Formación en Red

Ministerio de Educación, Cultura y Deporte

Aulas en red. Aplicaciones y Servicios. Linux

Servidor web Apache

Instituto Nacional de Tecnologías Educativas y
de Formación del Profesorado
2012

Introducción a Apache

El servidor web apache es una de las aplicaciones estrella del mundo Linux. Es el servidor web más implantado entre los distintos servidores que ofertan servicios web en Internet.

Entre las características más significativas destacamos:

- Es modular
- Permite crear servidores virtuales
- Permite crear servidores seguros https
- Permite crear sitios privados
- Permite crear sitios de usuario

En este curso haremos uso de éstas y otras características de apache.

Pregunta Verdadero-Falso

Si deseamos montar un servidor web en Linux, solo podemos utilizar Apache.

Verdadero ○ Falso ○

Organización del sitio web

La organización que realizaremos de nuestro servidor Apache, será la clásica en los sistemas Unix: **la página web de la intranet** se almacenará en la carpeta raíz del servidor web, las **páginas de los usuarios** se almacenarán en la carpeta home de cada usuario y para albergar las **páginas web de los distintos departamentos** didácticos del centro, lo más práctico es crear nuevos usuarios con el nombre del departamento.

Espacio web para la Intranet

Por defecto, la carpeta raiz del servidor web es la carpeta /var/www. Todos los documentos que se encuentren dentro de la carpeta raíz del servidor web, serán accesibles vía web. Dentro del raiz de documentos crearemos la **página web de nuestra intranet:**

- **Carpeta raíz del servidor web:** /var/www
- **Acceso a la web principal:** http://ip-del-servidor ó http://nombre-del-servidor

Para acceder vía web a la página almacenada en la carpeta raíz del servidor, desde un navegador debemos acceder directamente con la dirección IP a: http://ip-del-servidor o bien utilizando el nombre del mismo si tenemos el DNS funcionando: http://nombre-del-servidor. Si no tenemos el DNS funcionando, podemos añadir el nombre y la IP en /etc/hosts para resolver localmente.

Espacio web para cada usuario

Cada usuario del sistema dispondrá de un espacio web que se almacena dentro de su carpeta home en una carpeta llamada 'public_html'. Si dicha carpeta no existe, el propio usuario puede crearla y copiar dentro de ella su página web. Los permisos recomendados son 755 para que el 'grupo' y el 'resto' de usuarios tengan acceso de lectura y así se puedan visualizar las páginas.

Para acceder vía web a la página de un usuario, desde un navegador debemos acceder directamente con la dirección IP a: http://ip-del-servidor/~login-usuario/

El caracter '~'comúnmente conocido como gusanillo y que se obtiene con Alt Gr + 4 sirve para indicar a apache que debe servir la página desde el home del usuario (en Linux el 'gusanillo' equivale a la carpeta home). Ejemplo, si hemos creado un usuario javier y éste ha creado la carpeta /home/javier/public_html y ha copiado en ella su página web, desde cualquier PC de la red podremos acceder a dicha carpeta yendo a la dirección http://ip-del-servidor/~javier/. Para que la página aparezca automáticamente, es necesario crear un archivo llamado index.html.

- **Carpeta web de javier:** /home/javier/public_html
- **Acceso a la web de javier:** http://ip-del-servidor/~javier/

Espacio web para los departamentos

Para proporcionar espacio web a los departamentos, lo más sencillo es crear un usuario para cada departamento. Podemos crear los usuarios: matematicas, lengua, ingles, plastica (sin acentos), etc... Al igual que cada usuario del sistema, dispondrán de un espacio web dentro de su carpeta home en una carpeta llamada 'public_html'. Si dicha carpeta no existe, habrá que crearla y copiar dentro de ella la página web del departamento.

Para acceder vía web a la página del departamento, desde un navegador debemos acceder directamente con la dirección IP a: http://ip-del-servidor/~departamento. Ejemplo, si hemos creado un usuario matematicas y hemos creado la carpeta /home/matematicas/public_html y copiado en ella la web del departamento de matemáticas, desde cualquier PC de la red podremos acceder a dicha web yendo a la dirección http://ip-del-servidor/~matematicas. Para que la página aparezca automáticamente, es necesario crear un archivo llamado index.html.

