

**SWAP (2014-2015)**  
GRADO EN INGENIERÍA INFORMÁTICA  
UNIVERSIDAD DE GRANADA

---

# SWAP

---

21 de mayo de 2015

# Índice

<b>1. Introducción.</b>	<b>3</b>
1.1. ¿ Qué es Apache ?.	3
1.2. ¿ Por qué se utiliza Apache? .	3
1.3. Conclusión de elección del tema del trabajo .	4
<b>2. Directivas de seguridad en Apache.</b>	<b>4</b>
2.1. Directiva ServerSignature y ServerTokens .	4
<b>3. Directiva DocumentRoot y Alias</b>	<b>5</b>
<b>4. Directiva Directory.</b>	<b>7</b>
<b>5. Otras directivas.</b>	<b>8</b>
<b>6. Crear un certificado SSL de firma propia con OpenSSL y Apache HTTP Server.</b>	<b>9</b>

## Índice de figuras

2.1. Visualización de la orden curl. . . . .	4
2.2. Visualización del valor de la directiva. . . . .	4
2.3. Visualización del valor de la directiva. . . . .	5
3.1. Visualización del valor de la directiva. . . . .	6
3.2. Primera demostración de la directiva alias. . . . .	6
3.3. Segunda demostración de la directiva alias. . . . .	6
4.1. Visualización del valor de la directiva. . . . .	7
4.2. Demostración paso 1. . . . .	7
4.3. Demostración paso 2. . . . .	8
4.4. Demostración paso 3. . . . .	8
6.1. Demostración paso 3. . . . .	10
6.2. Demostración paso 3. . . . .	11
6.3. Demostración paso 3. . . . .	11
6.4. Demostración paso 3. . . . .	12
6.5. Demostración paso 3. . . . .	12
6.6. Demostración paso 3. . . . .	13

# 1. Introducción.

## 1.1. ¿ Qué es Apache ?.

El servidor HTTP Apache es un servidor web HTTP de código abierto, se puede utilizar para plataformas como Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, las cuales implementa el protocolo HTTP/1.1.

Lo podemos ver como un programa que nos permite acceder a las páginas web de un ordenador desde otro.

## 1.2. ¿ Por qué se utiliza Apache?

Apache es muy utilizado al porque presenta las siguientes características:

- Es multiplataforma: se puede ejecutar en varios Sistemas Operativos: Ubuntu, Microsoft...
- Es de código abierto, dándole transparencia y la capacidad de saber que es lo que estamos instalando.
- Es configurable y de código abierto, lo que nos permite cambiar la funcionalidad y con ello cambiar las características de los servicios que estamos ofreciendo a partir de las directivas, las cuales podemos ver desde:

<http://httpd.apache.org/docs/2.0/es/mod/directives.html>

- Trabaja con una gran variedad de lenguajes de programación ( PHP, JAVA), que nos facilitará el trabajo para desarrollar páginas web dinámicas( que son aquellas en la cual la información se genera a partir de una petición realizada por un usuario de la página).
- Tiene archivos Log, que poseen registros de información global del sistema (errores producidos en un determinado tiempo, los cuales pueden determinar la política de seguridad a seguir por los administradores del sistema). Podemos ver para que sirve cada uno en:

<http://httpd.apache.org/docs/2.0/es/logs.html>

- Es fácil obtener soporte, ya que al ser muy utilizado, podemos consultar en una inmensa cantidad de tutoriales y soportes de ámbito informático con el fin de resolver nuestros problemas. Además en su página web oficial

<http://www.apache.org>

nos ofrece una gran cantidad de ayuda, la cual desde el principio del trabajo se puede ver que la hemos usado( en los enlaces para las directivas y archivos log).

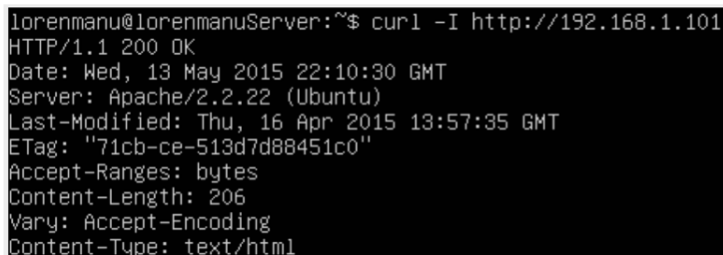
### 1.3. Conclusión de elección del tema del trabajo

Por ello nuestro trabajo tratará sobre la seguridad de este servidor, destacando las directivas que serán útiles para ese objetivo y finalmente abarcaremos la encriptación.

