COMO GENERAR CERTIFICADOS SSL

Esta sección explica cómo crear un certificado autofirmado. Para que los certificados queden ya en el lugar adecuado dirigirse a la carpeta donde se los guardará.

Generar una llave para el servidor:

openssl genrsa -des3 -out server.key 4096

Este comando pedirá una serie de cosas (país, provincia, etc..). Es importante poner el “Common Name (eg, YOUR name)” el nombre del servidor o, si no lo tuviera, la IP del mismo.

Los valores por defecto de las preguntas se guardan en /etc/ssl/openssl.cnf. Por ello, es bueno modificarlos allí si son varios los certificados a crear.

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

Luego firmar el requerimiento de firmado. El ejemplo genera un certificado firmado válido por 365 días.

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

Ahora crear una versión de server.key que no necesite contraseña.

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

mv server.key server.key.secure

mv server.key.insecure server.key

Generar una propia CA (Autoridad Certificante):

El Common Name (CN) de la CA y del certificado del servidor no deben coincidir o habrá una colisión de nombres y aparecerán errores más adelante. En el siguiente paso deben proveerse los datos de la CA y más adelante los del servidor.

CA:

Common Name (CN): CA-NetStorming

Organization (O): NetStorming

Organizational Unit (OU): Redes

<code>

Server:

Common Name (CN): www.netstorming.com.ar

Organization (O): NetStorming

Organizational Unit (OU): Redes

Generar el certificado de la CA:

openssl genrsa -des3 -out ca.key 4096

openssl req -new -x509 -days 365 -key ca.key -out ca.crt

*Generar una llave para el servidor y un requerimiento de firmado (csr): **Al elegir el nombre del servidor “Common Name (eg, YOUR name)” setear con el nombre que resuelva el dns o, en todo caso, con la IP del equipo. <code> openssl genrsa -des3 -out server.key 4096 openssl req -new -key server.key -out server.csr </code> Ahora firmar el requerimiento de firmado del certificado (csr) con la CA recién creada. Notar que se firma por 365 días, al cabo de los cuales hay que volver a firmarlo. También notar que se setea el serial a 01. Es importante que cada vez que se haga esto se cambie el serial, para los navegadores que tengan cacheado el certificado. <code> openssl x509 -req -days 365 -in server.csr -CA ca.crt -CAkey ca.key -set_serial 01 -out server.crt </code> Para examinar los componentes en caso de que se desee: <code> openssl rsa -noout -text -in server.key openssl req -noout -text -in server.csr openssl rsa -noout -text -in ca.key openssl x509 -noout -text -in ca.crt </code>* Generar una clave que no obligue al servidor a pedir la contraseña:**

Aquí creamos una versión insegura de la clave del servidor. La clave insegura se usará para iniciar el servicio y no requerir la password cada vez que esto ocurra. Lógicamente, como no pide la password la misma es almacenada lo cual hace que si alguien tiene acceso al archivo podria desencriptar toda la transmisión, por lo cual lo mejor es hacer un chown root:root y chmod 000 de los archivos de claves.

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

mv server.key server.key.secure

mv server.key.insecure server.key

Archivos obtenidos

Si se siguió el primer camino al momento debería haber 4 archivos:

server.crt: el certificado autofirmado.

server.csr: requerimiento de firmado del certificado.

server.key: la clave privada del servidor que no requiere password.

server.key.secure: la clave privada del servidor que requiere password.

Si se siguió el segundo camino debería haber dos archivos adicionales:

ca.crt: el certificado propio de la CA.

ca.key: la llave que usa la CA para firmar los certificados.

Los archivos de la CA son importantes para firmar nuevos certificados mientras la misma permanezca vigente.

Ahora sólo resta configurar el servicio para usar SSL con los certificados recién creados.

FUENTE: ***Departamento de Infraestructura - SENATICs***