Instalación y Mantenimiento de Servicios de Internet

Unidad 7

Servidor Web Apache seguro

IES Gonzalo Nazareno

Curso 2009/2010

María Jesús Gragera Fernández

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 algoritmos de cifrado simétricos, pero para intercambiar las claves de cifrado, hay una sesión inicial de cifrado asimétrico.

Generar el certificado

Para que nuestro servidor pueda servir páginas seguras con el protocolo https, necesita un certificado.

Dicho certificado permitirá que nuestro servidor 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.

Nos vamos a la carpeta /etc/apache2 y creamos la carpeta ssl y dentro de dicha carpeta generamos los ficheros necesarios para nuestro certificado

Para verificar que efectivamente contamos con todos los requisitos tecleamos en la terminal:

 # apt-get install apache2 apache2.2-common apache2-utils openssl openssl-blacklist openssl-blacklist-extra

Para configurar un servidor seguro, se utiliza en este caso criptografía de clave pública para crear un par de llaves, una pública y una privada. En la mayoría de los casos, el certificado que se genera localmente en el equipo tiene que ser enviado junto con varios requisitos a una Autoridad Certificadora (CA por sus siglas en inglés). La CA verifica su petición y su identidad, y luego le devuelve un certificado para asegurar el servidor. En este caso como no tenemos ni el dinero ni el tiempo suficiente (tendremos

que crear nuestro propio certificado, firmado por nosotros mismos. Sin embargo, es importante recalcar que estos certificados no deberán ser utilizados en ambientes de producción. El problema con este tipo de certificados es que no son aceptados de manera automática por los navegadores, por lo que tendremos que (al menos en FireFox) crear una excepción para el sitio con el cerficado que estamos creando nosotros. Pero para fines de demostrar los pasos será más que suficiente.

Primero: Generamos un CSR (Petición de Firma de Certificado)

Para generar el CSR debemos crear nuestra llave (key):

openssl genrsa -des3 -out server.key 4096

Puede ejecutar su propio servidor sin una clave. Esto es conveniente porque no necesitaría en este caso estar introduciendo la clave cada que requiera iniciar el servidor de https. Pero es demasiado inseguro, no recomendable ya que compromete seriamente la seguridad del sistema. De cualquier manera, es posible elegir ejecutar el servidor seguro sin clave eliminando la opción -des3 con lo que el comando quedaría de la siguiente manera:

openssl rsa -in server.key -out server.key.insegura

Al ejecutar este comando en cualquiera de sus dos formas presentadas anteriormente el sistema nos requerirá varia información que acompañará al certificado, la ubicación de la compañía, el nombre de la misma, el nombre de quien firma el certificado y varias cosas mas. Todos estos datos son guardados en el archivo /etc/ssl/openssl.cnf. Si se requieren mas llaves para más servidores o sitios que se ejecutarán en el mismo equipo se pueden agregar mediante:

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

Este certificado (el archivo server.csr en este caso) es el que deberemos enviar a la Autoridad Certificadora (CA).

Una vez que hemos generado nuestro certificado necesitamos instalarlo en el servidor.

Segundo: Crear un Certificado Firmado por nosotros mismos.

Para firmar nuestro certificado de manera local ejecutamos:

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

El comando anterior nos va a requerir que introduzcamos la clave de nuestra llave (si optamos por utilizar una clave segura, de otro modo no solicitará esta información), que es la clave que introdujimos al crear nuestra Petición de Firma de Certificado (CSR). Una vez que hemos ingresado la clave correcta, se generará un cerficiado y se almacenará con el nombre server.crt en nuestro servidor.

Tercero: Instalar el Certificado

Para instalar el certificado vamos a copiar los archivos server.crt y server.key al fichero /etc/apache2/ssl creado anteriormente

Cuarto: Habilitar el módulo SSL para Apache2

Ejecutamos el siguiente comando:

 # a2enmod ssl

Quinto: Crear y habilitar el sitio SSL

Vamos a crear un VirtualHost para nuestro sitio:

#/etc/apache2/sites-available/default /etc/apache2/sites-available/websegura

Luego, editamos el archivo websegura

nano /etc/apache2/sites-available/websegura

Y lo dejamos con el siguiente contenido:

```
<VirtualHost *:443>
   ServerAdmin webmaster@localhost
            ServerName www.misitio.com
            LogLevel warn
            ErrorLog /var/log/apache2/www.misitio.com.error.log
            CustomLog /var/log/apache2/www.misitio.com.access.log combined

            DocumentRoot /var/www/www.misitio.com/

            <Directory /var/www/www.misitio.com/>
                        Options Indexes FollowSymLinks MultiViews
                        AllowOverride All
                        Order allow,deny
                        allow from all
            </Directory>

       SSLEngine On
       SSLCertificateFile    /etc/apache2/ssl/server.crt
       SSLCertificateKeyFile /etc/apache2/ssl/server.key
       SetEnvIf User-Agent ".*MSIE.*" nokeepalive ssl-unclean-shutdown
</VirtualHost>
```

Por último, habilitamos el sitio que acabamos de crear:

 # a2ensite www.misitio.com

Sexto: Habilitar Apache2 para que escuche en el puerto 443.

Necesitamos editar el archivo ports.conf:

 # nano /etc/apache2/ports.conf

Al abrirlo se ve de la siguiente manera:

If you just change the port or add more ports here, you will likely also
have to change the VirtualHost statement in

/etc/apache2/sites-enabled/000-default

NameVirtualHost *:80

Listen 80

SSL name based virtual hosts are not yet supported, therefore no

NameVirtualHost statement here

Listen 443

Vamos a agregar el NameVirtualHost al puerto de https (443). Así que al terminar deberá quedar de esta manera:

If you just change the port or add more ports here, you will likely also

have to change the VirtualHost statement in

/etc/apache2/sites-enabled/000-default

NameVirtualHost *:80

Listen 80

SSL name based virtual hosts are not yet supported, therefore no
NameVirtualHost *:443

Listen 443

No hay que olvidar editar el archivo /etc/hosts y añadir nuestro servidor o como en nuestro caso añadirlo a nuestro servidor DNS.

Ahora reiniciamos el servidor:

 # /etc/init.d/apache2 restart

Si se ha optado por utilizar una llave con clave, el servidor requerirá que tecleemos la contraseña para poder continuar, en caso contrario no arrancará el servicio.

Séptimo: Acceder al servidor.

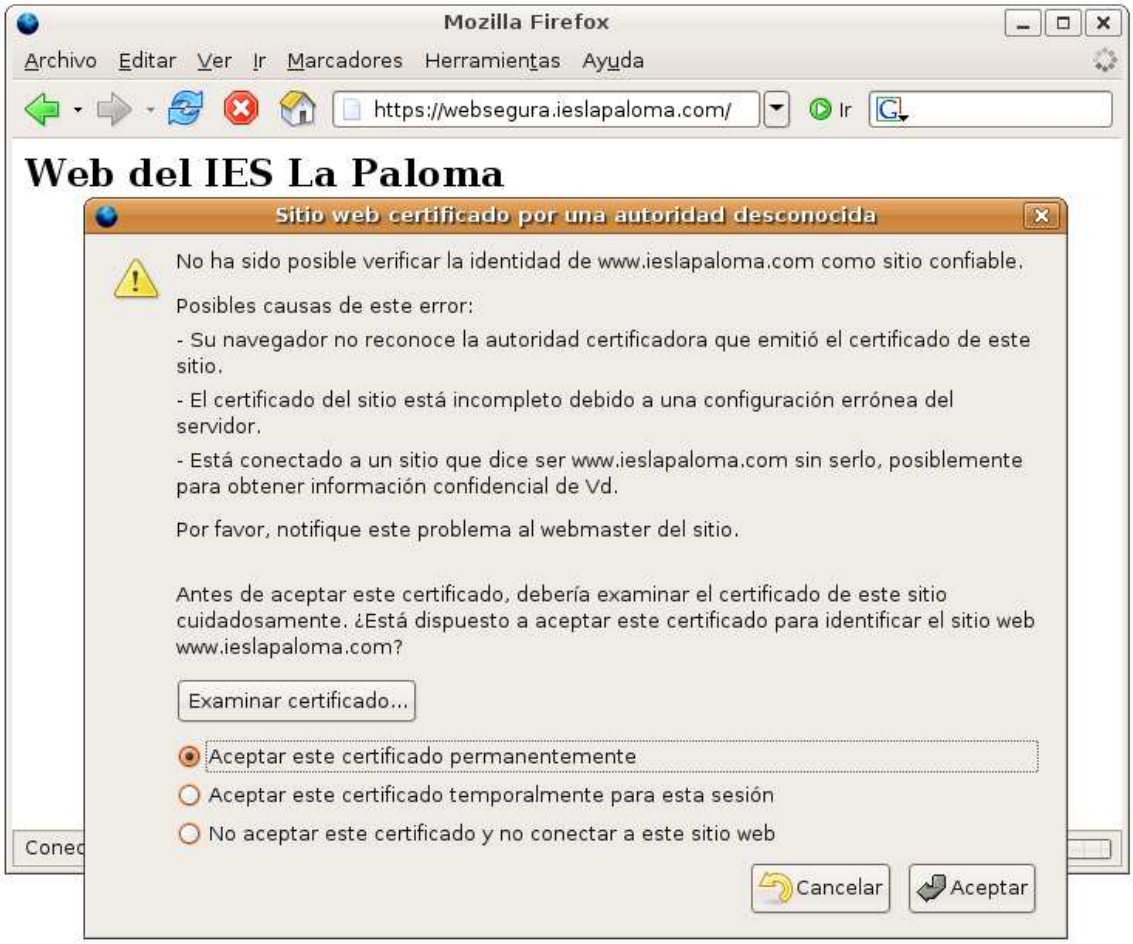
Ahora es posible acceder al servidor en la dirección https://www.misitio.com. Si se ha utilizado un certificado que nosotros mismos hemos firmado, el navegador nos indicará que ha fallado la conexión segura, es cuestión de ignorar simplemente el mensaje y agregar una excepción de seguridad para nuestro sitio.