## 2. Directivas de seguridad en Apache.

### 2.1. Directiva ServerSignature y ServerTokens

Para realizar una demostración de estas directivas necesitaremos el comando “curl” con la opción “-I”, la cual nos permitirá ver solo las cabeceras, sin necesidad de descargarnos la página, como podemos ver ahora:



```
lorenmanu@lorenmanuServer:~$ curl -I http://192.168.1.101
HTTP/1.1 200 OK
Date: Wed, 13 May 2015 22:10:30 GMT
Server: Apache/2.2.22 (Ubuntu)
Last-Modified: Thu, 16 Apr 2015 13:57:35 GMT
ETag: "71cb-ce-513d7d88451c0"
Accept-Ranges: bytes
Content-Length: 206
Vary: Accept-Encoding
Content-Type: text/html
```

Figura 2.1: Visualización de la orden curl.

En la anterior imagen podríamos ver la siguiente línea:

HTTP/1.1 200 OK

En esta línea se especifica el resultado de la petición que se realizó al servidor, en este caso nos quiere decir que la petición ha sido correcta. La información ofrecida por esta línea puede servir para ataques Cross Site Scripting o XSS, por lo que es aconsejable tenerla desactivada, para ello podemos comprobarla accediendo al archivo:

`/etc/apache2/conf.d/security`

Y ponemos la directiva TraceEnable con el siguiente valor:



```
TraceEnable Off
```

Figura 2.2: Visualización del valor de la directiva.

Des esta manera al volver a hacer curl a la página no nos saldrá la información de la petición, en la línea anteriormente dicha nos saldrá:

HTTP/1.1

Otra directiva es `ServerTokens`, la cual indica en el campo `Server` de las cabeceras de las respuestas si se especifica el sistema operativo del servidor así como información sobre los módulos compilados. Esta información puede ser usada por un atacante, para solucionarlo Apache ofrece diferentes soluciones, se recomienda utilizar la siguiente, ya que en ella solo se especifica el servidor que se usa, por lo que en el archivo :

`/etc/apache2/conf.d/security`

le pondremos a la directiva `ServerTokens` el siguiente valor:

A screenshot of a terminal window showing the command `ServerTokens Prod` in a light blue font on a black background.

Figura 2.3: Visualización del valor de la directiva.

De tal manera que en el campo `Server` de las cabeceras de respuesta nos saldrá:

`Server: Apache`

### 3. Directiva `DocumentRoot` y `Alias`

Esta directiva especifica el directorio desde el cuál `httpd` servirá los ficheros. A menos que especifique alguna otra equivalencia mediante una directiva `Alias`, el servidor añade la ruta de la URL solicitada a este directorio para construir la ruta del documento a servir. Ejemplo:

`DocumentRoot /usr/web`

esto quiere decir que una petición de acceso a

`http://www.my.host.com/index.html`

se refiere a

`/usr/web/index.html`

en el sistema de ficheros.

Podemos cambiar el valor de esta directiva desde:

`/etc/apache2/apache2.conf`

Una directiva similar sería `Alias`, la cual nos da la capacidad de acceder a un directorio mediante un nombre, es decir, si tuviéramos un directorio denominado `/var/www/sitio5`, así pues si pusieramos al final de la directiva `Directory`:

```
Alias /prueba1 "/var/www/index.html"
```

```
#<Directory />
#     AllowOverride None
#     Order Deny,Allow
#     Deny from all
#</Directory>

Alias /prueba1 "/var/www/index.html"
```

Figura 3.1: Visualización del valor de la directiva.