- **Carpeta web del dpto. de matemáticas:** /home/matematicas/public_html
- **Acceso a la web de dpto. de matemáticas:** http://ip-del-servidor/~matematicas/

De la misma manera, se pueden crear usuarios para proporcionar espacio web a otros órganos del centro, p.ej: ccp, orientacion, equipodirectivo, conserjeria, etc... para que dispongan de su propio espacio web.

Espacio web seguro

Además crearemos un sitio web virtual seguro en el servidor web Apache para poder tener acceso vía SSL a contenidos que deseamos que sean seguros, es decir, accesibles en el navegador mediante el protocolo "https", será la carpeta /var/www /websegura:

- **Carpeta web segura:** /var/www/websegura
- **Acceso a la web segura:** https://ip-del-servidor/websegura/

Dentro de esta estructura la mayoría de los contenidos serán públicos y cualquier usuario podrá acceder a ellos. Sin embargo, algunas de las carpetas serán privadas y solo se tendrá acceso a ellas identificándose con nombre de usuario y contraseña.

Raiz de documentos del servidor web

Por defecto, la carpeta raiz de documentos (DocumentRoot) de Apache es:

- /home/www
- /root/www
- /var/www

Instalación y configuración de Apache

Instalación de Apache2

Disponer de un servidor web en el centro nos permitirá alojar nuestras propias páginas y aplicaciones web de forma que den servicio tanto desde dentro de la intranet como desde Internet. Serán la base que facilitará el acceso a la información por parte de la comunidad educativa.

```
// Instalación de apache2
# apt-get install apache2
```

Con lo cual se instalarán los archivos necesarios para que funcione nuestro servidor web. Se instalará apache v2.

Configuración de Apache

Los archivos de configuración de apache2 se encuentran en la carpeta /**etc/apache2**. El archivo principal de configuración es **/etc/apache2/apache2.conf**. Antes de realizar cualquier cambio en este archivo, es conveniente realizar una copia de seguridad del mismo ya que si apache encuentra algún error en el archivo de configuración, no arrancará.

Se pueden configurar infinidad de parámetros. Aquí, para poner en marcha el servidor, editaremos el archivo apache2.conf y añadiremos únicamente el siguiente parámetro:

```
// Añadir en apache2.conf
ServerName www.ieslapaloma.com
```

Para que los PCs de la red local sepan que www.ieslapaloma.com es nuestro servidor web, debemos crear una entrada 'www' hacia su dirección IP en el servidor DNS, o bien editar el archivo /etc/hosts agregando la línea: '192.168.1.239 www.ieslapaloma.com' (si la IP del servidor fuera 192.168.1.239). Si no, no quedará más remedio que acceder utilizando la dirección IP del servidor.

Arranque y parada del servidor web apache

El servidor web apache2, al igual que todos los servicios en Debian, dispone de un script de arranque y parada en la carpeta /etc/init.d.

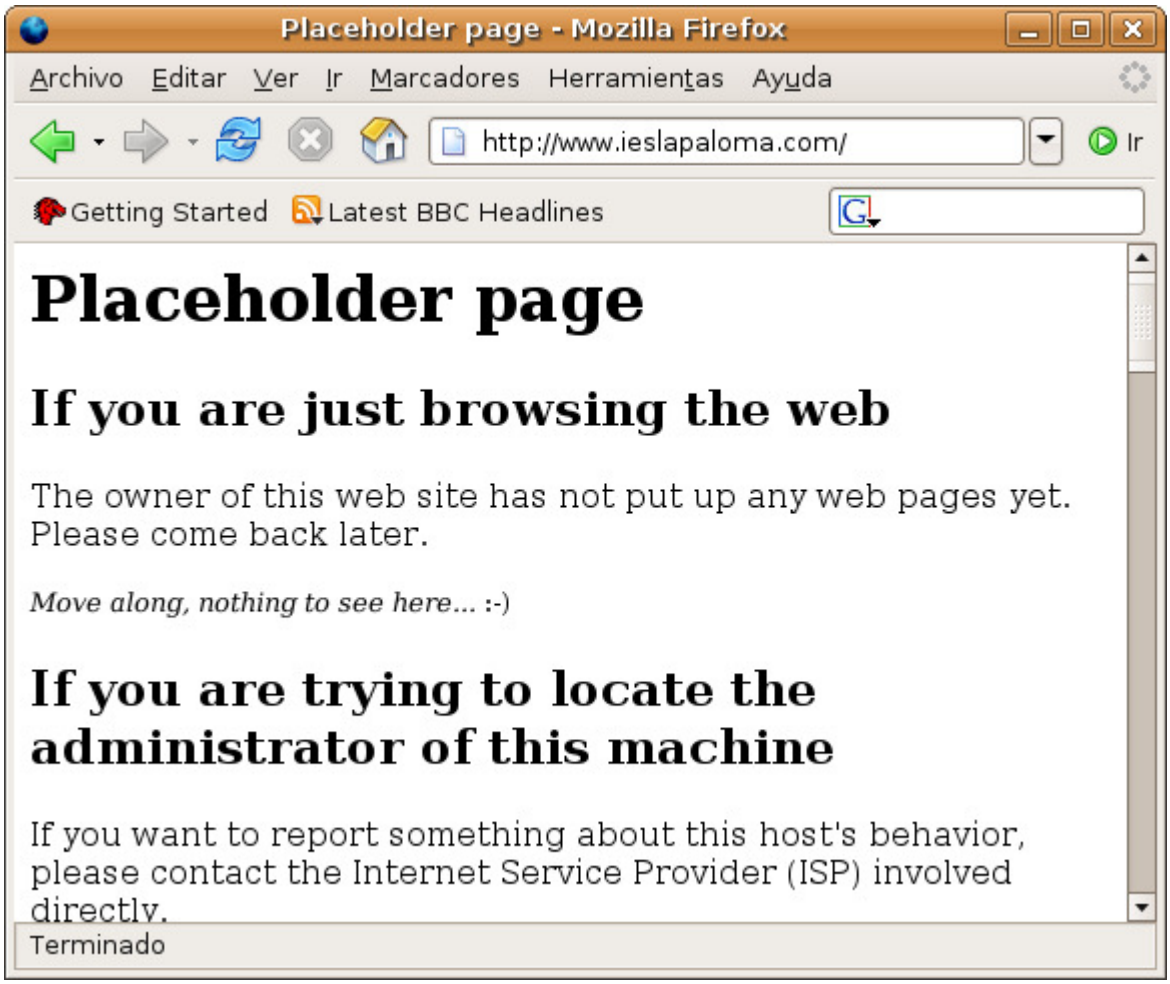
```
// Arrancar o reiniciar el servidor apache2
# /etc/init.d/apache2 restart
```

```
// Parar el servidor apache
root@cnice-desktop:/# /etc/init.d/apache stop
```

Arranque automático del servidor Web Apache al iniciar el sistema

Para un arranque automático del servicio al iniciar el servidor, debemos crear los enlaces simbólicos correspondientes tal y como se indica en el apartado **Trucos > Arranque automático de servicios al iniciar el sistema.**

Para comprobar que apache funciona perfectamente, desde el navegador de cualquier estación de trabajo de nuestro centro, debemos dirigirnos a 'http://ip-del-servidor'. Si tenemos el DNS funcionando, podemos acceder a 'http://www.ieslapaloma.com', visualizando la siguiente pantalla:



Servidor Web Apache funcionando perfectamente

Si no disponemos de servidor DNS, podemos editar el archivo /etc/hosts y añadir la dirección IP del servidor e indicar el nombre, tal que así:

```
//Resolver nombres de dominio de forma local
//Añadir en /etc/hosts una línea similar a esta:
192.168.1.239 www.ieslapaloma.com
```

Lo que siempre funcionará es ir con la dirección IP. Ejemplo, si la dirección IP de nuestro servidor fuera 192.168.1.239, podemos ir con el navegador a la dirección http://192.168.1.239 y obtendremos el mismo resultado. Podemos personalizar nuestra página modificando el archivo index.html que hay dentro de la carpeta /var/www.

Como vemos en la pantalla anterior, la instalación de Apache se produjo de forma adecuada, así pues hemos completado este apartado satisfactoriamente.

Espacio web de los usuarios

Cada usuario dispone de un espacio web que se almacena en la carpeta public_html dentro de su carpeta home. Si la carpeta public_html no existe, el propio usuario la puede crear y almacenar en ella su sitio web. La carpeta public_html deberá tener permisos 755 para que el 'grupo' y el 'resto' de usuarios tengan acceso de lectura y así se puedan visualizar las páginas.. Si queremos que la carpeta public_html se genere de forma automática al dar de alta al usuario, se puede crear en /etc/skel. Para que apache procese los espacios web de los usuarios, es necesario activar el módulo **userdir** mediante el siguiente comando:

```
// Activar el espacio web de los usuarios
a2enmod userdir

// Reinicar apache para poder comenzar a utilizar el espacio web de usuarios recien activado
/etc/init.d/apache2 restart
```

Para acceder vía web a la página de un usuario, desde un navegador debemos acceder directamente con la dirección IP a: http://ip-del-servidor/~login-usuario/

El caracter '~'comúnmente conocido como gusanillo y que se obtiene con Alt Gr + 4 sirve para indicar a apache que debe servir la página desde el home del usuario (en Linux el 'gusanillo' equivale a la carpeta home). Ejemplo, si hemos creado un usuario javier y éste ha creado la carpeta /home/javier/public_html y ha copiado en ella su página web, desde cualquier PC de la red podremos acceder a dicha carpeta yendo a la dirección http://ip-del-servidor/~javier/. Para que la página aparezca automáticamente, es necesario crear un archivo llamado index.html.

- **Carpeta web de javier:** /home/javier/public_html
- **Acceso a la web de javier:** http://ip-del-servidor/~javier/

Espacio web de los usuarios

Por defecto en Apache, el espacio web de los usuarios se encuentra:

- ○ Dentro de la carpeta home de cada usuario
- ○ Dentro del DocumentRoot (raíz de documentos) de Apache

Apache + PHP + MySQL

Para poder aprovechar al máximo las características del servidor web apache, es muy conveniente que pueda ejecutar scripts en servidor con un lenguaje de programación como PHP y pueda acceder a un sistema gestor de bases de datos como MySQL. A un sistema Linux con Apache, PHP y MySQL se le conoce comunmente como un sistema LAMP: Linux + Apache +Mysql + Php.

Las aplicaciones web más utilizadas actualmente en Internet como los gestores de contenidos para crear y mantener sitios web dinámicos, wikis, blogs, foros-web, repositorios de arhivos, etc., utilizan PHP y MySQL.

En el desarrollo web del mundo Linux, el lenguaje script en servidor más utilizado es el lenguaje PHP y el sistema gestor de bases de datos más utilizado es MySQL. Phpmyadmin es una excelente herramienta escrita en PHP, para administrar bases de datos MySQL.

Más información sobre cómo instalar y configurar PHP, MySQL y Phpmyadmin en el capítulo **Otros servicios**

- Otros servicios > Instalación y configuración de PHP
- Otros servicios > Instalación y configuración de MySQL
- Otros servicios > Instalación y configuración de PHPMyAdmin

Pregunta Verdadero-Falso

Un gran número de aplicaciones web necesitan que el servidor web disponga de PHP y MySQL para funcionar

Verdadero ○ Falso ○

Acceso a carpetas privadas

Otra posibilidad es que los profesores e incluso el sitio web de la Intranet de nuestro centro, pueda disponer de carpetas privadas accesibles mediante el navegador pero no por cualquier usuario. Por ejemplo, los profesores podrían disponer de una carpeta donde almacenar información confidencial accesible desde la web. Así mismo puede ocurrir que queramos tener en el servidor web de nuestra intranet, páginas a las que sólo puedan tener acceso de lectura los profesores del centro. El acceso sería mediante un nombre de usuario y una contraseña almacenando los usuarios en una base de datos LDAP como se verá más adelante.

Con el auge de las plataformas web y los gestores de contenido, cada vez se utilizan menos las carpetas privadas del servidor web ya que este tipo de herramientas web hacen una gestión de permisos de usuarios mucho más avanzada.

Acceso a carpetas seguras

Introducción

Una página web segura o un sitio web seguro es un sitio web que utiliza el protocolo https en lugar de utilizar el protocolo http.

El protocolo https es idéntico al protocolo http con la excepción de que la transferencia de información entre el cliente (navegador web) y el servidor (servidor web) viaja a través de Internet cifrada utilizando robustos algoritmos de cifrado de datos proporcionados por el paquete OpenSSL.

Los algoritmos de cifrado utilizados reunen las características necesarias para garantizar que la información que sale desde el servidor hacia el cliente, esté cifrada y solamente pueda ser descifrada por el cliente y que la información que sale desde el cliente hacia el servidor, esté cifrada y solamente pueda ser descifrada por el servidor. Si durante la transferencia de la información un 'hacker' hiciera copia de los paquetes de datos e intentara descifrarlos, los algoritmos garantizarían que no podría hacerlo por fuerza bruta (probando todas las claves posibles) en un plazo mínimo de varios años.

Durante la transmisión, se utilizan algotirmos de cifrado simétricos, pero para intercambiar las claves de cifrado, hay una sesión inicial de cifrado asimétrico.

