

Configuración de un certificado SSL en la Release2

Introducción

El protocolo SSL (Secure Socket Layer) está destinado a encriptar de manera segura los datos intercambiados entre dos máquinas. En su servidor dedicado, puede ser útil utilizar SSL con el fin de proteger datos sensibles.

Encontrará más información sobre el funcionamiento de SSL en esta página.

Procedimiento

Atención : Este procedimiento está basado en la Release 2 de OVH.

Esta guía le ayudará a crear su propio certificado SSL con sus datos de manera gratuita y posteriormente instalar este nuevo certificado en los servicios que desee de su servidor.

Igualmente si ha comprado un certificado de pago y dispone de una pareja de 2 ficheros (KEY y CRT) válida, podrá utilizar esta guía para instalar los dos ficheros de su certificado en los servicios deseados.

Lista de los ficheros de certificados en su servidor

Por defecto existen una serie de certificados SSL en su servidor que aseguran la comunicación mediante una conexión segura en diferentes servicios.

Así, utilizando estos certificados, tenemos HTTPS, SFTP, SSMTP, etc...

Algunos de los certificados que vienen instalados por defecto de estos servicios, son simples certificados de prueba y muchos de ellos son incorrectos, están incompletos o caducados.

Por defecto los ficheros en su máquina son los siguientes :

Certificado de Apache HTTPS :

- /etc/httpd/ssl.crt/server.crt
- /etc/httpd/ssl.key/server.key

Certificado para correo SSMTP :

- /var/qmail/control/servercert.pem

Certificados para SMYSQL

- /usr/share/mysql/mysql-test/client-cert.pem
- /usr/share/mysql/mysql-test/client-key.pem
- /usr/share/mysql/mysql-test/server-cert.pem
- /usr/share/mysql/mysql-test/server-key.pem

Certificado para Webmin :

- /etc/webmin/miniserv.pem

NOTA : La lista de los servicios y la localización de los archivos está basada en la distribución **Release 2** de OVH. Si dispone de una distribución distinta los servicios y la localización pueden variar

Generación de un certificado autofirmado

La generación de un certificado autofirmado, es una tarea sencilla que puede realizarse en 4 sencillos pasos.

Paso 1. Clave del servidor

Nos conectamos en SSH como root y creamos un directorio para crear el certificado.

```
# mkdir certificado
# cd certificado
certificado #
```

A continuación ejecutamos una orden para crear una clave para el servidor

```
certificado # openssl genrsa -out server.key 1024
```

Generating RSA private key, 1024 bit long modulus

```
.....++++++
.....++++++
e is 65537 (0x10001)
```

Este comando creará el fichero **server.key**

Paso 2. Creación del certificado

A continuación creamos el certificado con los datos de nuestra empresa y nuestra página web.

Los datos presentados a continuación son un mero ejemplo :

```
certificado # openssl req -new -key server.key -out server.csr
```

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]:Madrid
Locality Name (eg, city) [ ]:Madrid
Organization Name (eg, company) [Internet Widgits Pty Ltd]:OVH HISPANO
Organizational Unit Name (eg, section) [ ]:Soporte
Common Name (eg, YOUR name) [ ]:www.ovh.es
Email Address [ ]:soporte@ovh.es
```

Please enter the following 'extra' attributes to be sent with your certificate request

A challenge password []:

An optional company name []:**OVH**

Este comando creará el fichero **server.csr**

Nota : Si quiere crear un certificado para todos los dominios del servidor, puede rellenar el **Common Name** con un asterisco (*) en lugar de un sitio web concreto (**www.ovh.es**).

Paso 3. Firma del certificado

Para firmar un certificado necesitamos tener un número de serie con los certificados que vamos firmando. Si es nuestro primer certificado, el número de serie debería ser 1, aunque no es obligatorio.

Elegimos un número al azar y completamos la línea :

```
certificado # openssl x509 -req -days 365 -set_serial [número_de_serie]
-in server.csr -signkey server.key -out server.crt
```

Signature ok

```
subject=/C=ES/ST=Madrid/L=Madrid/O=OVH
```

HISpano/OU=Soporte/CN=www.ovh.es/emailAddress=soporte@ovh.es
Getting Private key

Paso 4. Finalización

Combinamos el certificado y la clave en un solo archivo .pem

```
certificado # cat server.key server.crt > server.pem
```

Al final cada certificado (en nuestro caso con el nombre *server*) debe tener cuatro ficheros creados :

```
server.crt server.csr server.key server.pem
```

Si queréis ver los detalles de vuestro certificado podéis ejecutar esta orden:

```
certificado # openssl x509 -in server.pem -noout -text
```

Certificate:

```
Data:  
Version: 1 (0x0)  
Serial Number: 1 (0x1)  
Signature Algorithm: sha1WithRSAEncryption  
[...]
```

Utilización de un certificado externo

Si ha comprado o bien tiene ya un certificado que desee utilizar, sólo debe copiar los ficheros en una carpeta.

Nos conectamos en SSH como root y creamos un directorio para crear el certificado.

```
# mkdir certificado  
# cd certificado  
certificado #
```

A continuación copiamos los ficheros del certificado en la carpeta :

```
certificado # scp servidor_origen/server.key server.key
```

```
certificado # scp servidor_origen/server.crt server.crt
```

Finalmente combinamos el certificado y la clave en un solo archivo .pem

```
certificado # cat server.key server.crt > server.pem
```

Al final, el certificado (en nuestro caso con el nombre *server*) debe tener tres ficheros :

```
server.crt server.key server.pem
```

Si queréis ver los detalles de vuestro certificado podéis ejecutar esta orden:

```
certificado # openssl x509 -in server.pem -noout -text
```

Certificate:

Data:

Version: 3 (0x2)

Serial Number: 1051699855 (0x3eafaa8f)

Signature Algorithm: sha1WithRSAEncryption

[...]

Copia de los archivos en cada directorio según el servicio

A continuación vamos a copiar el certificado *server* que hemos creado para el servicio elegido.

En cada caso, posteriormente a la copia, debemos reiniciar los servicios para que los cambios surtan efecto.

Nota : Es conveniente que creéis diferentes certificados con un Common Name (CN) diferente para cada servicio. Seguid las instrucciones del mismo modo cambiando el nombre *server* por el nombre que decidáis en cada comando.

Certificado de Apache HTTPS

Ejecutamos los comandos :

```
certificado # mv /etc/httpd/ssl.crt/server.crt
/etc/httpd/ssl.crt/server.crt.orig
certificado # cp server.crt /etc/httpd/ssl.crt/server.crt

certificado # mv /etc/httpd/ssl.key/server.key
/etc/httpd/ssl.key/server.key.orig
certificado # cp server.key /etc/httpd/ssl.key/server.key
```

Certificado para correo SSMTTP

Ejecutamos el comando :

```
certificado # mv /var/qmail/control/servercert.pem
/var/qmail/control/servercert.pem.orig
certificado # cp server.pem /var/qmail/control/servercert.pem
```

Otros certificados

Los certificados para Webmin y SMYSQL llevan su propia CA (Autoridad de certificado) para firmarlas y no es necesario actualizarlas.

Si necesita actualizar los certificados de estos servicios deberá firmar el certificado con las claves de CA siguientes :

CA para Webmin :

- /etc/webmin/acl/ca.pem

CA para MySQL :

- /usr/share/mysql/mysql-test/cacert.pem

Más información

- : `ServidorSSL` :: Configuración de un certificado SSL en la Release1
- : `PleskCertificadosSSL` :: Configuración de un certificado SSL en Plesk

- : `SslRelease2` :: Configuración de un certificado SSL de OVH en la Release2
- : `SslPlesk` :: Configuración de un certificado SSL de OVH en Plesk