Podríamos acceder a index.html de dos formas tal y como mostramos en las siguientes dos figuras:

```
lorenmanu@lorenmanuServer:~$ curl http://192.168.1.100/index.html
<html><body><h1>It works!</h1>
<p>This is the default web page for this server.</p>
<p>The web server software is running but no content has been added, yet.</p>
<p> Soy la maquina original.
</body></html>
lorenmanu@lorenmanuServer:~$
```

Figura 3.2: Primera demostración de la directiva alias.

```
lorenmanu@lorenmanuServer:~$ curl http://192.168.1.100/prueba1
<html><body><h1>It works!</h1>
<p>This is the default web page for this server.</p>
<p>The web server software is running but no content has been added, yet.</p>
<p> Soy la maquina original.
</body></html>
lorenmanu@lorenmanuServer:~$
```

Figura 3.3: Segunda demostración de la directiva alias.

Podemos cambiar el valor de esta directiva en el archivo:

```
/etc/apache2/conf.d/security
```

Reflexión sobre las directivas: estas directivas como podemos observar nos servirán para indicar las rutas de archivos que ofreceremos en nuestro servidor web, por lo que será importante comprobarlas cuando tengamos nuestro servidor para que los atacantes o usuarios no accedan a archivos que nosotros no queramos.

## 4. Directiva Directory.

Nos ofrece un grupo de directivas que se aplicarán solamente al directorio del sistema de ficheros especificado y a sus subdirectorios. Su sintaxis es la siguiente:

```
<Directory /usr/local/httpd/htdocs>
Options Indexes FollowSymLinks
</Directory>
```

Unas de las opciones que nos ofrece y que nos ha sido muy importante destacar es la de permitir a unas IPs determinadas acceder a la ruta de archivos especificada, es decir, establecemos un firewall. Para accederemos al archivo de configuración:

```
<Directory /usr/local/httpd/htdocs>
/etc/apache2/conf.d/security
</Directory>
```

Y pondremos lo siguiente:

```
#<Directory />
#     AllowOverride None
#     Order Deny,Allow
#     Deny from all
#     Allow from 192.168.1.100
#</Directory>
```

Figura 4.1: Visualización del valor de la directiva.

En el cual estamos indicando que niegue todos los equipos y que permita solo a los que tiene la IP 192.168.1.100.

Así pues podemos ver que si el equipo tiene la dirección IP 192.168.1.100 no accederá a la página web:

```
eth1      Link encap:Ethernet  direcciónHW 08:00:27:41:7a:40
Direc. inet:192.168.1.100  Difus.:192.168.1.255  Másc:255.255.255.0
Dirección inet6: fe80::a00:27ff:fe41:7a40/64 Alcance:Enlace
ACTIVO DIFUSIÓN FUNCIONANDO MULTICAST MTU:1500 Métrica:1
Paquetes RX:129 errores:0 perdidos:0 overruns:0 frame:0
Paquetes TX:106 errores:0 perdidos:0 overruns:0 carrier:0
colisiones:0 long.colatX:1000
Bytes RX:18519 (18.5 KB)  TX bytes:16508 (16.5 KB)
lo        Link encap:Bufo local
```

Figura 4.2: Demostración paso 1.

```
lorenmanu@lorenmanuServer:~$ curl http://192.168.1.100
<html><body><h1>It works!</h1>
<p>This is the default web page for this server.</p>
<p>The web server software is running but no content has been added, yet.</p>
<p> Soy la maquina original.
</body></html>
lorenmanu@lorenmanuServer:~$ _
```

Figura 4.3: Demostración paso 2.

```
lorenmanu@lorenmanuServer:~$ curl http://192.168.1.100
curl: (7) couldn't connect to host
lorenmanu@lorenmanuServer:~$
```

Figura 4.4: Demostración paso 3.

## 5. Otras directivas.

Otras directivas que se pueden destacar para evitar Ataques DDOS son las siguientes:

- **Timeout:** Esta directiva indica el tiempo que el servidor esperará para que un evento termine antes de fallar. Su valor por defecto es de 300 segundos. Es bueno mantener este valor lo más bajo posible si nuestro sitio es constantemente sometido a este tipo de ataque.
- **MaxClients:** Esta directiva nos permite configurar el número de conexiones que vamos a permitir simultáneamente. Las conexiones nuevas serán puestas en cola a partir del límite que configuremos.
- **KeepAliveTimeout:** Es el tiempo máximo que el servidor esperará para un requerimiento posterior antes de cerrar la conexión.
- **LimitRequestFields:** Nos ayuda a limitar el número de solicitudes de encabezados HTTP que aceptaremos de un cliente. Su valor por defecto es 100. Es recomendable que reduzcamos el valor si constantemente somos blancos de ataques DDOS.