/etc/init.d/apache2 restart

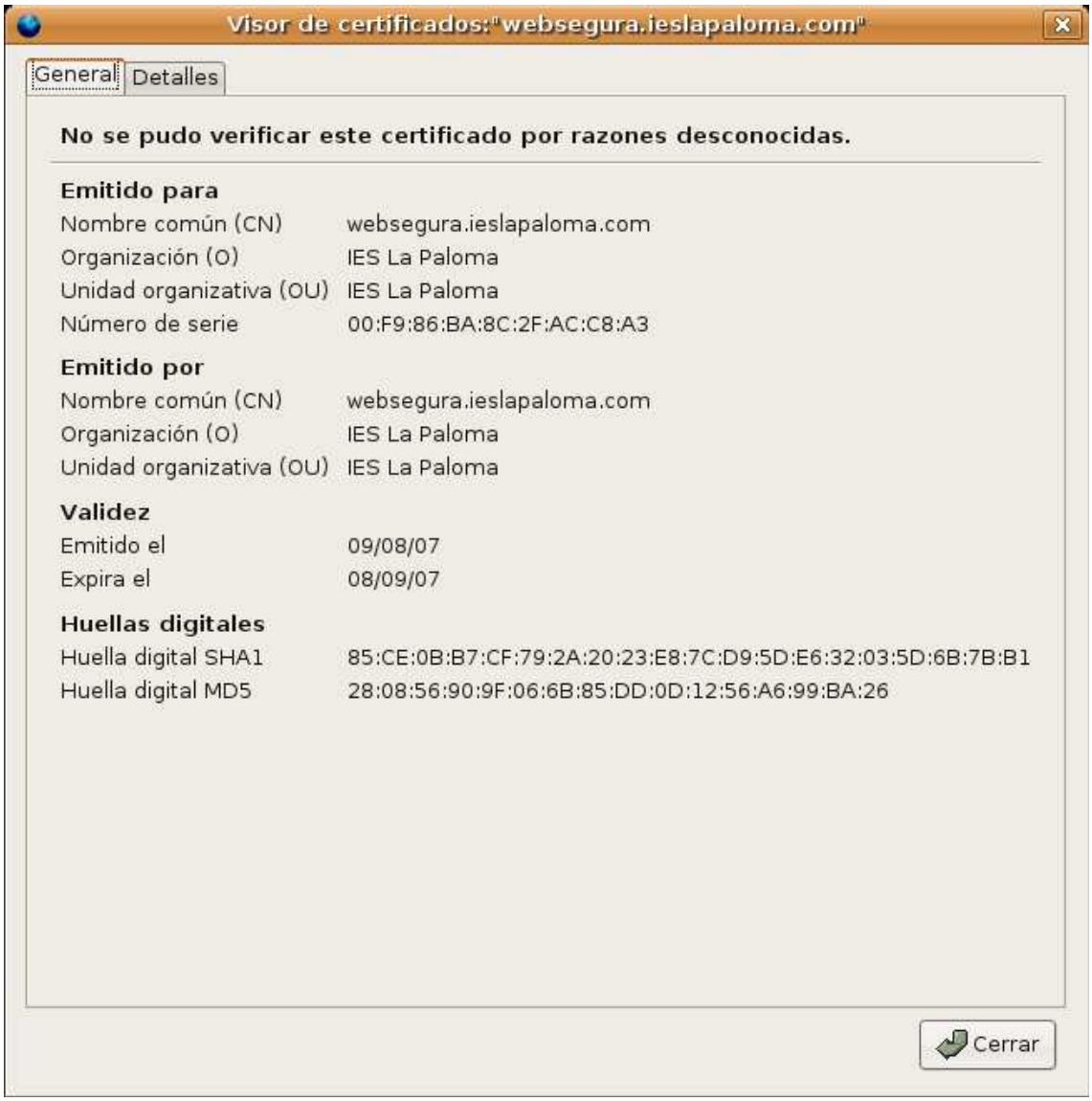
Probando el acceso a la página web segura