Entidad Certificadora

Una entidad certificadora (en inglés CA Certification Authority) es alguien que puede firmar certificados de usuarios y garantizar su autenticidad. Por ejemplo en España, una entidad certificadora es la FNMT - Fábrica Nacional de Moneda y timbre http://www.cert.fnmt.es

Los certificados permiten identificar y autentificar a sus titulares (usuarios, equipos, servidores,...), siempre y cuando estén firmados por una CA de confianza. Ejemplo, el usuario Pepe puede tener un certificado firmado por la FNMT que le sirve para autentificarse en la Agencia Tributaria. La Agencia Tributaria le permitirá el acceso ya que confía en los certificados firmados por la FNMT.

Si confiamos en una CA, confiamos en sus certificados

Si confiamos en una CA, debemos aceptar (instalar) su certificado raíz y de ésta forma confiaremos en todos los certificados firmados por dicha CA. Un certificado raíz es un certificado autofirmado por una CA.

Cuando accedemos a una página web segura mediante el protocolo https, el servidor deberá demostrar su autenticidad mediante un certificado firmado por una CA de nuestra confianza. Si la CA no es de nuestra confianza, el navegador preguntará al usuario si desea continuar o por el contrario, cancelar la comunicación.

La comunicación se realiza de forma segura ya que se utilizan algoritmos de cifrado asimétrico. Para saber más del cifrado asimétrico, consultar el apartado **Autentificacion segura con OpenLDAP**.

Confiar en un certificado de una CA desconocida

Nuestro servidor Linux puede comportarse como una CA y ofrecer certificados a un solicitante. Crearemos nuestra propia CA para poder utilizar páginas web seguras en nuestro servidor web Apache y para otros servicios como LDAP, mediante el protocolo SSL. Nuestra CA no será válida en Internet y sólo tendrá vigencia en el ámbito de nuestro dominio (ejemplo: 'ieslapaloma.com') pero obviamente es suficiente para el fin que pretendemos.

Instalación de OpenSSL

OpenSSL es el software que nos permitira crear certificados y convertirnos en una entidad certificadora. Utilizaremos apt-get para instalarlo:

```
// Instalación de OpenSSL
sudo apt-get install openssl
```

Configuración de OpenSSL

El archivo de configuración de openssl es **/etc/ssl/openssl.cnf**. En dicho archivo únicamente vamos a configurar los valores por defecto de nuestra organización para que el resto de aplicaciones y programas que usen openssl tomen dichos valores por defecto de forma automática. Dichos valores debemos configurarlos en la sección [req_distinguished_name]. En el resto de secciones no es necesario que modifiquemos nada ya que nos sirve con las opciones configuradas por defecto.
// Configuración particular de nuestra CA. Archivo /etc/ssl/openssl.cnf

[req_distinguished_name]

countryName = Country Name (2 letter code)
countryName_default = ES
countryName_min = 2
countryName_max = 2

stateOrProvinceName = State or Province Name (full name)
stateOrProvinceName_default = Soria

localityName = Soria

0.organizationName = Organization Name (eg, company)
0.organizationName_default = I.E.S. La Paloma

we can do this but it is not needed normally #1.organizationName = Second Organization Name (eg, company)
#1.organizationName_default = World Wide Web Pty Ltd

organizationalUnitName = I.E.S. La Paloma
#organizationalUnitName_default =

commonName = www.ieslapaloma.com
commonName_max = 64

emailAddress = root@ieslapaloma.com
emailAddress_max = 64

Módulo ssl para apache2

Para poder disfrutar con apache de un servidor seguro, debemos activar el módulo ssl mediante el siguiente comando:

```
// Habilitar el módulo ssl
# a2enmod ssl

// Reiniciamos apache para poder utilizar ssl
# /etc/init.d/apache2 restart
```

Generar el certificado

Para que nuestro servidor pueda servir páginas seguras con el protocolo https, necesita un certificado. Dicho certificado permitirá que nuestro servidor pueda utilizar cifrado asimétrico para intercambiar las claves de cifrado con los clientes, antes de iniciar una trasmisión segura de información. Inicialmente, el cliente deberá aceptar el certificado del servidor, ya que generaremos un certificado autofirmado. Si queremos evitarlo, deberíamos contratar un certificado a una entidad certificadora confiable, pero tiene un coste que no merece la pena soportar en un entorno educativo. Para generar nuestro certificado autofirmado, ejecutaremos el comando:

```
// Generar certificado autofirmado
# make-ssl-cert /usr/share/ssl-cert/ssleay.cnf /etc/ssl/certs/apache2.pem
```

Durante la ejecución de comando make-ssl-cert, quizás nos pregunte algunas sencillas preguntas como el nombre del servidor, el país, etc... y después se creará el archivo /etc/ssl/certs/apache.pem que contiene la claves que permitirán al servidor utilizar cifrado asimétrico. El siguiente paso será configurar un servidor virtual para que utilice dicho certificado.