## 6. Crear un certificado SSL de firma propia con OpenSSL y Apache HTTP Server.

Un certificado SSL es un certificado digital utilizado para cifrar la información de un sitio y crear conexión segura. Este certificado es proporcionado por un proveedor autorizado (Digicert, Comodo, Verisign...) y es enviado por el servidor con quien estamos establecido la conexión al cliente.

El certificado que vamos crear por OpenSSL tiene el mismo nivel de cifrado como cualquier proveedor autorizado.

Primero definimos el módulo mod-ssl que es un módulo del servidor HTTP Apache, el cual provee soporte para SSL versiones 2 y 3 y TLS versión 1. Luego seguimos estos pasos:

- Activamos el módulo con el siguiente comando:

```
sudo a2enmod ssl
```

- Después reiniciamos el servidor apache para efectuar los cambios:

```
sudo service apache2 restart
```

- Creamos nuevo directorio donde vamos a almacenar la clave del servidor y el certificado:

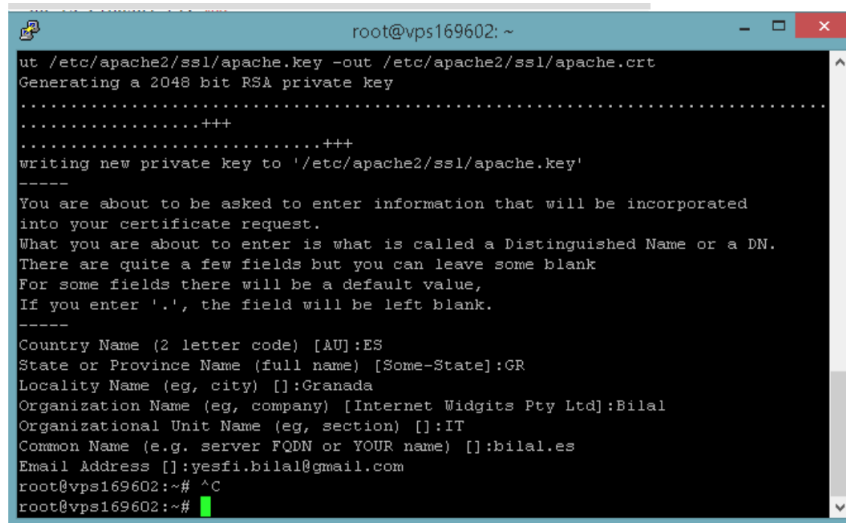
```
sudo mkdir /etc/apache2/ssl
```

- Creamos un certificado autofirmado SSL. Cuando solicitamos un nuevo certificado, podemos especificar el tiempo de validez del certificado con cambiar el número de 365 al número de días que preferimos. En nuestro caso el certificado se expira después de un año.

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc.
```

Con el anterior comando, creamos el certificado SSL autofirmado y la clave del servidor que lo protege, y colocamos ambos en el nuevo directorio.

El campo más importante es “Common Name” donde debe introducir el nombre de su dominio o la dirección IP.

A terminal window titled 'root@vps169602: ~' showing the execution of the command 'openssl req -x509 -newkey rsa:2048 -keyout /etc/apache2/ssl/apache.key -out /etc/apache2/ssl/apache.crt'. The terminal output includes: 'Generating a 2048 bit RSA private key', a separator of dots, 'writing new private key to "/etc/apache2/ssl/apache.key"', and a series of prompts for a Distinguished Name (DN). The user provides the following information: Country Name (AU), State or Province Name (Some-State), Locality Name (Granada), Organization Name (Internet Widgits Pty Ltd), Organizational Unit Name (IT), Common Name (bilal.es), and Email Address (yesfi.bilal@gmail.com). The prompt 'root@vps169602:~# ^C' is followed by a green cursor.

```
root@vps169602: ~
ut /etc/apache2/ssl/apache.key -out /etc/apache2/ssl/apache.crt
Generating a 2048 bit RSA private key
.....+++
.....+++
writing new private key to '/etc/apache2/ssl/apache.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]:GR
Locality Name (eg, city) []:Granada
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Bilal
Organizational Unit Name (eg, section) []:IT
Common Name (e.g. server FQDN or YOUR name) []:bilal.es
Email Address []:yesfi.bilal@gmail.com
root@vps169602:~# ^C
root@vps169602:~#
```

Figura 6.1: Demostración paso 3.

- Instalación del certificado: En este paso hay que configurar los hosts virtuales para mostrar el nuevo certificado. Abrimos el archivo de configuración SSL con el siguiente comando :

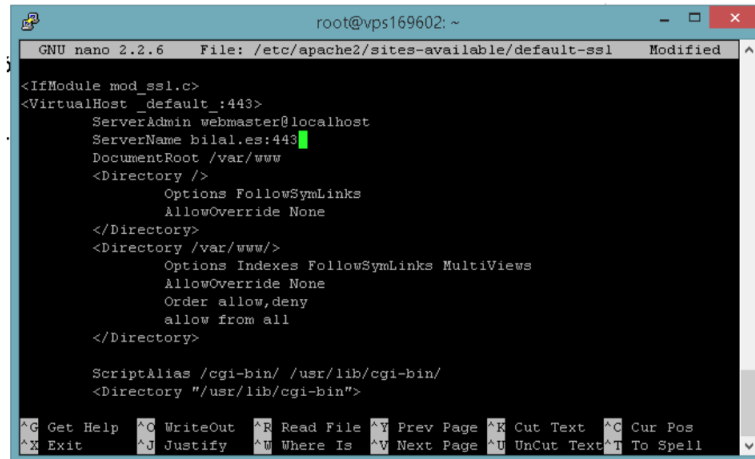
```
sudo /etc/apach2/sites-available/default-ssl
```

Luego, añadimos dentro de la sección

```
<VirtualHost _default_:443>
```

la siguiente línea:

```
ServerName example.com:443
```



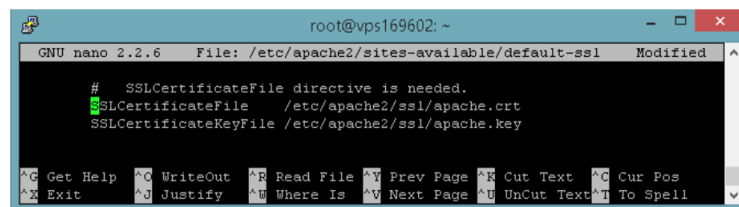
```
root@vps169602: ~
GNU nano 2.2.6 File: /etc/apache2/sites-available/default-ssl Modified
<IfModule mod_ssl.c>
<VirtualHost _default_:443>
    ServerAdmin webmaster@localhost
    ServerName bilal.es:443
    DocumentRoot /var/www
    <Directory />
        Options FollowSymLinks
        AllowOverride None
    </Directory>
    <Directory /var/www/>
        Options Indexes FollowSymLinks MultiViews
        AllowOverride None
        Order allow,deny
        allow from all
    </Directory>

    ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
    <Directory "/usr/lib/cgi-bin">

^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

Figura 6.2: Demostración paso 3.

- Busca las siguientes líneas y modifica el campo  
SSLCertificateFile con /etc/apache2/ssl/apache.crt  
SSLCertificateKeyFile con /etc/apache2/ssl/apache.key



```
root@vps169602: ~
GNU nano 2.2.6 File: /etc/apache2/sites-available/default-ssl Modified
# SSLCertificateFile directive is needed.
SSLCertificateFile /etc/apache2/ssl/apache.crt
SSLCertificateKeyFile /etc/apache2/ssl/apache.key

^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

Figura 6.3: Demostración paso 3.

Guarda los cambios y salir.

- Activar el nuevo Virtual-host:

Para que nuestro dominio use el puerto 443, hay que ejecutar el siguiente comando:

```
sudo a2ensite default-ssl
```

Finalmente, reiniciamos el apache:

```
sudo service apache2 reload
```

- Verificación del certificado SSL desde un navegador:

Cuando ingresamos la primera vez nos va a dar una alerta de que el certificado no es de confianza, esto es por no encontrar una identificación de un proveedor autorizado.

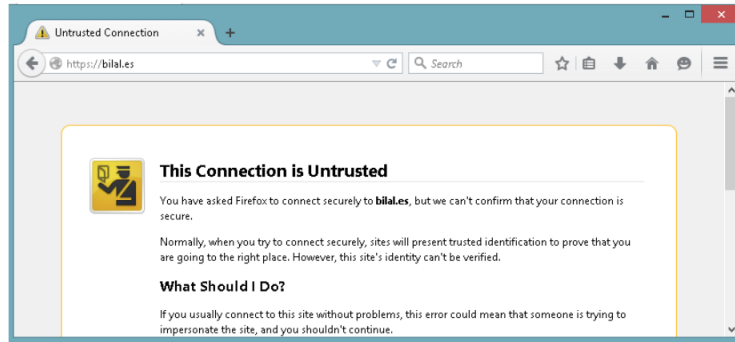


Figura 6.4: Demostración paso 3.

En caso de usar el navegador Mozilla, da clic en Entiendo el riesgo y luego en Añadir una excepción .

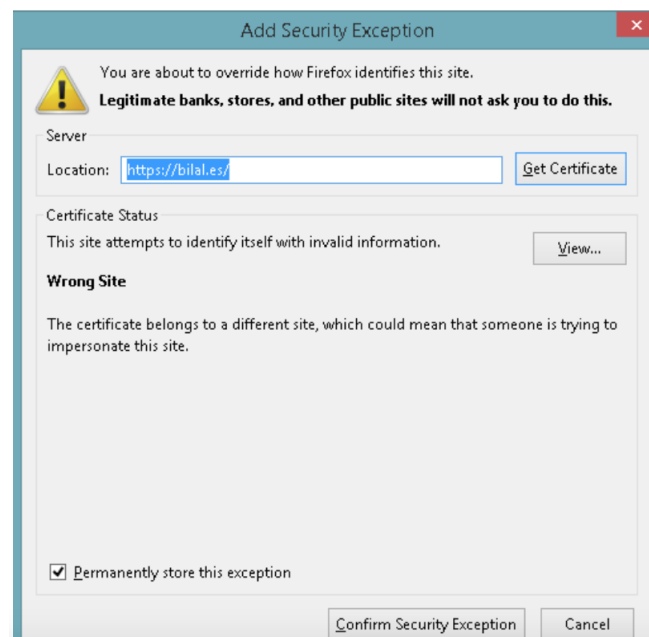


Figura 6.5: Demostración paso 3.

En la pantalla anterior da clic en Confirmar excepción de seguridad, con eso la comunicación con el dominio es segura.

Si damos clic al candado encontramos más información acerca del certificado, como por ejemplo la última vez que se modificó el certificado.

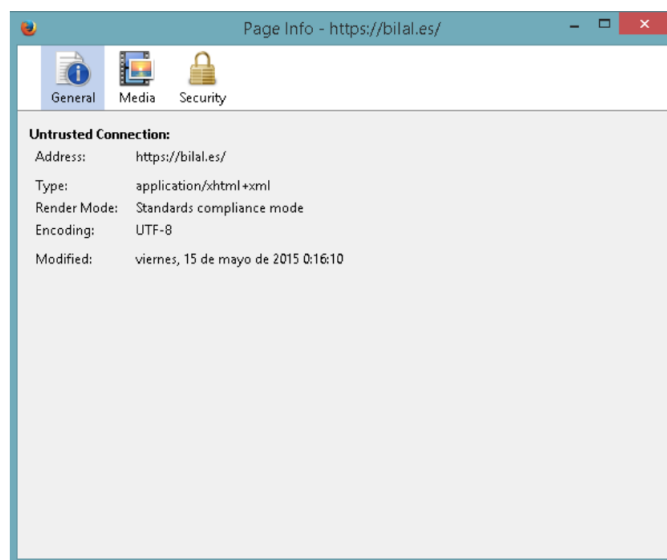


Figura 6.6: Demostración paso 3.