.1.1      hector-HUAWEI-MateBook-D

# The following lines are desirable for IPv6 capable hosts
::1           ip6-localhost ip6-loopback
fe00::0       ip6-localnet
ff00::0       ip6-mcastprefix
ff02::1       ip6-allnodes
ff02::2       ip6-allrouters
root@hector-HUAWEI-MateBook-D:/home/hector#
```

Ilustración 28: Modificación de fichero hosts

A la izquierda debemos poner la dirección ip local (127.0.0.1) y a la derecha el nombre de dominio que le asociamos a dicha ip.

Por último, para que nuestro dominio este correctamente configurado debemos modificar el fichero: `/etc/apache/sites-available/000-default.conf`, y escribir lo siguiente:

1. `ServerName adsysshop.com:80`
2. `ServerAlias www.adsysshop.com`
3. `ServerAdmin webmaster@adsysshop.com`
4. `DocumentRoot /var/www/html`

El siguiente paso es editar el fichero de configuración de Apache:

1. `pico /etc/apache/apache.conf`

Insertamos las siguientes líneas de código:

1. `<IfModule mpm_prefork_module>`
2. `StartServers 5`
3. `MinSpareServers 5`
4. `MaxSpareServers 10`
5. `MaxClients 100`
6. `MaxRequestsPerChild 0`
7. `</IfModule>`

```

GNU nano 2.9.3 /etc/apache2/apache2.conf

LogFormat "%v:%p %h %l %u %t \"%r\" %>s %O \"%{Referer}i\" \"%{User-Agent}i\"" vhost_combined
LogFormat "%h %l %u %t \"%r\" %>s %O \"%{Referer}i\" \"%{User-Agent}i\"" combined
LogFormat "%h %l %u %t \"%r\" %>s %O" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent

# Include of directories ignores editors' and dpkg's backup files,
# see README.Debian for details.

# Include generic snippets of statements
IncludeOptional conf-enabled/*.conf

# Include the virtual host configurations:
IncludeOptional sites-enabled/*.conf

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
<IfModule mpm_prefork_module>
    StartServers 5
    MinSpareServers 5
    MaxSpareServers 10
    MaxClients 100
    MaxRequestsPerChild 0
</IfModule>

```

Ilustración 29: configuración apache2.conf

El significado de los campos es el siguiente:

- **StartServers:** Número de procesos que se ejecutan al iniciar Apache.
- **MinSpareServers:** Mínima cantidad de procesos que se mantienen en espera.
- **MaxSpareServers:** Cantidad máxima de procesos en espera
- **MaxClients:** Número máximo de clientes que se pueden ejecutar
- **MaxRequestsPerChild:** Número de peticiones que atiende cada hilo de ejecución

Por último, tenemos que reiniciar el servicio para actualizar la configuración:

1. `systemctl restart apache2`

Habilitar fichero .htaccess

El archivo .htaccess es un archivo de configuración muy importante que se aplica a cada subcarpeta de nuestro servidor. En este archivo podemos hacer cosas como bloquear ciertas páginas, pedir autenticación para cierta página...

Para configurarlo, primero debemos activar el módulo rewrite de la siguiente manera:

1. `a2enmod rewrite`

Posteriormente debemos crear una sección *Directory* dentro de la sección *VirtualHost* que encontramos en el archivo: `/etc/apache/sites-available/000-default.conf`, en este archivo introducimos lo siguiente:

1. `<Directory /var/www/html>`
2. `Options Indexes FollowSymlinks MultiViews`
3. `AllowOverride All`
4. `Require all granted`
5. `Order allow,deny`
6. `allow from all`
7. `</Directory>`

Una vez hemos modificado es archivo tan solo reiniciamos apache y estará funcionando:

```
1. systemctl restart apache2
```

Mover HTML

Este paso es el más simple ya que si queremos que aparezca nuestra página web cuando la buscamos en el navegador en vez de la página de bienvenida de apache debemos copiar nuestros archivos html, css, js dentro de, en nuestro caso, `/var/www/html/`.

Cuando hagamos esto y pongamos la ip del servidor nos debería salir algo como esto:

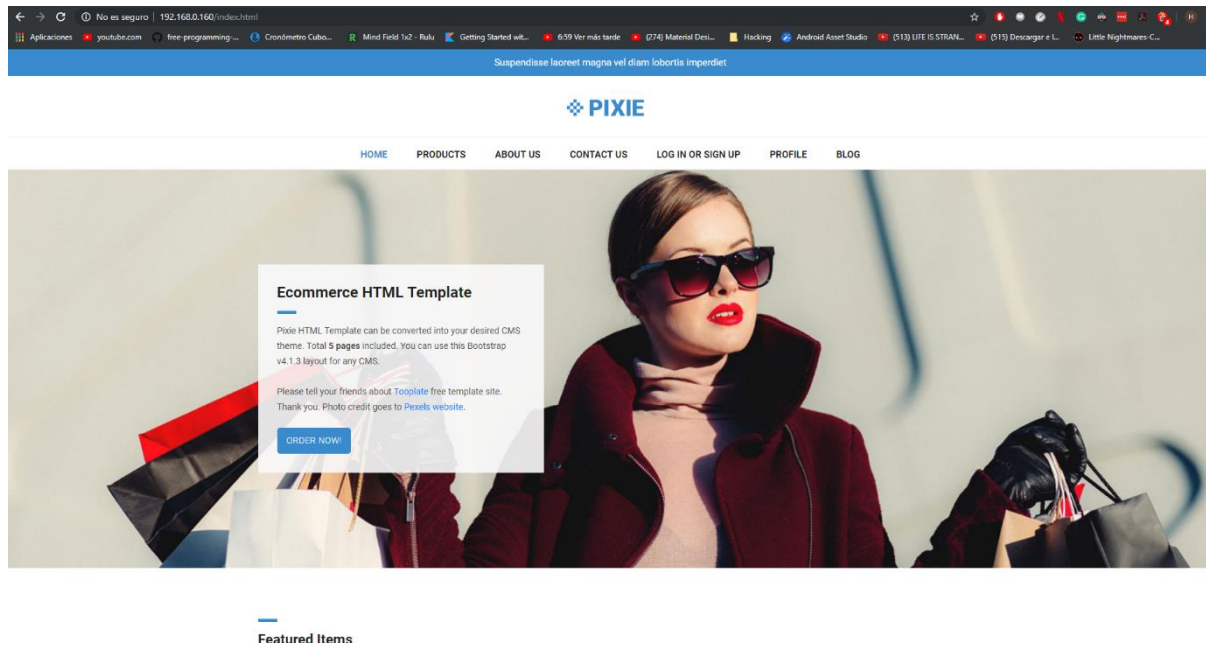


Ilustración 30: Home Page

Protección anti-ataques DoS

Un ataque de denegación de servicio tiene como objetivo inhabilitar el uso de un sistema, una aplicación o una máquina, con el fin de bloquear el servicio para el que está destinado. Este ataque puede afectar, tanto a la fuente que ofrece la información como puede ser una aplicación o el canal de transmisión, como a la red informática.

Debido a esto vamos a instalar un módulo que nos permita protegernos de este tipo de ataques, para ello instalamos el paquete *libapache2-mod-evasive*:

```
1. apt-get install libapache2-mod-evasive
```

Creamos una carpeta donde se almacenarán los registros de actividad:

```
1. mkdir -p /var/log/apache2/evasive
```

Damos permisos a apache para que pueda escribir en esta carpeta:

```
1. chown -R www-data:root /var/log/apache2/evasive
```

A partir de ahora si alguien nos ataca, en la carpeta de registros */var/log/apache2/evasive* se creará un archivo de texto cuyo nombre será la IP del atacante.

Una vez instalado el módulo debemos configurarlo, para ello debemos de editar el archivo */etc/apache2/mods-available/mod-evasive.load*

El archivo estará vacío por lo que escribimos lo siguiente:

```
1. LoadModule evasive20_module /usr/lib/apache2/modules/mod_evasive20.so
2. DOSHashTableSize 2048
3. DOSPageCount 20
4. DOSSiteCount 30
5. DOSPageInterval 1.0
6. DOSSiteInterval 1.0
7. DOSBlockingPeriod 10.0
8. DOSLogDir "/var/log/apache2/evasive"
9. DOSEmailNotify root@adsysshop.com
```

Por último, reiniciamos apache para que el archivo de configuración se cargue:

```
1. systemctl restart apache2
```

Página segura HTTPS con SSL-RSA

Si queremos tener la posibilidad de establecer conexiones cifradas a través de HTTPS, deberemos tener un certificado SSL. SSL (*Secure Socket Layer*) es el protocolo de cifrado más usado en la web y RSA es el algoritmo que cifrará la información que se envía a través de SSL.

En primer lugar, tenemos que activar el módulo SSL de Apache, para ello ejecutamos:

```
1. a2enmod ssl
```

Después debemos crear un archivo de configuración para los sitios seguros de nuestro servidor ejecutando el siguiente comando:

```
1. a2ensite default-ssl
```

Por último, reiniciamos apache:

```
1. systemctl restart apache2
```

El siguiente paso es crear las claves RSA, una publica y otra privada con las cuales se podrá establecer una comunicación segura bidireccional entre el servidor y el cliente. Para la creación de estas claves usaremos el comando openssl que viene instalado por defecto en Debian 9.

Creamos una clave de longitud 2048 bits:

```
1. openssl genrsa -des3 -out server.key 2048
```

Obtendremos una salida como la siguiente:

```
root@marlam-Aspire-A515-51G:/home/mariam/Escritorio# openssl genrsa -des3 -out server.key 2048
Generating RSA private key, 2048 bit long modulus
.....+++
.....+++
e is 65537 (0x010001)
Enter pass phrase for server.key:
Verifying - Enter pass phrase for server.key:
root@marlam-Aspire-A515-51G:/home/mariam/Escritorio#
```

Ilustración 31: Creación clave server.key

Ahora debemos crear el certificado en base a la clave que acabamos de generar. Un certificado es un archivo que acredita al navegador que la conexión proviene del servidor al que estamos contactando y que no hay usurpación de identidad. A través de él se proporciona la información para establecer la conexión segura. Para esto ejecutamos la siguiente orden:

```
1. openssl req -new -key server.key -out server.csr
```

Nos pedirá una serie de datos, lo rellenamos y nos debería salir algo como esto:

```

root@mariaM-Aspire-A515-51G:/home/mariam/Escritorio# openssl req -new -key server.key -out server.csr
Enter pass phrase for server.key:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:ES
State or Province Name (full name) [Some-State]:Spain
Locality Name (eg, city) []:Salamanca
Organization Name (eg, company) [Internet Widgits Pty Ltd]:adminshop.com
Organizational Unit Name (eg, section) []:Blog
Common Name (e.g. server FQDN or YOUR name) []:Hector Chamorro
Email Address []:admin@adminshop.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:pass
An optional company name []:\n
root@mariaM-Aspire-A515-51G:/home/mariam/Escritorio#

```

Ilustración 32: Creación de certificado

Por último, habrá que firmar el certificado para que el cliente tenga la certeza de que realmente ha sido enviado por nuestro servidor y no por un atacante. Para esto ejecutamos la siguiente orden:

1. `openssl x509 -req -days 365 -in server.csr -signkey server.key -out server.crt`

El parámetro 365 nos dice que la validez del certificado será de 1 año, aunque después de este periodo de tiempo las conexiones seguirán siendo cifradas y seguras.

Una vez ejecutadas las ordenes anterior disponemos de 2 archivos (en el directorio donde hayamos ejecutado estas órdenes) que debemos copiarlos a `/etc/ssl/` para que sean reconocidos por Apache. Para esto, ejecutamos las siguientes líneas:

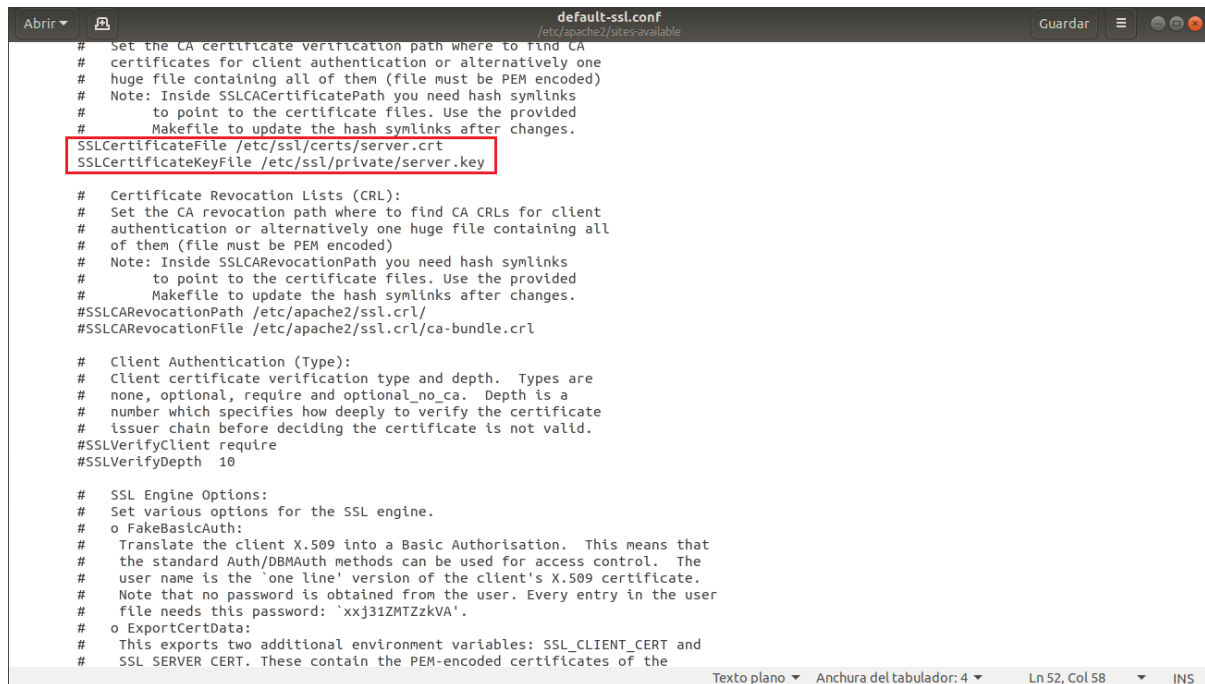
1. `cp server.crt /etc/ssl/certs/`
2. `cp server.key /etc/ssl/private/`

Editamos el archivo de configuración para sitios seguros `default-ssl.conf` que se encuentra en el directorio `/etc/apache2/sites-available`.

En este archivo debemos modificar las siguientes líneas con el contenido que aparece:

1. `SSLCertificateFile /etc/ssl/certs/server.crt`
2. `SSLCertificateKeyFile /etc/ssl/private/server.key`

Nos quedaría algo como esto:



```

# Set the CA certificate verification path where to find CA
# certificates for client authentication or alternatively one
# huge file containing all of them (file must be PEM encoded)
# Note: Inside SSLCertificatePath you need hash symlinks
# to point to the certificate files. Use the provided
# Makefile to update the hash symlinks after changes.
SSLCertificateFile /etc/ssl/certs/server.crt
SSLCertificateKeyFile /etc/ssl/private/server.key

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

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

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

```

Ilustración 33: Configuración de ficheros SSL

Una vez hecho esto reiniciamos apache:

1. `systemctl restart apache2`

Al reiniciar el servicio nos pedirá que introduzcamos la contraseña que usamos para generar las claves.

La petición de esta contraseña ocurrirá siempre que reiniciemos Apache por lo que para automatizar esta tarea y que no la pida siempre haremos lo siguiente:

1. `cd /etc/ssl/private/`
2. `cp server.key server.key.otr`
3. `openssl rsa -in server.key.otr -out server.key`

Una vez todo listo podemos probar el funcionamiento a través de un navegador usando el protocolo HTTPS, para ello escribimos `https://nuestra_ip`, y obtendremos lo siguiente:

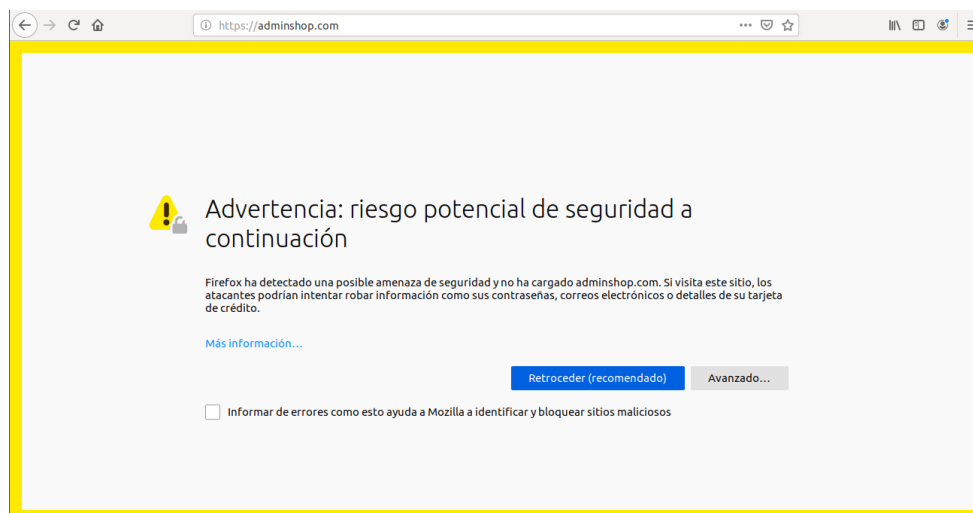


Ilustración 34: Conexión HTTPS

Nos aparece este mensaje debido a que nuestro certificado no se encuentra dentro del certificado raíz del navegador ya que no proviene de una entidad certificadora, sin embargo, esto no compromete la seguridad de la conexión, podemos ver el certificado que ha obtenido el navegador:



Ilustración 35: Certificado HTTPS

Al añadir la excepción vamos a nuestra página web. Si observamos los detalles del enlace, vemos que, por ejemplo, en Firefox se ha establecido una conexión cifrada:

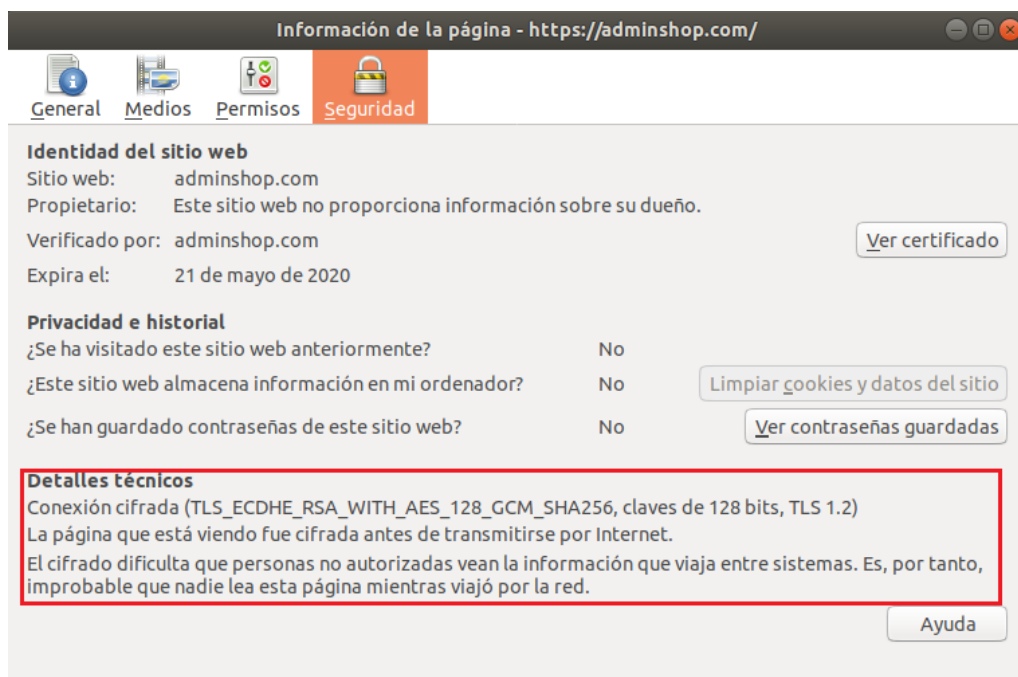


Ilustración 36: Prueba conexión cifrada

AWStats

AWStats es un programa que nos permite registrar la actividad de nuestro servidor Apache. Genera estadísticas de visitas que ordena temporalmente y tiene una gran cantidad de información útil para el administrador.

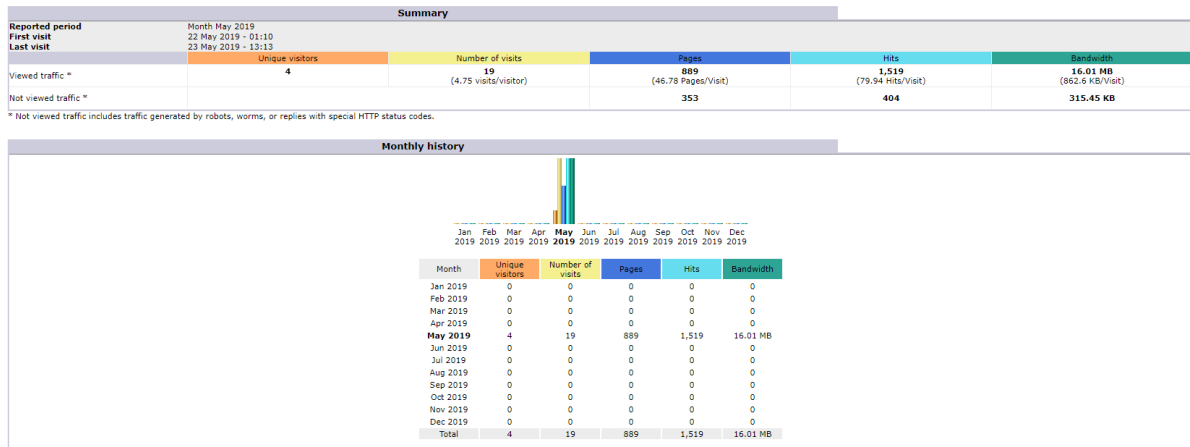


Ilustración 37: Ejemplo AWStats

Para la instalación de AWStats debemos instalar algunos paquetes:

1. `apt-get install awstats libnet-ip-perl libgeo-ipfree-perl`

Obtendremos una salida como esta:

```
root@marian-Aspire-A515-51G:/etc/ssl/private# apt-get install awstats
libnet-ip-perl libgeo-ipfree-perl
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
libnet-ip-perl ya está en su versión más reciente (1.26-1).
Se instalarán los siguientes paquetes NUEVOS:
  awstats libgeo-ipfree-perl libnet-xwhois-perl
0 actualizados, 3 nuevos se instalarán, 0 para eliminar y 6 no actuali-
zados.
Se necesita descargar 2.315 kB de archivos.
Se utilizarán 8.197 kB de espacio de disco adicional después de esta o-
peración.
Des:1 http://es.archive.ubuntu.com/ubuntu bionic/main amd64 awstats al-
l 7.6+dfsg-2 [1.839 kB]
Des:2 http://es.archive.ubuntu.com/ubuntu bionic/universe amd64 libgeo-
ipfree-perl all 1.151940-1 [455 kB]
Des:3 http://es.archive.ubuntu.com/ubuntu bionic/main amd64 libnet-xwh-
ois-perl all 0.90-4 [21,0 kB]
Descargados 2.315 kB en 2s (1.486 kB/s)
Seleccionando el paquete awstats previamente no seleccionado.
(Leyendo la base de datos ... 171750 ficheros o directorios instalados
actualmente.)
Preparando para desempaquetar .../awstats_7.6+dfsg-2_all.deb ...
Desempaquetando awstats (7.6+dfsg-2) ...
Seleccionando el paquete libgeo-ipfree-perl previamente no seleccionad
o.
Preparando para desempaquetar .../libgeo-ipfree-perl_1.151940-1_all.de
b ...
Desempaquetando libgeo-ipfree-perl (1.151940-1) ...
Seleccionando el paquete libnet-xwhois-perl previamente no seleccionad
o.
Preparando para desempaquetar .../libnet-xwhois-perl_0.90-4_all.deb ..
.
Desempaquetando libnet-xwhois-perl (0.90-4) ...
Configurando libgeo-ipfree-perl (1.151940-1) ...
Configurando libnet-xwhois-perl (0.90-4) ...
```

Ilustración 38: Instalación AWStats

Una vez realizado este paso, editaremos el archivo de configuración de AWStat que se encuentra en: `/etc/share/doc/awstats/examples/awstats_configure.pl` y cambiaremos las siguientes líneas con el siguiente contenido:

```

1. $AWSTATS_PATH='/usr/share/awstats';
2. $AWSTATS_ICON_PATH='/usr/share/awstats/icon';
3. $AWSTATS_CSS_PATH='/usr/share/awstats/css';
4. $AWSTATS_CLASSES_PATH='/usr/share/awstats/lib';
5. $AWSTATS_CGI_PATH='/usr/lib/cgi-bin';
6. $AWSTATS_MODEL_CONFIG='/usr/share/doc/awstat/examples/awstats.model.conf';

```

Para que Apache pueda acceder al programa para mostrar las estadísticas, hay que cambiar los permisos de este fichero de la siguiente forma:

```
1. chown www-data /usr/lib/cgi-bin/awstats.pl
```

Ahora debemos crear nuestra instancia de AWStats para nuestro dominio, para ello debemos editar/crear el archivo `/etc/awstats/awstats.midominio.com.conf`. Dentro de este archivo escribimos lo siguiente:

```

1. LogFile="/var/log/apache2/access_dplinux.log"
2. LogFormat=1
3. SiteDomain="adsysshop.com"
4. DNSLookup=0
5. LoadPlugin="tooltips"
6. LoadPlugin="geoipfree"

```

El siguiente paso es modificar los permisos de la carpeta de registros de Apache para que AWStats pueda tener acceso a ella, lo hacemos de la siguiente forma:

```
1. chmod 755 /var/log/apache2
```

Ahora configuramos el servicio cron de Linux. Este servicio ejecuta procesos periódicamente. Para añadir este archivo, abrimos el fichero `/etc/crontab` y añadimos la siguiente línea al final:

```
1. */10 * * * * root /usr/lib/cgi-bin/awstats.pl -config=adsysshop.com -
   update > /dev/null
```

Esta línea significa que cada 10 minutos se va a ejecutar el script `awstats.pl` con el usuario `root` usando la configuración para `adsysshop.com`.

Por último, vamos a restringir el acceso a estas estadísticas para que solo el administrador tenga permisos para verlas, para ello nos dirigimos al directorio `/usr/lib/cgi-bin/` y creamos un archivo `.htaccess` con el siguiente contenido:

```

1. <Files "awstats.pl">
2.     AuthName "Introduzca credenciales"
3.     AuthType Basic
4.     AuthUserFile /var/www/html/awstats/.htpasswd
5.     require valid-user
6. </Files>

```

Con este archivo estamos diciendo que cuando se acceda al archivo `awstats.pl` desde el navegador, se requerirán credenciales para poder visualizarlo. El siguiente paso es crear una carpeta llamada `awstats` dentro del directorio `/var/www/html/`, accedemos al directorio que acabamos de crear y escribimos lo siguiente:

```
1. htpasswd -c /var/www/html/awstats/.htpasswd adminUser
```

Nos pedirá la contraseña del usuario que le hayamos indicado.

Una vez hecho esto, editamos el fichero `/etc/apache2/sites-available/000-default.conf` y añadimos al final de la sección `VirtualHost` lo siguiente:

```

1. Alias /icon/ /usr/share/awstats/icon/
2. <Directory /usr/share/awstats/icon>
3.     Options None
4.     AllowOverride None
5.     Require all granted
6.     Allow from all
7. </Directory>
8. ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
9. <Directory "/usr/lib/cgi-bin">
10.     AllowOverride All
11.     Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
12.     Require all granted
13.     Allow from all
14. </Directory>
15. Alias /awstatsclasses "/usr/share/awstats/lib/"
16. Alias /awstats-icon/ "/usr/share/awstats/icon/"
17. Alias /awstatscss "/usr/share/doc/awstats/examples/css"
18. ScriptAlias /Estadisticas/ /usr/lib/cgi-bin/

```

Nos debe quedar algo como esto:

```

ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined

# For most configuration files from conf-available/, which are
# enabled or disabled at a global level, it is possible to
# include a line for only one particular virtual host. For example the
# following line enables the CGI configuration for this host only
# after it has been globally disabled with "a2disconf".
#Include conf-available/serve-cgi-bin.conf
Alias /icon/ /usr/share/awstats/icon/
<Directory /usr/share/awstats/icon>
    Options None
    AllowOverride None
    Require all granted
    Allow from all
</Directory>
ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
<Directory "/usr/lib/cgi-bin">
    AllowOverride All
    Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
    Require all granted
    Allow from all
</Directory>
Alias /awstatsclasses "/usr/share/awstats/lib/"
Alias /awstats-icon/ "/usr/share/awstats/icon/"
Alias /awstatscss "/usr/share/doc/awstats/examples/css"
ScriptAlias /Estadisticas/ /usr/lib/cgi-bin/
</VirtualHost>
# vim: syntax=apache ts=4 sw=4 sts=4 sr noet

```

Ilustración 39: Configuración Awstats

Con esto decimos que en algún momento querremos acceder al contenido del directorio `/usr/lib/cgi-bin` que es donde se encuentra AWStats, esto se hace a través de un enlace simbólico con la función `ScriptAlias`. También le hemos dicho que al acceder a la carpeta Estadísticas dentro del navegador ejecute el contenido de `/usr/lib/cgi-bin`.

Generamos las estadísticas por primera vez con la siguiente línea de código:

```
1. /usr/lib/cgi-bin/awstats.pl -config=adminshop.com -update
```

Habilitamos el módulo CGI si no estaba activado y reiniciamos Apache:

```
1. a2enmod cgi
2. systemctl restart apache2
```

En el caso de encontrar algún problema modificar la siguiente línea del archivo `/etc/awstats/awstats.conf`:

```
1. SiteDomain="tudominio.com"
```



```
# server name, used to reach the web site.
# If you share the same log file for several virtual web servers, this
# parameter is used to tell AWStats to filter record that contains records for
# this virtual host name only (So check that this virtual hostname can be
# found in your log file and use a personalized log format that include the
# %virtualname tag).
# But for multi hosting a better solution is to have one log file for each
# virtual web server. In this case, this parameter is only used to generate
# full URL's links when ShowLinksOnUrl option is set to 1.
# If analyzing mail log, enter here the domain name of mail server.
# Example: "myintranetserver"
# Example: "www.domain.com"
# Example: "ftp.domain.com"
# Example: "domain.com"
#
SiteDomain="adminshop.com"
```

Ilustración 40: AWStats.conf

Para acceder a las estadísticas nos debemos dirigir a <http://tudominio.com/Estadisticas/awstats.pl>, nos pedirá el usuario y contraseña que configuramos anteriormente y podremos ver una gran cantidad de información, mostraremos aquí algunas para visualizar el correcto funcionamiento de la herramienta:

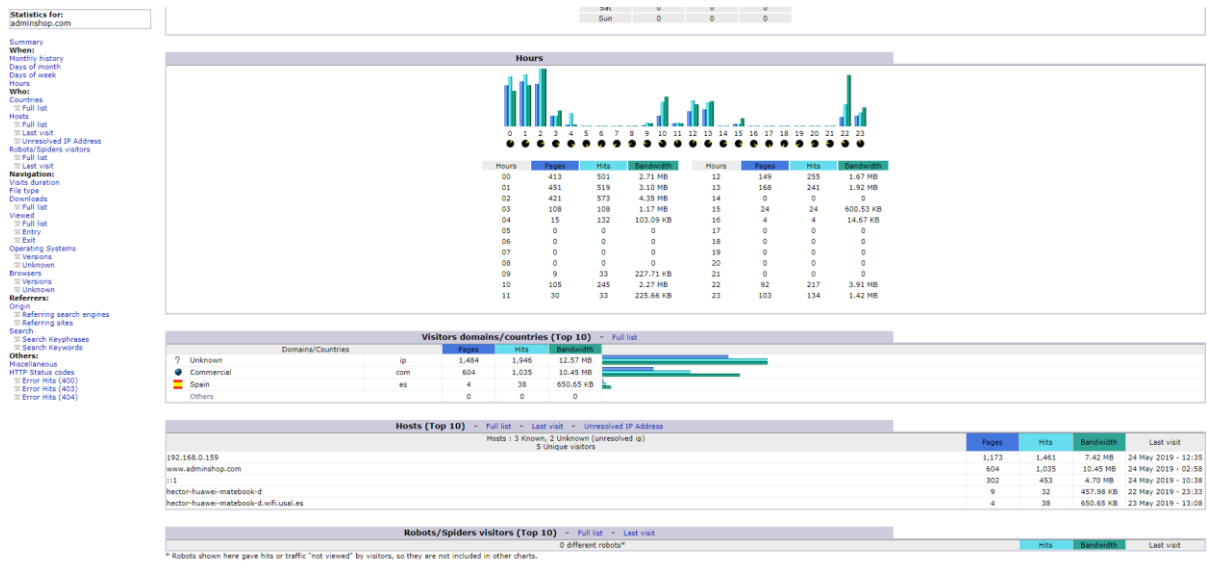


Ilustración 41: Prueba-1 AWStats

Visits duration			
Number of visits: 40 - Average: 1,183 s			
	Number of visits	Percent	
0s-30s	24	60 %	
30s-1mn	1	2.5 %	
2mn-5mn	1	2.5 %	
5mn-15mn	2	5 %	
15mn-30mn	1	2.5 %	
30mn-1h	3	7.5 %	
1h+	11	27.5 %	

Ilustración 42: Prueba-2 AWStats

HTTP Status codes			
HTTP Status codes*			
	Hits	Percent	Bandwidth
403 Forbidden	201	24 %	105.81 KB
404 Document Not Found (hits on favicon excluded)	165	27.9 %	83.49 KB
500 Internal server Error	88	14.8 %	75.54 KB
302 Moved temporarily (redirect)	71	12 %	43.98 KB
401 Unauthorized	51	8.6 %	35.99 KB
301 Moved permanently (redirect)	15	2.5 %	11.80 KB

* Codes shown here gave hits or traffic "not viewed" by visitors, so they are not included in other charts.

Ilustración 43: Prueba-3 AWStats

MariaDB

En este apartado veremos cómo configurar MariaDB junto con Mysql y phpMyAdmin para guardar la información de los usuarios que se registran en nuestro servidor así como información que usan otros servidores que hemos instalado en nuestro servidor.

El primer paso es instalar mariadb junto con los paquetes necesarios, la orden es:

1. apt-get install mariadb-server mariadb-client
2. apt-get install php php-cgi libapache2-mod-php php-common php-pear
3. apt-get install default-libmysqlclient-dev

La salida que obtenemos es la siguiente:

```
Se instalarán los siguientes paquetes NUEVOS:
galera-3 gawk libaio1 libconfig-inifiles-perl libdbd-mysql-perl
libdbi-perl libhtml-template-perl libjemalloc1 libreadline5
libsigsegv2 libterm-readkey-perl mariadb-client
mariadb-client-10.1 mariadb-client-core-10.1 mariadb-common
mariadb-server mariadb-server-10.1 mariadb-server-core-10.1 socat
0 actualizados, 19 nuevos se instalarán, 0 para eliminar y 6 no actualizados.
Se necesita descargar 23,3 MB de archivos.
Se utilizarán 180 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] S
Des:1 http://es.archive.ubuntu.com/ubuntu bionic/main amd64 libsigsegv
2 amd64 2.12-1 [14,7 kB]
Des:2 http://es.archive.ubuntu.com/ubuntu bionic/main amd64 gawk amd64
1:4.1.4+dfsg-1build1 [401 kB]
Des:3 http://es.archive.ubuntu.com/ubuntu bionic-updates/universe amd6
4 mariadb-common all 1:10.1.38-0ubuntu0.18.04.2 [16,1 kB]
Des:4 http://es.archive.ubuntu.com/ubuntu bionic/universe amd64 galera
-3 amd64 25.3.20-1 [947 kB]
Des:5 http://es.archive.ubuntu.com/ubuntu bionic/main amd64 libdbi-per
l amd64 1.640-1 [724 kB]
Des:6 http://es.archive.ubuntu.com/ubuntu bionic/main amd64 libconfig-
inifiles-perl all 2.94-1 [40,4 kB]
Des:7 http://es.archive.ubuntu.com/ubuntu bionic-updates/main amd64 li
baio1 amd64 0.3.110-5ubuntu0.1 [6.476 B]
Des:8 http://es.archive.ubuntu.com/ubuntu bionic/main amd64 libreadlin
e5 amd64 5.2+dfsg-3build1 [99,5 kB]
Des:9 http://es.archive.ubuntu.com/ubuntu bionic-updates/universe amd6
4 mariadb-client-core-10.1 amd64 1:10.1.38-0ubuntu0.18.04.2 [4.756 kB]
Des:10 http://es.archive.ubuntu.com/ubuntu bionic/universe amd64 libje
malloc1 amd64 3.6.0-11 [82,4 kB]
Des:11 http://es.archive.ubuntu.com/ubuntu bionic-updates/universe amd
64 mariadb-client-10.1 amd64 1:10.1.38-0ubuntu0.18.04.2 [5.621 kB]
52% [11 mariadb-client-10.1 4.989 kB/5.621 kB 89%]
```

Ilustración 44: Instalación MariaDB

Después procedemos a instalar phpmyadmin, para ello utilizaremos la siguiente orden:

1. apt-get install phpmyadmin php-mbstring pgp-gettext

Obtenemos la siguiente salida:

```

root@maria-Aspire-A515-51G:/etc/ssl/private# apt-get install phpmyadm
in php-mbstring php-gettext
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  dbconfig-common dbconfig-mysql javascript-common
  libapache2-mod-php7.2 libcurl4 libjs-jquery libjs-sphinxdoc
  libjs-underscore libzip4 php php-bz2 php-cli php-common php-curl
  php-gd php-mysql php-pear php-php-gettext php-phpseclib php-tcpdf
  php-xml php-zip php7.2 php7.2-bz2 php7.2-cli php7.2-common
  php7.2-curl php7.2-gd php7.2-json php7.2-mbstring php7.2-mysql
  php7.2-openssl php7.2-readline php7.2-xml php7.2-zip
Paquetes sugeridos:
  php-libsodium php-mcrypt php-gmp php-imagick
Se instalarán los siguientes paquetes NUEVOS:
  dbconfig-common dbconfig-mysql javascript-common
  libapache2-mod-php7.2 libcurl4 libjs-jquery libjs-sphinxdoc
  libjs-underscore libzip4 php php-bz2 php-cli php-common php-curl
  php-gd php-gettext php-mbstring php-mysql php-pear php-php-gettext
  php-phpseclib php-tcpdf php-xml php-zip php7.2 php7.2-bz2
  php7.2-cli php7.2-common php7.2-curl php7.2-gd php7.2-json
  php7.2-mbstring php7.2-mysql php7.2-openssl php7.2-readline
  php7.2-xml php7.2-zip phpmyadmin
0 actualizados, 38 nuevos se instalarán, 0 para eliminar y 6 no actual
izados.
Se necesita descargar 18,0 MB de archivos.
Se utilizarán 72,3 MB de espacio de disco adicional después de esta op
eración.
¿Desea continuar? [S/n]

```

Ilustración 45: Instalación PhpMyAdmin

En los pasos de la instalación seleccionamos lo siguiente:

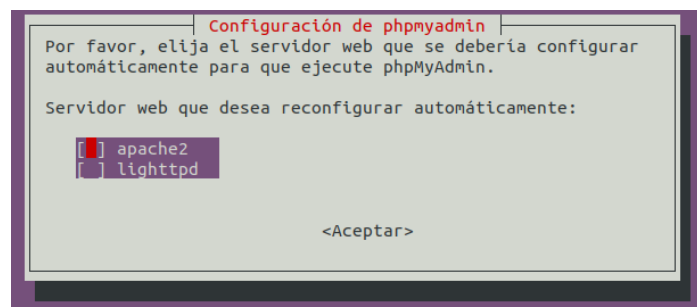


Ilustración 46: Configuración phpmyadmin 1

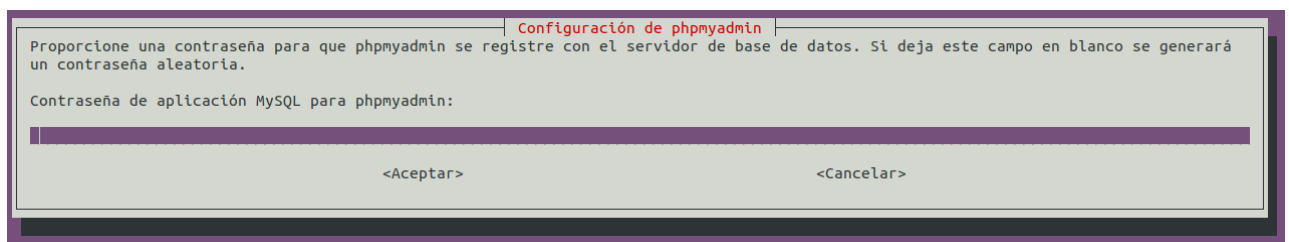


Ilustración 47: Configuración phpmyadmin 2

En este paso seleccionamos la contraseña que queramos.

Una vez llegado este punto, debemos crear un usuario en mariadb que tenga los privilegios de todas las bases de datos, para ello realizamos lo siguiente:

1. mariadb -u root -p
2. **CREATE USER** 'admin'@'localhost' IDENTIFIED BY 'password';


```
3. GRANT ALL PRIVILEGES ON *.* TO 'admin'@'localhost' WITH GRANT OPTION;  
4. FLUSH PRIVILEGES;  
5. exit;
```

Por último, para activar phpmyadmin junto con apache debemos crear un enlace simbólico del archivo de configuración de apache de la siguiente manera:

```
1. ln -s /etc/phpmyadmin/apache.conf /etc/apache2/conf-available/phpmyadmin.conf
```

Activamos phpmyadmin y reiniciamos Apache de la siguiente forma:

```
1. a2enconf phpmyadmin  
2. systemctl restart apache2
```

Accedemos a <http://midominio.com/phpmyadmin> e iniciamos sesión con el usuario que hemos creado:

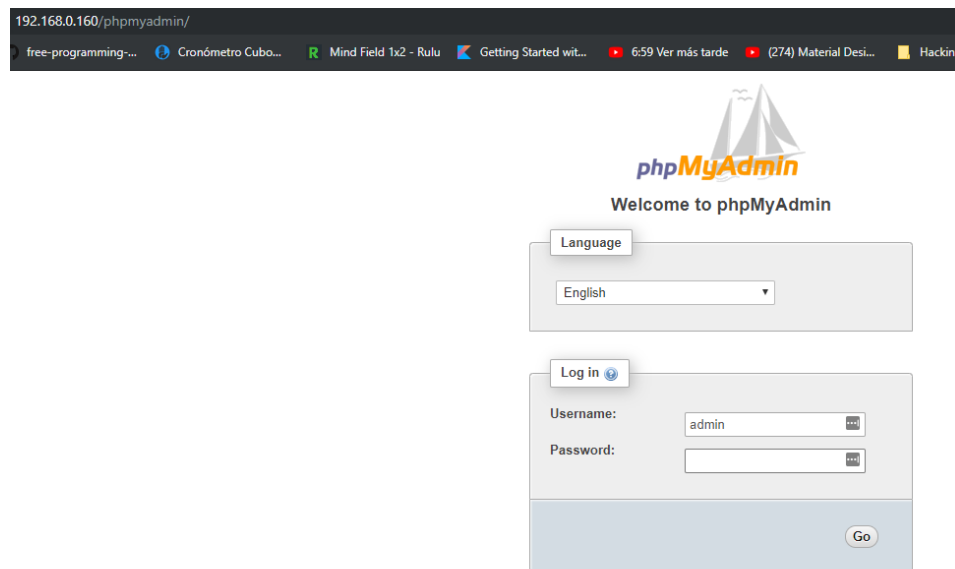


Ilustración 48: Login PhpMyAdmin

Al iniciar sesión nos debemos ir a la opción *New* y crear, en nuestro caso, la base se llamará *Users*

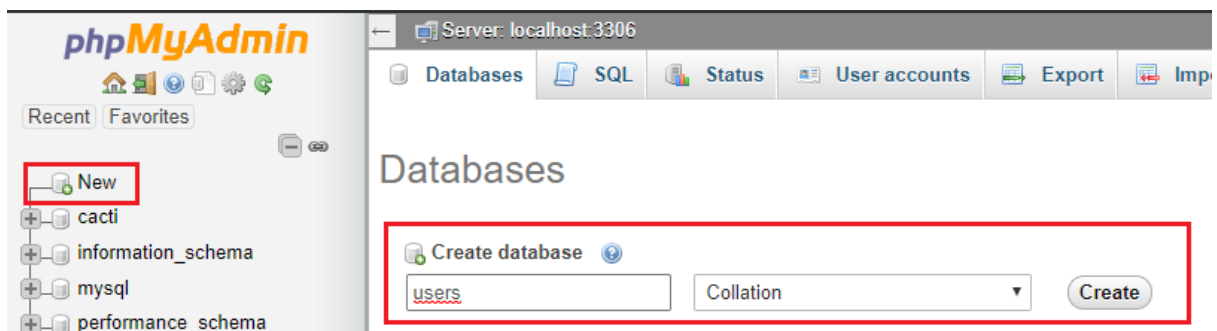


Ilustración 49: Crear Base de datos users

Una vez creada esta base de datos accedemos a ella y dentro al apartado de SQL donde pondremos las sentencias SQL para la creación de las tablas necesarias en nuestro servidor:

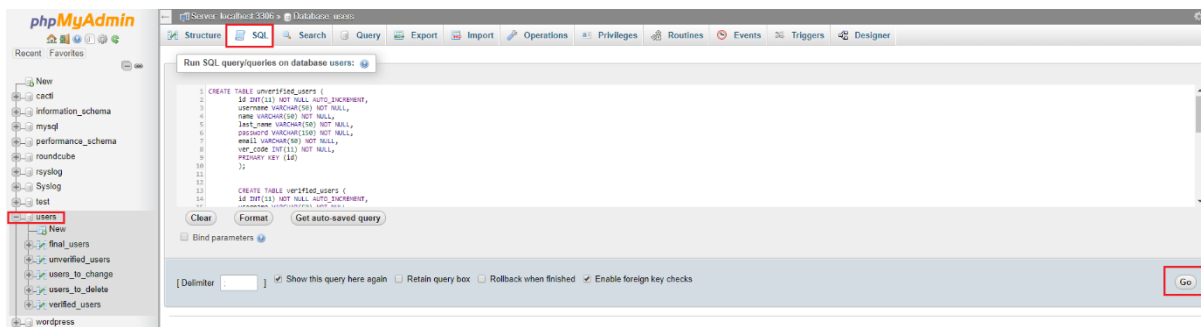


Ilustración 50: Crear tablas

Las sentencias de SQL que usaremos son las siguientes:

```

1. CREATE TABLE unverified_users (
2.     id INT(11) NOT NULL AUTO_INCREMENT,
3.     username VARCHAR(50) NOT NULL,
4.     name VARCHAR(50) NOT NULL,
5.     last_name VARCHAR(50) NOT NULL,
6.     password VARCHAR(150) NOT NULL,
7.     email VARCHAR(50) NOT NULL,
8.     ver_code INT(11) NOT NULL,
9.     PRIMARY KEY (id)
10. );
11.
12. CREATE TABLE verified_users (
13.     id INT(11) NOT NULL AUTO_INCREMENT,
14.     username VARCHAR(50) NOT NULL,
15.     name VARCHAR(50) NOT NULL,
16.     last_name VARCHAR(50) NOT NULL,
17.     password VARCHAR(150) NOT NULL,
18.     email VARCHAR(50) NOT NULL,
19.     PRIMARY KEY (id)
20. );
21. CREATE TABLE final_users (
22.     id INT(11) NOT NULL AUTO_INCREMENT,
23.     username VARCHAR(50) NOT NULL,
24.     name VARCHAR(50) NOT NULL,
25.     last_name VARCHAR(50) NOT NULL,
26.     password VARCHAR(150) NOT NULL,
27.     email VARCHAR(50) NOT NULL,
28.     PRIMARY KEY (id)
29. );
30. CREATE TABLE users_to_delete (
31.     username VARCHAR(50) NOT NULL
32. );
33. CREATE TABLE users_to_change (
34.     username VARCHAR(50) NOT NULL
35. );

```

Esto nos creara 5 tablas:

1. **Unverified_users:** Tabla en la cual guardaremos los usuarios que se han registrado en el sistema, pero **no** han verificado su correo electrónico/usuario. El campo que destacar es **ver_code** el cual utilizamos para asignar al usuario un código de verificación.
2. **Verified_users:** Tabla en la cual guardaremos los usuarios que han confirmado su correo electrónico/usuario pero que aún no se ha procesado la creación del usuario en el servidor.
3. **Final_users:** Tabla en la cual guardaremos los usuarios finales que tienen acceso al servidor y a su página principal.

4. **Users_to_delete:** Tabla en la cual guardaremos el nombre de los usuarios que han solicitado que se borre su usuario.
5. **Users_to_change:** Tabla en la cual guardaremos el nombre de los usuarios que han solicitado cambiar su contraseña.

Creación de Blogs

Para la creación de blogs hemos usado Wordpress ya que es la opción más popular y dispone de una gran comunidad detrás que crea y mantiene plugins que añaden interesantes funciones a este gestor de contenidos, para instalarlo debemos seguir los siguientes pasos:

Primero lo descargamos desde su página principal con la siguiente orden:

```
1. Wget https://es.wordpress.org/wordpress-5.2.1-es_ES.tar.gz
```

Lo descomprimos con la siguiente orden:

```
1. Tar xzvf wordpress-5.2.1-es_ES.tar.gz
```

Lo movemos a la carpeta de apache con la siguiente orden:

```
1. Mv /home/usuario/wordpress /var/www/html/
```

Cambiamos el propietario de la carpeta para que Apache pueda acceder y manipular el contenido cuando lo necesite:

```
1. chown -R www-data:www-data /var/www/html/wordpress/
```

Al ir a la dirección `http://tudominio.com/wordpress/` nos saldrá lo siguiente:

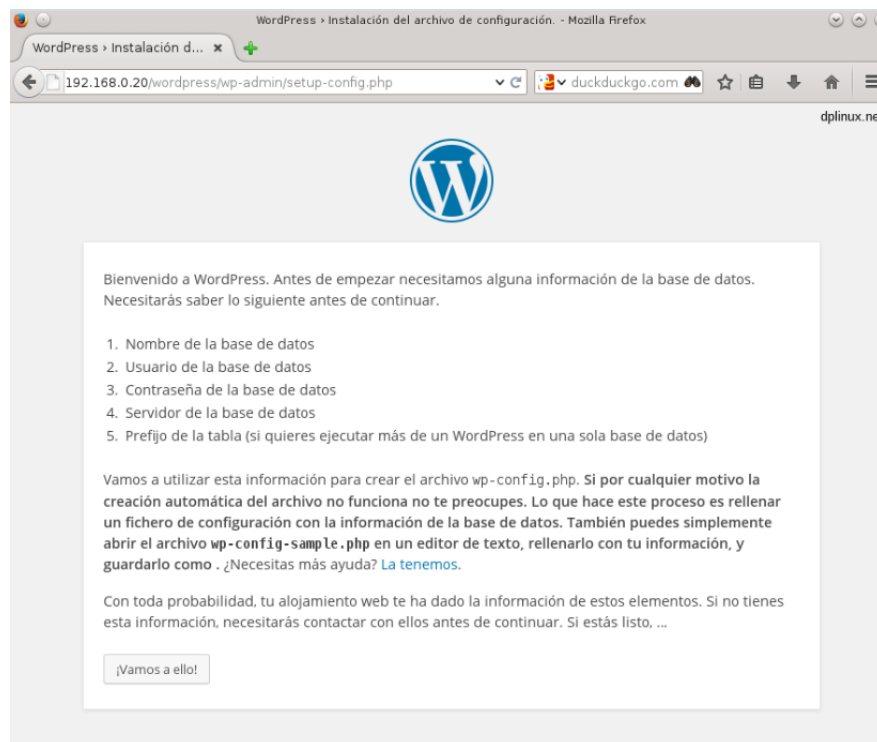


Ilustración 51: wordpress bienvenida

Antes de continuar debemos crear una base de datos, para ello escribimos:

```
1. mariadb -u admin -p
2. create database wordpress;
3. exit;
```

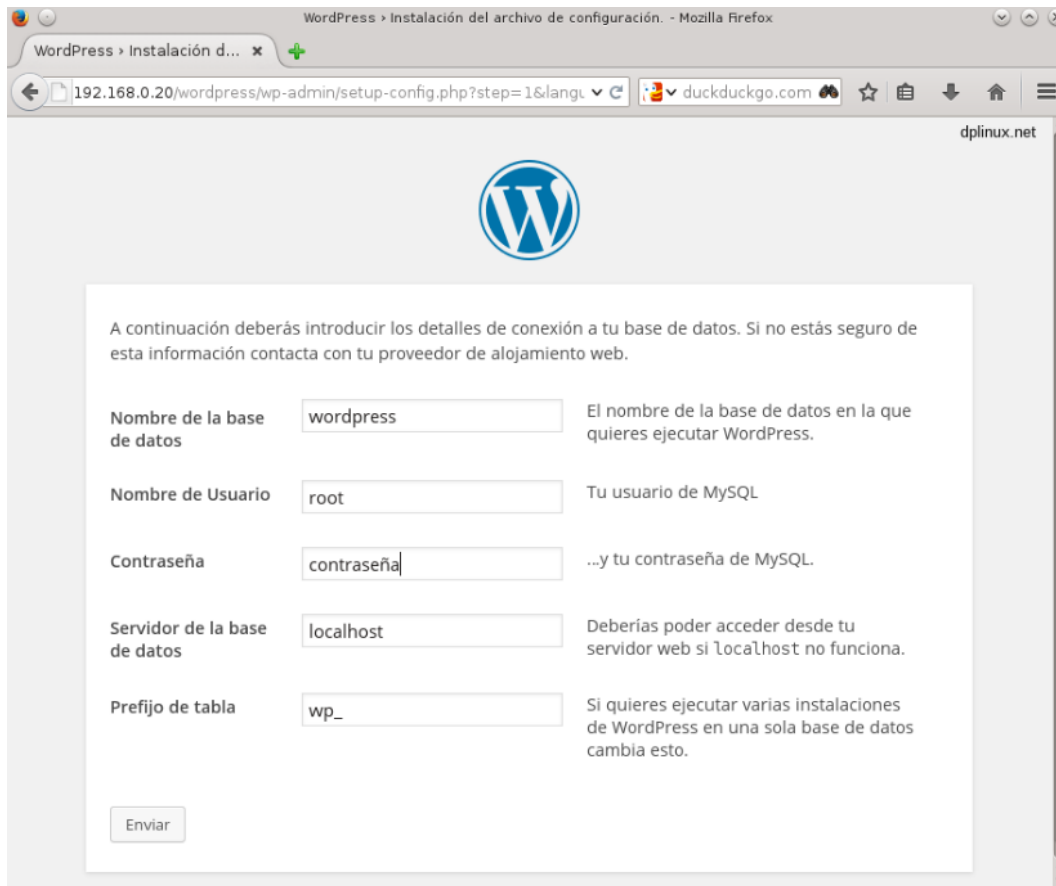


Ilustración 52: wordpress config

Introducimos el nombre de la base de datos que acabamos de crear y el nombre de usuario y contraseña del usuario con todos los permisos.

Cacti

Cacti es un visualizador de la actividad de nuestro servidor a través de una interfaz web que permite ver las estadísticas de uso en distintos periodos.

Para instalarlo tan solo debemos realizar la siguiente orden:

1. `apt-get install cacti`

Una vez instalado si vamos a la dirección <http://tudominio.com/cacti/> podremos iniciar sesión y monitorizar nuestro servidor.



Ilustración 53: Ejemplo cacti

Servidor de correo electrónico (Roundcube)

Para instalar Roundcube el primer paso que debemos seguir es escribir lo siguiente:

1. `apt-get install php5 php5-mysql postfix apache2 dovecot-imapd dovecot-pop3d mysql-server mysql-client roundcube`

Una vez instalados todos los paquetes, roundcube nos pedirá un usuario y contraseña, estos datos son los del usuario administrador en nuestra base de datos.

Editamos el fichero de configuración que se encuentra en `/etc/apache2/conf-available/roundcube.conf` y descomentamos la línea de `Alias`:

```
# Those aliases do not work properly with some versions of mod_alias
# Uncomment them to use it or adapt them to your setup
Alias /roundcube /var/lib/roundcube
```

Ilustración 54: Configuración roundcube

Después reiniciamos el servicio de apache con la orden:

```
1. systemctl restart apache2
```

Debemos crear un enlace simbólico con una nueva ruta para roundcube dentro de webmail:

```
1. Ln -s /var/lib/roundcube/ /var/www/html/webmail
```

Una vez hecho esto si entramos en la dirección <http://midominio.com/roundcube> podremos iniciar sesión tanto poniendo en la parte de servidor *localhost* como la ip del servidor.

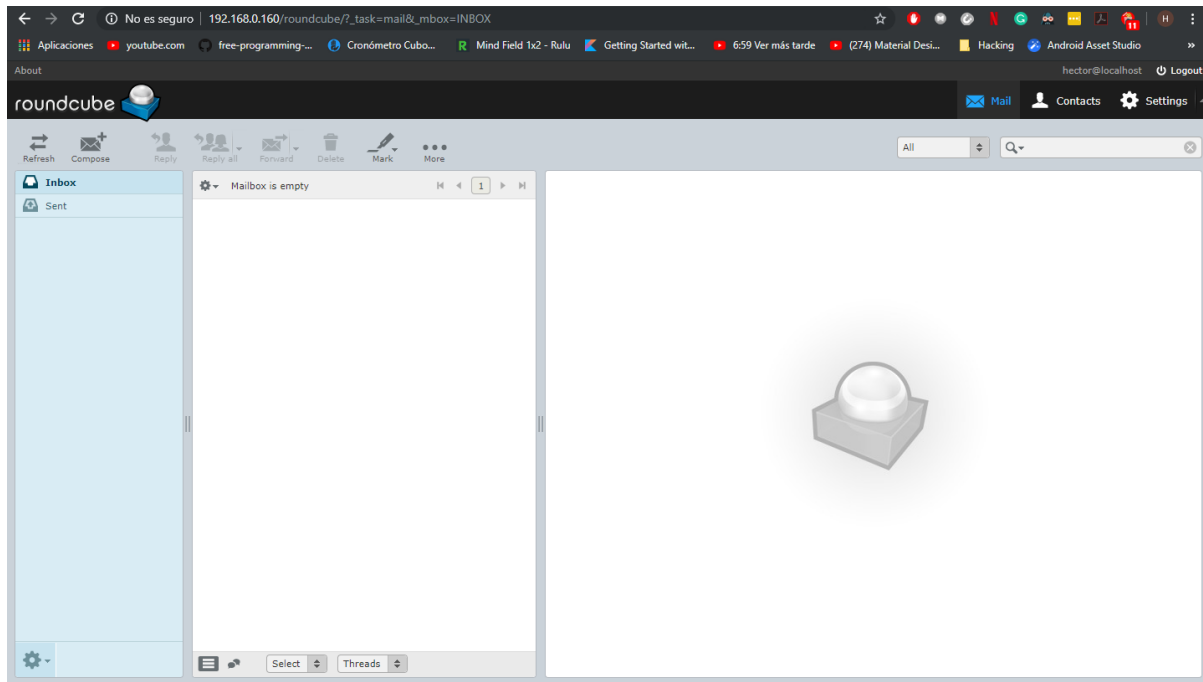


Ilustración 55: Login de roundcube

Servidor FTP

Vamos a implementar 2 métodos, uno para la bajada de archivos mediante *proftpd* y otro mediante *filezilla*.

Vamos a ver como instalar el primero de ellos:

Lo primero que debemos hacer es instalar *proftpd* y lo hacemos con la siguiente orden:

1. `apt-get install proftpd`

Configuramos el archivo *proftpd.conf* que se encuentra en */etc/proftpd* y añadir las siguientes líneas al final del archivo:

1. `<Global>`
2. `RootLogin off`
3. `RequireValidShell off`
4. `</Global>`

Nos debe quedar algo así:

```
# <Directory *>
#   <Limit WRITE>
#       DenyAll
#   </Limit>
# </Directory>
#
# # Uncomment this if you're brave.
# # <Directory incoming>
# #   # Umask 022 is a good standard umask to
# #   # (second parm) from being group and world
# #   Umask                                022 0
# #
#       <Limit READ WRITE>
#       DenyAll
#       </Limit>
#       <Limit STOR>
#       AllowAll
#       </Limit>
# # </Directory>
#
# </Anonymous>

# Include other custom configuration files
Include /etc/proftpd/conf.d/

<Global>
    RootLogin off
    RequireValidShell off
</global>
```

Ilustración 56: *proftpd.conf*

Si accedemos al servidor de la siguiente forma: *ftp://localhost/* nos pedirá un usuario y contraseña para poder descargar los archivos de ese usuario.

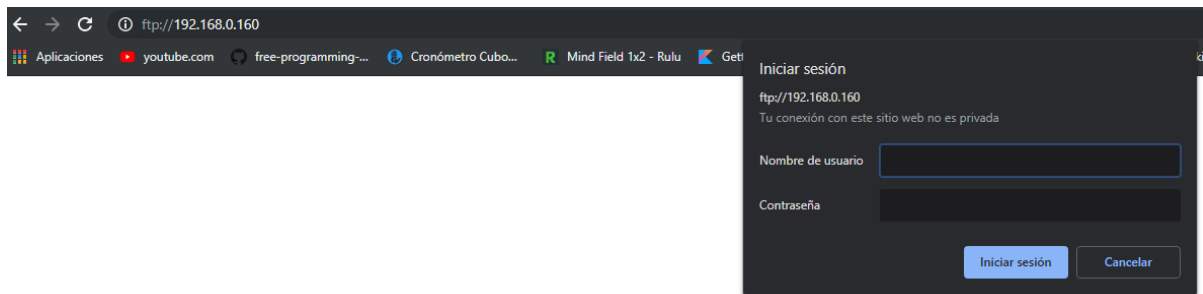
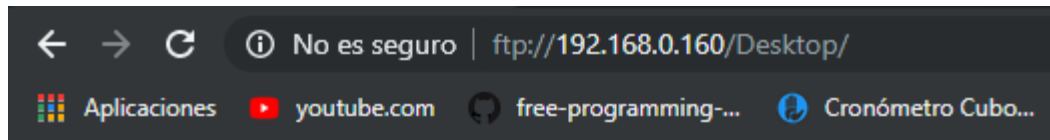



Ilustración 57: login proftpd



Índice de /Desktop/

 [directorio principal]










	Nombre	Tamaño	Fecha de modificación
	exit.cgi	334 B	16/5/19 16:30:00
	index.html	819 B	18/5/19 22:03:00
	logs.pl	274 B	24/5/19 1:09:00
	pruebas/		23/5/19 12:34:00
	pruebas.pl	169 B	22/5/19 23:53:00
	sesion.txt	15 B	24/5/19 0:58:00
	startup_linux_server/		23/5/19 23:31:00
	wordpress-4.8.1-es_ES.tar.gz	8.3 MB	18/9/17 2:00:00
	wordpress-5.2.1-es_ES.tar.gz	11.3 MB	22/5/19 19:00:00

Ilustración 58: proftps escritorio usuario

Explicación scripts CGI

En este apartado vamos a explicar el funcionamiento de todos los scripts que tenemos instalados en nuestro servidor. No entraremos en detalles de programación ya que de lo contrario esta memoria se haría extremadamente extensa por lo que explicaremos el funcionamiento de cada uno y destacaremos aspectos importantes.

Todos estos scripts se han introducido en la carpeta `/var/www/html/cgi-enabled`

- **Change_password.pl:** Este script se ejecuta cuando el usuario, escribe una nueva contraseña. Este script se encarga de validar que el usuario tenga la contraseña que realmente dice, en el caso de que sea su contraseña, le borra de la base de datos de usuarios finales, lo modifica y lo vuelve a meter en la base de datos con su contraseña actualizada. También se encarga de meter el nombre del usuario en la base de datos de `users_to_change` para que el script que se ejecuta cada minuto sepa que alguien quiera cambiar su contraseña, este script es el encargado de actualizar el usuario en Linux
- **Confirm.pl:** Este script se encarga de verificar a los usuarios que hacen click en el link de verificación que se le manda por correo. Obtiene los parámetros de la URL y busca al usuario en la base de datos de usuarios no verificados, si ese usuario se encuentra en allí y su código coincide lo borra de la tabla de no verificados y lo introduce en la de verificados.
- **Delete_user.pl:** Este script se encarga de que cuando un usuario quiere borrar su cuenta, lo elimina de la base de datos, cierra su sesión e introduce su nombre de usuario en la tabla de usuarios a borrar para que el script que se ejecuta cada minuto sepa que tiene que eliminarlo.
- **Logout.pl:** Este script se ejecuta cuando un usuario quiere cerrar sesión.
- **Forgot_password.pl:** Cuando un usuario olvida su contraseña se le envía una nueva por correo. Primero se genera una contraseña aleatoria y se le envía al usuario.
- **Login.pl:** Este script recoge los datos del formulario y busca al usuario en la base de datos de usuarios finales, en caso de encontrarle, crea una sesión y lo redirige a su perfil, en el caso de no encontrarle le busca en las distintas tablas para mostrarle un mensaje informativo de por qué el acceso es erróneo (por ejemplo que el usuario este siendo verificado)
- **Private.pl:** Se asegura de que hay una sesión activa, en el caso de que la haya le redirecciona a su perfil y en el caso de que no la haya le redirecciona a login.
- **Register.pl:** Script que recoge los datos del usuario introducidos en el formulario, crea un usuario con una contraseña aleatoria y lo introduce en la base de datos de usuarios sin verificar. También le manda un correo al usuario con el link de confirmación y su contraseña.

Problemas encontrados

Hemos encontrado un gran número de problemas en prácticamente todos los aspectos de la práctica que hacen que sea muy difícil explicarlos todos.

La gran mayoría acababan siendo problemas de:

- Permisos
- Falta de paquetes
- Falta de configuración
- Incompatibilidades

La gran mayoría hemos logrado solucionarlos pero hay 2 problemas que no hemos podido solventar y son los siguientes:

- **Tripwire:** Al instalar tripwire en un servidor este recoge la ip en los archivos de su configuración por lo que, en nuestro caso al cambiar el servidor de ordenador (Cosa que no es nada habitual en la vida real) tripwire deja de funcionar y hay que solucionarlo a mano reiniciando una configuración
- **Wordpress:** Ocurre lo mismo que con Tripwire pero son una gran cantidad de archivos para renombrar por lo que se optara por la instalación el día de la defensa para poder observar sus funcionalidades.

Posibles mejoras

Hemos implementado en el servidor una gran número de funcionalidades pero aun nos han faltado algunas que si hubiéramos tenido más tiempo las hubiéramos implementado, estas mejoras son:

- Implementar un servidor DNS para poder resolver el nombre de dominio en la red de área local o incluso conseguir una ip publica de algún proveedor para poder hacer referencia al servidor desde fuera o dentro del área local.
- Mejorar la interfaz del perfil del usuario.
- Implementar el chat empresarial.
- Implementar OwnCloud.
- Usar otro tipo de comunicación entre el servidor y los ficheros de código.
- Enviar mail al usuario cuando exceda el limite soft.

Conclusiones

Esta es una de las practicas más completas y complicadas (en cuanto a quebraderos de cabeza) nos hemos encontrado, al menos a nuestro parecer, pero es una práctica que te hace aprender mucho sobre la administración de sistemas y de cómo configurar nuestros propios servidores.

Nos hubiera gustado algo más de libertad para la elección de lenguajes, o métodos de creación de servidores pero dado que está enfocado a administrar y no a crear un servidor eficiente y simple no ha sido posible.

Referencias

- <https://ubuntuforums.org/showthread.php?t=2258746>
- https://medium.com/@manivannan_data/python-cgi-example-install-and-simple-example-59e049128406
- https://hardlimit.com/guia-servidor-en-debian/#_RefHeading_124_1337659763
- <https://stackoverflow.com/questions/23293589/perl-module-install-error-cpan>
- <https://gist.github.com/Hikingyo/549698845d166e49eb3d238d03d49236>
- <http://www.leonardoborda.com/blog/how-to-add-a-new-partition-to-the-fstab-file/>
- https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/5/html/deployment_guide/ch-disk-quotas
- <https://www.golinuxhub.com/2018/08/step-by-step-guide-implement-quota-edquota-grace-period-linux.html>
- <https://informaticocurioso.wordpress.com/2016/11/15/instalacion-roundcube/>
- <https://websiteforstudents.com/install-apache2-php-phpmyadmin-ubuntu-17-04-17-10/>
- <https://stackoverflow.com/questions/23293589/perl-module-install-error-cpan>
- <https://stackoverflow.com/questions/17572951/unable-to-find-mysql-config-when-installed-dbdmysql-on-amazon-ec2>
- <https://www.perl.com/article/43/2013/10/11/How-to-schedule-Perl-scripts-using-cron/>
- <https://www.osi.es/es/actualidad/blog/2018/08/21/que-son-los-ataques-dos-y-ddos>
- https://www-solvetic-com.cdn.ampproject.org/v/s/www.solvetic.com/tutoriales/article/4437-instalar-usar-tripwire-detectar-archivos-modificados-ubuntu-17/?ampmode=1&usqp=mq331AQFCAGgAQA%3D&_js_v=0.1#referrer=https%3A%2F%2Fwww.google.com&_tf=De%20%251%24s&share=https%3A%2F%2Fwww.solvetic.com%2Ftutoriales%2Farticle%2F4437-instalar-usar-tripwire-detectar-archivos-modificados-ubuntu-17%2F
- <https://hardlimit.com/guia-servidor-en-debian/>