Crear servidor virtual seguro en apache2

Primero crearemos una carpeta de nombre 'websegura' dentro de '/var/www'. Dicha carpeta será el raíz de documentos (DocumentRoot) de nuestro servidor virtual seguro, de modo que todo lo que coloquemos en dicha carpeta deba ser accedido vía 'https'. Eso lo indicaremos más adelante mediante el parámetro SSLRequireSSL. El protocolo https utiliza el puerto 443, por lo tanto, tendremos habilitar dicho puerto para que apache lo utilice. Si ya está habilitado el puerto 443, no hacer nada.

```
// Habilitar puerto 443. Añadir en /etc/apache2/ports.conf
Listen 443
```

Después debemos crear el servidor virtual en apache. Dicho servidor virtual dispondrá de una url de acceso diferente a la de nuestra web principal (websegura.ieslapaloma.com en nuestro ejemplo) y será accesible mediante https, por tanto tendremos que habilitar SSL e indicar la ruta del archivo que contiene el certificado. Todo ello lo haremos editando el archivo /etc/apache2/sites-available/default:

```
// Añadir al final en /etc/apache2/sites-available/default

NameVirtualHost websegura.ieslapaloma.com:443
<VirtualHost websegura.ieslapaloma.com:443>
ServerName websegura.ieslapaloma.com
DocumentRoot /var/www/websegura
SSLEngine On
SSLCertificateFile /etc/ssl/certs/apache2.pem
ErrorLog /var/log/apache2/error.log
CustomLog /var/log/apache2/access.log combined
</VirtualHost>

<Directory "/var/www/websegura">
Options Indexes FollowSymlinks MultiViews
AllowOverride None
Order allow,deny
Allow from all
SSLRequireSSL
</Directory>
```

Probando el acceso a la página web segura

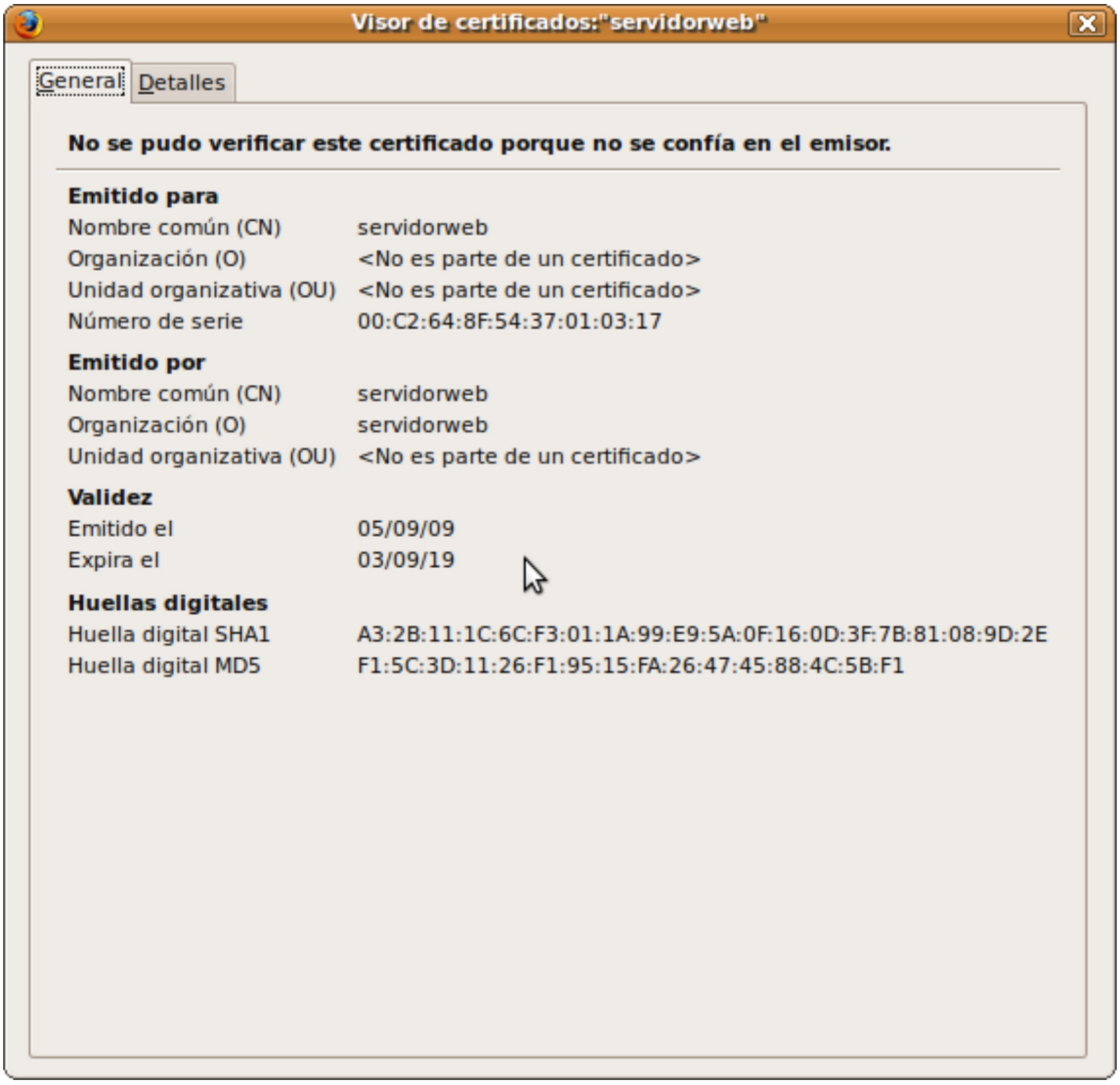
Nota: Si no tenemos un DNS funcionando, debemos incluir en /etc/hosts una línea para resolver localmente el nombre de nuestro servidor por su IP, porque en este caso, navegar con la dirección IP no funcionará. Ejemplo:

```
//Resolver el nombre localmente. Añadir en /etc/hosts
192.168.1.239 websegura.ieslapaloma.com
```

Para acceder a las páginas seguras de nuestro servidor web, tecleamos desde el navegador 'https://websegura.ieslapaloma.com'. Lo primero que se muestra es la alerta de seguridad que nos indica que el certificado no está emitido por una CA en la que confiamos:

Acceso a una página segura de nuestro servidor web

Para continuar debemos ir a añadir una excepción > obtener certificado. Si pulsamos sobre el botón 'Ver' veremos la información tanto del certificado como de la entidad certificadora que lo firma:



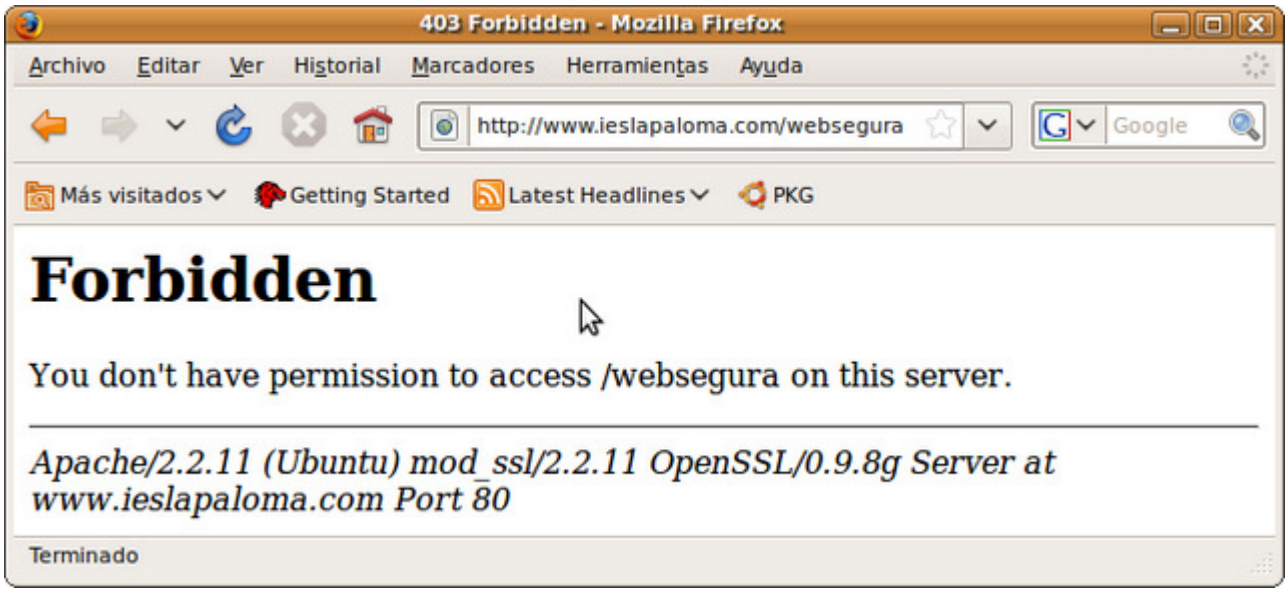
Se añade la excepción > obtener certificado

