

Memoria de Prácticas: Servidor Web con HTTPS



OpenSSL

Cryptography and SSL/TLS Toolkit

Índice

Índice	2
Configurar un servidor web (https)	3
Crear un certificado SSL real	4
Crear su propia CA	7

Configurar un servidor web (https)

Vamos a crear un servidor web (https).

Una vez instalado correctamente el apache en el docker y funcionando perfectamente, para tener nuestro servidor web funcionando.

Vamos a convertir el servidor en seguro (https).

Para ejecutar el tráfico web a través de SSL, la configuración más sencilla es COPY montar (-v) su [server.crt](#) y [server.key](#) dentro de [/usr/local/apache2/conf/](#) y luego personalizar [/usr/local/apache2/conf/httpd.conf](#)

LoadModule socache_shmcb_module modules/mod_socache_shmcb.so

LoadModule ssl_module modules/mod_ssl.so

Include conf/extra/httpd-ssl.conf

El archivo de configuración [conf/extra/httpd-ssl.conf](#) usará los archivos de certificado agregados previamente y le indicará al daemon que también escuche en el puerto 443.

Asegúrese de agregar algo como -p 443:443 tu docker run para reenviar el puerto https.

Esto podría lograrse con una línea sed similar a la siguiente:

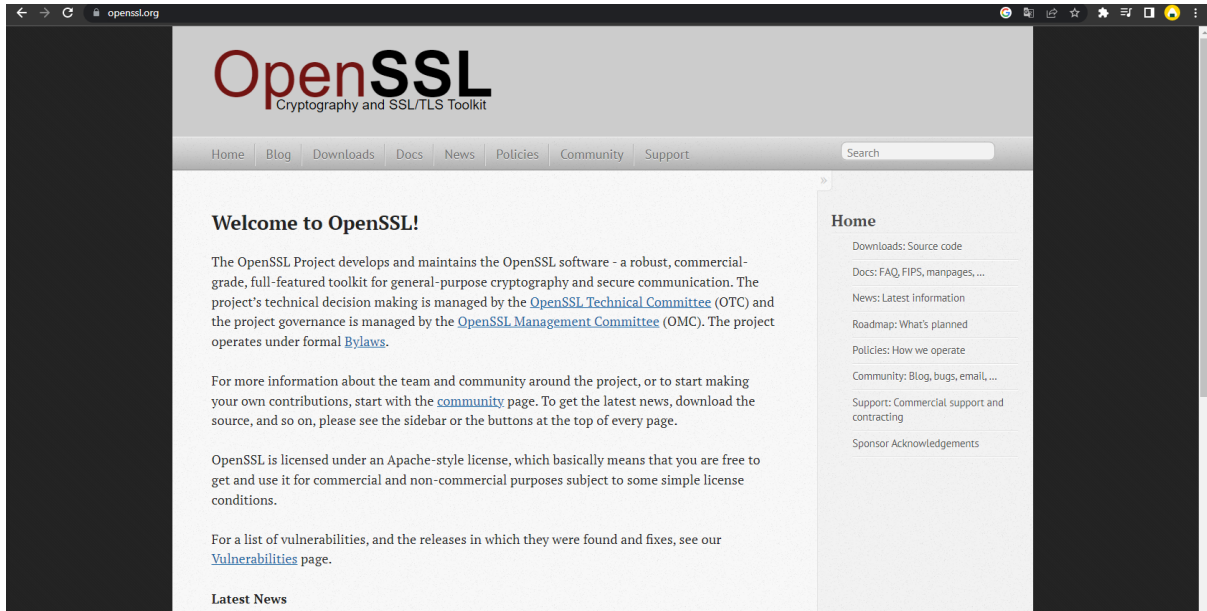
**RUN sed -i **

```
-e 's/^#(Include .*httpd-ssl.conf)/\1/' \  
-e 's/^#(LoadModule .*mod_ssl.so)/\1/' \  
-e 's/^#(LoadModule .*mod_socache_shmcb.so)/\1/' \  
conf/httpd.conf
```

Con estos pasos debería funcionar todo correctamente pero es recomendable personalizar los archivos conf para producción.

