Apache 2

- Para instalar el Apache2

```
apt-get install apache2
```

- Para arrancar el apache

```
/etc/init.d/apache2 start
```

- Para recargar la configuración del apache

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

- Para ver el estado del servidor apache

```
apache2ctl status
```

Los archivos de configuración se encuentran en /etc/apache2/

Restricción de acceso

Podremos restringir el acceso de dos maneras: Por usuario y por IP

Para restringir por usuario, añadimos lo siguiente a la configuración del apache

```
<Location /url>
AuthType Basic
AuthName "Paco"
AuthUserFile /ruta/fichero.pass
Require valid-user
</Location>
```

De esa manera cerramos el acceso a http://IP/url y todo lo que haya por debajo. El /ruta/fichero.pass es un fichero donde estarán los usuarios válidos. Lo podemos crear manual o con una pequeña herramienta que trae apache, el htpasswd.

Para hacerlo a mano, sólo hay que tener en cuenta que el formato que sigue el fichero es nombreusuario:contraseña. La contraseña se puede obtener usando mkpasswd

Con el htpasswd, bastaría con htpasswd -mb /ruta/fichero.pass usuario contraseña

Para restringir por IPs, usaremos 3 directivas: Order, Deny y Allow. Por ejemplo, para cerrar el acceso a todos y que sólo entren los de la subred 10.1.2.0/24, añadiríamos

```
Order Allow,Deny
```

```
Allow from 10.1.2.0/24
Deny from all
```

Tanto las restricciones por IP como las de usuarios tienen que ir dentro de un <Location>, <Directory> o en un fichero .htaccess

Usar conexiones seguras con SSL

- Habilitamos el módulo ssl:

```
a2enmod ssl
```

- Ejecutamos un script para crear nuestro certificado de seguridad para el servidor (estará autofirmado).

```
apache2-ssl-certificate
```

Nos hará una serie de preguntas...

```
apache2-ssl-certificate
```

```
creating selfsigned certificate
replace it with one signed by a certification authority (CA)
enter your ServerName at the Common Name prompt
If you want your certificate to expire after x days call this programm
with -days x

Generating a 1024 bit RSA private key
........++++++
.....................++++++

writing new private key to '/etc/apache2/ssl/apache.pem'
-----
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) [GB]:ES
State or Province Name (full name) [Some-State]:Canarias
Locality Name (eg, city) []:Las Palmas de Gran Canaria
Organization Name (eg, company; recommended) []:mi empresa
Organizational Unit Name (eg, section) []:mi empresa
server name (eg. ssl.domain.tld; required!!!) []:apache.miempresa.com
Email Address []:micorreo@miempresa.com
```

- Dependiendo de nuestra configuración podemos hacer que la comunicación de nuestro sitio sea

por defecto bajo SSL o crear otro sitio

Si lo queremos por defecto la configuración sería de la siguiente forma:

Puesto que se va a usar para una aplicación, lo mejor (tanto lo más seguro como lo más cómodo) es forzar a que todas las conexiones vayan por https. Así, además, nos ahorramos tener dos configuraciones de lo mismo (una para http y otra para https). Para hacer que la configuración actual vaya por https, tocaremos el sites-available/default

1. Pondremos la línea NameVirtualHost *:443 al principio
2. Dejamos la linea de VirtualHost como <VirtualHost *:443>
3. Añadimos las siguientes líneas, por ejemplo debajo de Serversignature on

```
SSLEngine On

SSLCertificateFile /etc/apache2/ssl/apache.pem
```

Si no lo queremos por defecto, crearemos la configuración de "el sitio" para el servidor seguro basándonos en la que lleva por defecto:

```
cp /etc/apache2/sites-available/default /etc/apache2/sites-available/ssl
```

```
ln -s /etc/apache2/sites-available/ssl /etc/apache2/sites-enabled/ssl
```

/etc/apache2/sites-enabled/ssl tiene que empezar de la siguiente manera:

```
NameVirtualHost *:443

<VirtualHost *:443>

ServerAdmin webmaster@localhost

DocumentRoot /var/www/ssl.miempresa.net/htdocs

<Directory />

Options FollowSymLinks

AllowOverride None

</Directory>

<Directory /var/www/ssl.miempresa.net/htdocs>
```

En /etc/apache2/sites-enabled/default también hay que configurarlo de la misma forma:

```
NameVirtualHost *:80

<VirtualHost *:80>

ServerAdmin webmaster@localhost
```

```
DocumentRoot /var/www/miempresa.net/htdocs

<Directory />

Options FollowSymLinks

AllowOverride None

</Directory>

<Directory /var/www/miempresa.net/htdocs>
```

Por último, sólo basta añadir dentro del fichero "/etc/apache2/sites-enabled/ssl" en cualquier lugar (por ejemplo justo debajo de "ServerSignature On"):

```
SSLEngine On

SSLCertificateFile /etc/apache2/ssl/apache.pem
```

- Después hay que indicarle al apache que escuche peticiones por https y use ese certificado.

Para ello añadir al fichero /etc/apache2/ports.conf la línea:

```
Listen 443
```

- Hay que asegurarse también de que el módulo para SSL está cargado. Para ello, miramos en el directorio /etc/apache2/mods-enabled si existen los ficheros ssl.conf y ssl.load. Si no están, ejecutamos

```
cd mods-enabled

ln -s ../mods-available/ssl.* .
```

- Y por último, reiniciamos apache2:

```
/etc/init.d/apache2 force-reload
```

Configuración por directorios con .htaccess

Podemos colocar la configuración a los dentro del propio directorio, a través de los ficheros .htaccess.

Primero hay que colocar un AllowOverride en el <Directory> donde queremos poder cambiar su configuración con los .htaccess. El .htaccess se aplicará además a todos los subdirectorios.

El AllowOverride define qué parámetros de configuración se pueden modificar desde los .htaccess. Lo más habitual será ponerle un All para poder cambiar cualquier cosa o un None para que no se pueda cambiar nada.

```
<Directory /directorio/que/quermos/cambiar>

```

```
AllowOverride All

...

</Directory>
```

Dentro de los .htaccess podremos poner casi cualquier directiva que se puede poner en un <Directory>. En la documentación de las directivas siempre aparece una pequeña cabecera con cierta información de cada directiva, y hay una («Context») que dice si esa directiva puede ir en un .htaccess o no.

Referencias

Blog de jHernandez http://jhernandez.gpltarragona.org/blog/?p=239