Si aceptamos el certificado significa que, a pesar de estar firmado por una entidad certificadora que no es de confianza para el navegador (lo hemos firmado nosotros mismos), lo aceptamos. Tendremos que indicar al navegador si aceptamos el certificado para siempre o solo para ahora. Como tenemos la seguridad de que el certificado es bueno porque acabamos de crearle nosotros mismos, podemos aceptarlo para siempre y así el navegador no volverá a preguntarnos más sobre él ya que hemos indicado manualmente que confiamos en este certificado.

Ahora ya tenemos acceso a la web segura mediante el protocolo https lo que nos garantiza que la información de la página segura, antes de salir del servidor, ha sido cifrada y por tanto la trasferencia de datos desde el servidor a nuestro navegador se ha producido de forma segura. Al llegar a nuestro navegador, se han descifrado los datos. El candado cerrado que aparece abajo a la derecha en el navegador, indica que la trasferencia de datos se ha realizado de forma segura.

Acceso a la web segura

Como sabemos la ruta de la carpeta segura, si intentamos acceder a la carpeta segura utilizando el protocolo http yendo con el navegador a 'http://www.ieslapaloma.com/websegura', apache denegará el acceso ya que en '/etc/apache2/sites-available /default' se ha especificado que la carpeta debe ser accedida mediante https:



Acceso denegado por el servidor Apache

Carpetas seguras de usuario

Si en el centro existiera la necesidad de que los profesores dispongan de una carpeta web segura donde poder colocar contenidos accesibles vía SSL, como serán casos excepcionales, una solución sencilla es crear una carpeta dentro de la carpeta '/var/www/websegura' para dicho profesor y para que éste tenga acceso de forma autónoma a subir contenidos a dicha carpeta, se le puede crear un usuario adicional cuyo home sea la carpeta correspondiente, ejemplo, para el profesor Javier podemos crear otro usuario llamado javier-s (javier-seguro) cuyo home sea /var/www/websegura/javier. Podría subir contenidos por ftp utilizando el usuario javier-s. El acceso a los contenidos desde un navegador sería yendo a la dirección https://websegura.ieslapaloma.com/javier

Este proceso habría que hacerlo para todos los profesores o departamentos de nuestro centro que requieran de carpeta segura.

Archivos log de apache

Por defecto, apache utiliza dos archivos de registro: access.log y error.log que están almacenados en la carpeta /var/log /apache2.

En el archivo **/var/log/apache2/access.log**, apache va registrando todos los accesos que los PCs hacen al servidor web y en cada línea de dicho archivo va almacenando la IP, la fecha y la hora, el comando HTTP enviado por el cliente, la url solicitada y la versión del navegador y el sistema operativo. Analizando este archivo podemos ver las veces que se ha descargado una página o un archivo, o las IPs más activas. Este archivo de registro es utilizado por los programas que presentan estadísticas de acceso al servidor web como awstats.

En el archivo **/var/log/apache2/error.log**, apache registra todas las incidencias o errores que se van produciendo. Ejemplo, cuando un cliente solicita una página inexistente o cuando un cliente intenta entrar en una carpeta prohibida o protegida. Si estamos configurando algo en apache (carpetas privadas, carpetas seguras, servidores web virtuales, alias, etc...) y no funciona, una buena idea es hacer pruebas y examinar el archivo error.log ya que nos puede dar pistas para encontrar la solución a nuestro problema.

```
//Ver últimas 20 líneas del access.log para ver quien está accediendo
# tail -n 20 /var/log/apache2/access.log
```

Pregunta Verdadero-Falso

¿El acceso a una carpeta segura es igual que el acceso a una carpeta privada?

Verdadero ○ Falso ○