Crear un certificado SSL real

Lo primero es descargar **OpenSSL**



Con el siguiente comando en Linux: **apt install openssl**

Una vez descargado e instalado correctamente en **PATH**

Cree una clave privada RSA para su servidor Apache (estará cifrada con Triple-DES y formateada con PEM):

\$ openssl genrsa -des3 -out server.key 2048

```
estudiante@DAW1:~/Downloads/web/paginaWebServidor$ openssl genrsa -des3 -out server.key 2048
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
```

Copia de seguridad de este archivo `server.key` y la frase de contraseña que ingresó en una ubicación segura.

Puede ver los detalles de esta clave privada RSA usando el comando:

\$ openssl rsa -noout -text -in server.key

Despliegue de aplicaciones web.
Román Millán Díaz

```
estudiante@DAW1:~/Downloads/web/paginaWebServidor$ openssl rsa -noout -text -in
server.key
Enter pass phrase for server.key:
Private-Key: (2048 bit, 2 primes)
modulus:
 00:e0:b0:ea:8a:e8:5a:02:47:68:cf:65:a8:16:f2:
 1e:d8:bf:e2:22:ea:0c:93:ee:ce:40:c2:d4:7f:20:
 bb:eb:b5:17:bf:2c:78:f3:3f:5b:73:8f:84:9c:97:
 59:b3:78:3c:e5:e3:fb:a9:76:ad:f9:9d:47:03:25:
 57:76:04:a4:3e:06:40:27:50:21:ac:c2:de:41:e1:
 77:53:3c:b2:19:0c:fd:ef:da:9b:f0:5c:7d:3a:72:
 2e:68:01:fc:cc:fb:a3:f6:b5:2f:c0:a8:5a:b8:f2:
 62:dd:80:a3:67:cf:d8:4a:ea:c0:b7:0e:c7:79:2f:
 d2:3e:21:98:83:81:85:11:20:99:8d:9a:e1:51:d9:
 1f:0b:df:8f:9f:54:3b:da:90:66:a0:9a:23:d6:4a:
 05:61:35:1d:fc:16:b4:78:09:f5:80:0b:d9:89:e6:
 23:f2:df:7c:bd:28:b2:a5:3e:3b:c8:46:72:e7:61:
 2d:44:dd:a3:f3:51:a2:a6:f7:7e:38:76:f9:8e:99:
 8e:0b:96:89:1c:42:fd:f3:ec:00:88:4c:65:90:bc:
 4f:79:59:20:e7:36:52:3b:d5:9b:f3:a2:8c:68:cd:
 5d:80:7a:9d:34:71:a0:fd:71:b5:cc:85:93:49:4a:
 2e:99:90:c7:1f:b7:62:8a:7d:c6:84:ac:93:5d:08:
 fe:09
publicExponent: 65537 (0x10001)
privateExponent:
 1f:ea:16:71:03:2e:6b:0d:c8:33:52:80:72:3c:93:
 13:bd:08:2f:bc:08:16:a8:60:1f:64:0d:04:14:45:
 87:90:97:53:8e:20:e2:86:bd:ef:47:69:51:f1:fa:
 c9:ba:6b:06:9e:ea:a0:ad:67:6a:01:b9:93:c1:7a:
 ba:1b:fa:aa:dc:a9:c1:6b:be:52:12:f3:b8:19:2a:
 f6:65:b3:f3:9e:f1:75:5f:70:5a:ce:fd:9d:39:99:
 4c:7d:f6:71:b4:79:c6:3c:57:66:09:58:fe:d5:5e:
```

Si es necesario, también puede crear una versión PEM descifrada (no recomendada) de esta clave privada RSA con:

\$ openssl rsa -in server.key -out server.key.unsecure

```
estudiante@DAW1:~/Downloads/web/paginaWebServidor$ openssl rsa -in server.key -o
ut server.key.unsecure
Enter pass phrase for server.key:
writing RSA key
```

Cree una solicitud de firma de certificado (CSR) con la clave privada RSA del servidor (la salida tendrá formato PEM):

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

```
estudiante@DAW1:~/Downloads/web/paginaWebServidor$ 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]:PK  
State or Province Name (full name) [Some-State]:Sind  
Locality Name (eg, city) []:Lahore  
Organization Name (eg, company) [Internet Widgits Pty Ltd]:PTCL  
Organizational Unit Name (eg, section) []:Mian Muhammad  
Common Name (e.g. server FQDN or YOUR name) []:Arif Alvi  
Email Address []:qte11456@cdfaq.com  
  
Please enter the following 'extra' attributes  
to be sent with your certificate request  
A challenge password []:usuario  
An optional company name []:Borat
```

Asegúrese de ingresar el FQDN ("Nombre de dominio completamente calificado") del servidor cuando OpenSSL le solicite el "Nombre común", es decir cuando genere una CSR para un sitio web al que se accederá más tarde a través <https://www.foo.dom/> de , ingrese "www.foo.dom" aquí.

Puede ver los detalles de esta CSR usando:

\$ openssl req -noout -text -in server.csr

```
estudiante@DAW1:~/Downloads/web/paginaWebServidor$ openssl req -noout -text -in  
server.csr  
Certificate Request:  
Data:  
  Version: 1 (0x0)  
  Subject: C = PK, ST = Sind, L = Lahore, O = PTCL, OU = Mian Muhammad, CN  
= Arif Alvi, emailAddress = qte11456@cdfaq.com  
  Subject Public Key Info:  
    Public Key Algorithm: rsaEncryption  
    Public-Key: (2048 bit)  
    Modulus:  
      00:e0:b0:ea:8a:e8:5a:02:47:68:cf:65:a8:16:f2:  
      1e:d8:bf:e2:22:ea:0c:93:ee:ce:40:c2:d4:7f:20:  
      bb:eb:b5:17:bf:2c:78:f3:3f:5b:73:8f:84:9c:97:  
      59:b3:78:3c:e5:e3:fb:a9:76:ad:f9:9d:47:03:25:  
      57:76:04:a4:3e:06:40:27:50:21:ac:c2:de:41:e1:  
      77:53:3c:b2:19:0c:fd:ef:da:9b:f0:5c:7d:3a:72:  
      2e:68:01:fc:cc:fb:a3:f6:b5:2f:c0:a8:5a:b8:f2:  
      62:dd:80:a3:67:cf:d8:4a:ea:c0:b7:0e:c7:79:2f:  
      d2:3e:21:98:83:81:85:11:20:99:8d:9a:e1:51:d9:  
      1f:0b:df:8f:9f:54:3b:da:90:66:a0:9a:23:d6:4a:  
      05:61:35:1d:fc:16:b4:78:09:f5:80:0b:d9:89:e6:  
      23:f2:df:7c:bd:28:b2:a5:3e:3b:c8:46:72:e7:61:  
      2d:44:dd:a3:f3:51:a2:a6:f7:7e:38:76:f9:8e:99:  
      8e:0b:96:89:1c:42:fd:f3:ec:00:88:4c:65:90:bc:
```

Ahora debe enviar esta Solicitud de firma de certificado (CSR) a una Autoridad de certificación (CA) para que la firme.

Una vez que se haya firmado el CSR, tendrá un Certificado real, que puede ser utilizado por Apache.

Puede tener una CSR firmada por una CA comercial o puede crear su propia CA para firmarla.

Las CA comerciales generalmente le piden que publique la CSR en un formulario web, hay que pagar por la firma y luego enviar un certificado firmado.

Puede almacenarse en un archivo **server.crt**.