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, ejemplo: 192.168.1.239 **websegura.gonzanaza.com**, ya que en este caso, navegar con la dirección IP no funcionará.

Para acceder a las páginas seguras de nuestro servidor web, tecleamos desde el navegador 'https://websegura.gonzanaza.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:

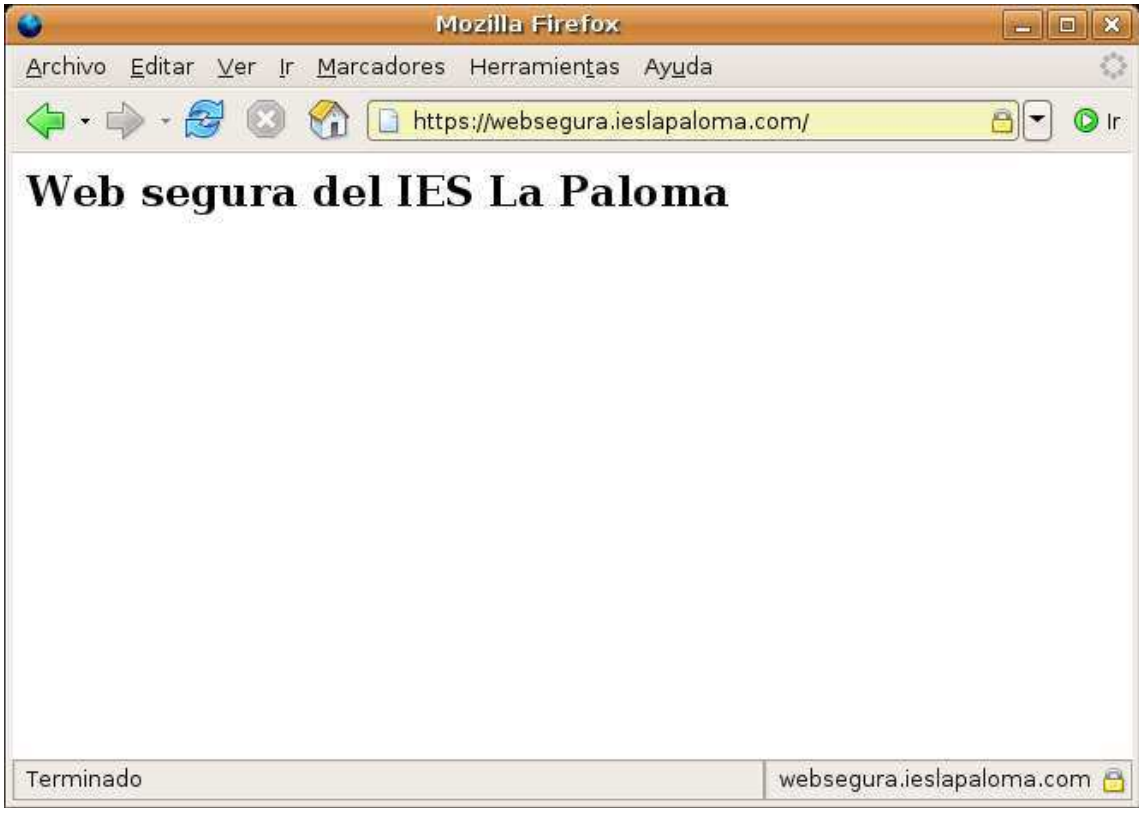


Si pulsamos sobre el botón 'Examinar certificado' veremos la información tanto del certificado como de la entidad certificadora que lo firma:

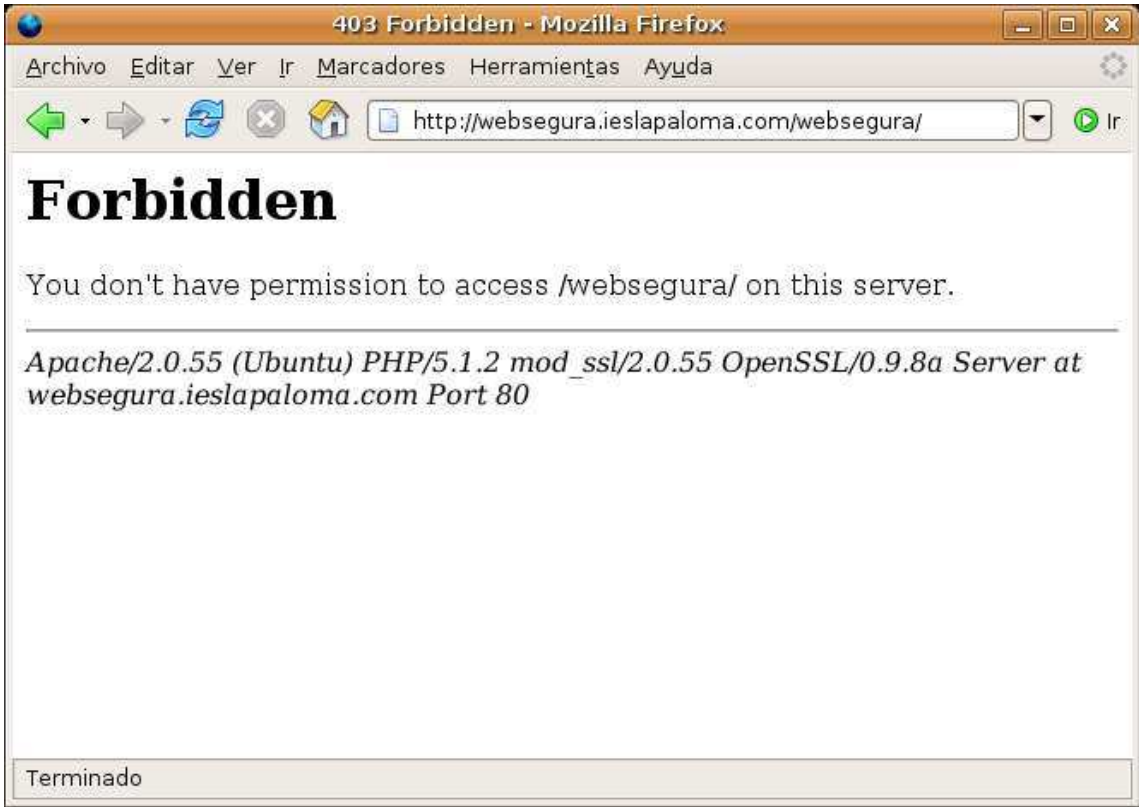


Si aceptamos el certificado significa que, a pesar de estar firmado por una entidad certificadora que no es de confianza para el navegador (lo hemos firmado nosotros mismos), lo aceptamos. Tendremos que indicar al navegador si aceptamos el certificado para siempre o solo para ahora. Como tenemos la seguridad 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.



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.gonzanaza.com/websegura', apache denegará el acceso ya que en 'apache2.conf' se ha especificado que la carpeta debe ser accedida mediante https:



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