Una vez que se haya firmado su CSR, puede ver los detalles del Certificado de la siguiente manera:

\$ openssl x509 -noout -text -in server.crt

```
41:0d:48:14
estudiante@DAW1:~/Downloads/web/paginaWebServidor$ openssl x509 -noout -text -in
server.crt
```

Ahora debería tener dos archivos: server.key y server.crt. Estos se pueden utilizar de la siguiente manera en su archivo httpd.conf:

**SSLCertificateFile "/ruta/a/este/servidor.crt" SSLCertificateKeyFile
"/ruta/a/este/servidor.clave"**

El archivo **server.csr** ya no es necesario.

Crear su propia CA

Para obtener detalles sobre cómo crear su propia CA y utilizarla para firmar una CSR:

La respuesta corta es usar el script CA.sho CA.pl proporcionado por **OpenSSL**.

A menos que tenga una buena razón para no hacerlo, debe usarlos con preferencia.

Si no puede, puede crear un certificado autofirmado de la siguiente manera:

Cree una clave privada RSA para su servidor (estará cifrada con Triple-DES y formateada con PEM):

\$ openssl genrsa -des3 -out server.key 2048

Despliegue de aplicaciones web.
Román Millán Díaz

Copia de seguridad de este archivo **server.key** y la frase de contraseña que ingresó en una ubicación segura.

Puede ver los detalles de esta clave privada RSA:

```
$ openssl rsa -noout -text -in server.key
```

Si es necesario, también puede crear una versión PEM descifrada (no recomendada) de esta clave privada RSA:

```
$ openssl rsa -in server.key -out server.key.unsecure
```

Cree un certificado autofirmado con la clave RSA que acaba de crear (la salida tendrá formato PEM):

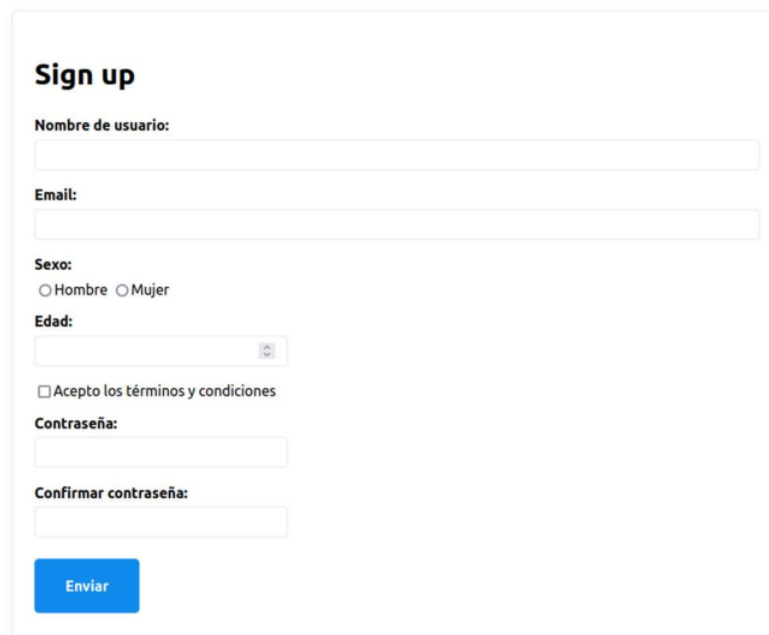
```
$ openssl req -new -x509 -nodes -sha1 -days 365 -key server.key -out server.crt  
-extensions usr_cert
```

Esto firma la CSR del servidor y da como resultado un archivo **server.crt**.

Puede ver los detalles de este Certificado:

```
$ openssl x509 -noout -text -in server.crt
```

Una vez terminado todos los pasos anteriores, debe aparecerle de la siguiente manera.
Buscando por <https://localhost>



Sign up

Nombre de usuario:

Email:

Sexo:
☐ Hombre ☐ Mujer

Edad:

☐ Acepto los términos y condiciones

Contraseña:

Confirmar contraseña: