Bisoños Usuarios de GNU/Linux de Mallorca y Alrededores | Bergantells Usuaris de GNU/Linux de Mallorca i Afegitons

Certificado Digitales con OpenSSL I (80364 lectures)

Per **Javier Garcia**, ***capi*** (http://www.shadowsland.com)

Creado el 23/02/2006 01:51 modificado el 27/03/2006 03:03

Hola,

En mi web[1] he publicado varios artículos sobre como montar certificados digitales con OpenSSL. Y me he animado a publicarlo también aquí para que llegue a un mayor número de personas.

En este artículo explico como instalar un certificado digital en el servidor apache para que podaís conectaros a vuestro servidor mediante https. Para ello creo una CA y su correspondiente certificado y hago la instalación en apache2.

Espero que os guste,
Capitán Linux.

Para empezar yo trabajo con la distribución debian por lo que la descarga de la aplicación openssl la haré desde sus repositorios. Si alguien tiene otra distribución solo le diré que para probar los ejemplos necesitará el openssl, el cual se lo tendrá que bajar de la web del proyecto o de los paquetes de su distribución de linux.

Bueno nos descargamos el paquete openssl abriendo un terminal y utilizando el maravilloso apt teclamos los siguientes comandos:

apt-get update

Con el apt-get update obtenemos las listas de las ultimas versiones de paquetes. Y ahora hacemos un:

apt-cache search openssl
Con el apt-cache search buscamos los paquetes que tengan algo llamado openssl. De la lista que nos sale , el que mas me convence es el openssl (que es lo que yo busco) , por tanto lo instalamos tecleando lo siguiente:

apt-get install openssl

Ya tenemos la herramienta openssl con la cual podremos crear nuestros certificados digitales , de forma gratuita. Comprar un certificado cuesta mucho dinero, por si no lo sabéis. ;)

Para poder crear un certificado primero tenemos que tener una CA (Autoridad Certificadora). Que es la que nos valida y confirma que nuestro certificado es valido.

Pues bien vamos a crear una CA. Para esto desde nuestra terminal (os aconsejo crear un directorio y estar dentro de este ya que estos comandos generan ficheros ;)) hacemos lo siguiente :

openssl req -x509 -newkey rsa:2048 -keyout cakey.pem -days 3650 -out cacert.pem

Con este comando creamos un CA para certificados X509 con algoritmo de encriptación rsa de 2048 bytes. Con el -keyout le indicamos que la clave privada de nuestra CA se almacene en el fichero cakey.pem y la clave publica -out en el cacert.pem.
Seguidamente nos pedirá un password para nuestra CA y se lo damos. (Apuntarlo o memorizarlo ya que sirve para los

próximos pasos). También nos pedirá una serie de datos por ejemplo país, nombre de empresa, que nos identifica como CA.
Observar que le he añadido el parámetro -days con 3650, de esta manera indicamos que la CA no expire en 10 años.

Ahora vamos a crear un certificado digital , es decir con nuestra CA creada vamos a crearnos un certificado. Podremos hacer tantos como queramos los pasos son los mismos, a partir de ahora:

Primero generamos la clave privada del que sera nuestro certificado digital:

openssl genrsa -des3 -out serv-priv.pem -passout pass:mipassword 2048

Con esto generamos la clave privada la cual tendrá un algoritmo de cifrado triple des (-des3) de 2048 y se almacenara en el fichero (-out) serv-priv.pem y con el comando -passout pass: indicamos la passfrase para nuestra clave privada yo le he puesto mipassword :)

Ahora vamos a generar la petición del certificado, antes de hacer un certificado , hay que hacer una petición donde se define el propietario del certificado.

openssl req -new -subj "/DC=shadowsland.com/OU=com/CN=shadowsland" -key serv-priv.pem -passin pass:mipassword -out petic-certificado-serv.pem

Observamos que le indicamos (req -new) con lo que indicamos que es una petición con el parametro -subj le indicamos a quien pertenece el certificado, para ello ponemos entre comillas cada uno de los apartados que identifican separados por / , observar que yo he usado shadowsland, vosotros ponéis vuestro servidor o lo que sea. A la petición le indicamos que usara la clave privada que hemos hecho en el anterior comando -key serv-priv.pem y le indicamos el password que usemos en el anterior (-passing pass):mipassword y como salida (-out)le indicamos que genere el siguiente fichero petic-certificado-serv.perm

Finalmente ya podemos emitir el certificado. Para definir las características de un certificado openssl dispone del directorio /etc/ssl donde hay un fichero openssl.cnf con lo que podemos definir las características. Pero hay una forma mas simple de darle estas características, mediante la generación de un fichero de configuración por parte nuestra. Por tanto con VI o con nuestro editor favorito generamos un fichero que contenga lo siguiente:

basicConstraints = critical,CA:FALSE
extendedKeyUsage = serverAuth

Y lo guardamos por ejemplo con el nombre config1.txt , en el fichero le indicamos el basicConstraints =critical,CA:FALSE para que cumpla con el X509v3 y con la RFC3280, paja mental vamos. Lo ponemos y punto. Y también le indicamos que el certificado servira para un servidor con lo siguiente extendedKeyUsage=serverAuth, por ejemplo un servidor web que es lo que queremos certificar.

Y ahora con el ficherito hecho con la configuración hacemos el certificadillo:

openssl x509 -CA cacert.pem -CAkey cakey.pem -req -in petic-certificado-serv.pem -days 3650 -extfile config1.txt -sha1 -CAcreateserial -out servidor-cert.pem

Indicamos que sera un certificado del tipo x509 cuya CA (-CA) esta definida en el fichero cacert.pem (lo hemos hecho lo primero) y que usa como clave privada (-CAkey) el fichero cakey.pem y que el certificado a generar tendrá las especificaciones definidas en el apartado anterior (-req -in) las cuales están en el fichero de petición petic-certificado-serv.pem.
El certificado tendrá una validez de diez años (-days 3650) no valdría pasta esto ni na ;)

Y le indicamos que el certificado es para un servidor, como esto lo tenemos en nuestro fichero de configuración se lo indicamos poniendo -extfile y nuestro fichero config1.txt y utilizaremos un algoritmo de cifrado SHA (-sha1).

Luego como es nuestro primer certificado le indicamos que la Ca lo numere con lo cual le pondrá 2 , ya que el 1 es la CA -CAcreateserial y finalmente le decimos donde generar el certificado (-out) en el fichero servido-cert.pem

Una vez lanzado el comando nos pedirá el password de la CA que lo emite y el fichero se generará.

Ahora os listo los fichero generados solo los importantes que hemos generado , el resto se puede eliminar :

Por parte de la CA:

cacert.pem

cacert.srl

cakey.pem

Por parte del certificado :

servidor-cert.pem

serv-priv.pem
Utilidad :

config1.txt
Vale y todo esto para que sirve, pues fácil para poder crear un sitio certificado bajo SSL (https) con apache. Para ello tenéis que copiar los ficheros servidor-cert.pem y servidor-priv.pem en /etc/apache2/ssl si no disponéis del directorio ssl crearlo solo sirve para guardar esto en algún lado y que apache lo vea. Al apache2 hay que indicarle que también escuche por el puerto 443 que es el del Secure Socket Layer (ssl) para eso se modifica el fichero **/etc/apache2/ports.conf** para que ponga:

Listen 80
Listen 443

Ahora tenemos que decirle apache que soporte el ssl para ello instalamos el modulo ssl: apt-get install libapache-mod-ssl
Una vez instalado esto le decimos a apache que cargue el modulo, tecleando lo siguiente :

a2enmod ssl

Y con esto ya solo nos queda configurar nuestro fichero de sites-availables donde tenemos nuestra configuración para poder acceder a nuestro site mediante https.
editamos el fichero **/etc/apache2/sites-availables/default** y creamos el virtual host adecuado para el puerto 443 :

```
NameVirtualHost *:443
<VirtualHost *:443>
ServerAdmin webmaster@localhost
DocumentRoot /var/local/mipagina
SSLEngine on
SSLCertificateFile /etc/apache2/ssl/servidor-cert.pem
SSLCertificateKeyFile /etc/apache2/ssl/serv-priv.pem
ServerName mipagina.midominio.com
<Directory "/var/local/mipagina">
  Options Indexes FollowSymLinks MultiViews
  AllowOverride None
  Order allow,deny
  allow from all
</Directory>
</VirtualHost>
```

Poniendo en el DocumentRoot el lugar donde tenemos nuestra pagina. Le indicamos que utilizamos certificados (SSLEngine on) y le decimos donde tiene que que leer el certificado (SSLCertificateFile) y la clave privada de este

(SSLCertificateKeyFile). Si no os queda algo claro de este tema consultar la instalación de apache de la pagina de jakarta.

Finalmente despues de guardar el fichero default nos vamos al **/etc/apache2/sites-enabled** y hacemos un vinculo simbólico al fichero default del sites-availables y le llamamos default-000
ln -s ../sites-availables/default default-000

Ahora solo nos queda parar y arrancar nuestro apache

/etc/init.d/apache2 stop
/etc/init.d/apache2 start

Observareis que os pide password , el cual es el password de vuestro certificado. Se lo ponéis y el servidor pim pam pim pam se arranca. Como lo del password puede ser un problema ya que ahora si la maquina se arranca y cuando arranque el apache, se quedara parado hasta metáis el password. Pero ante todos los grandes problemas hay una solución, aunque en este caso es un poco trapera ;)

Y es ponerle en el fichero /etc/sites-availables/default una directiva (fuera de los virtual host) como la siguiente :

SSLPassPhraseDialog exec:/etc/apache2/generadorclave

Donde generadorclave es un fichero bash que creamos nosotros para que se invoque cuando apache arranque y le proporcione la clave. El contenido del fichero podría ser el siguiente :

#!/bin/sh
echo mipassword

Observar que simplemente es un echo con el password ;) Ahora solo le tenéis que dar permiso de ejecucion y fiesta!

chmod 700 generadorclave

Volveis a generar lo del vinculo simbolico y volveis a rearrancar apache y ya podéis probar vuestro site poniendo en el navegador por ejemplo https://mipagina.midominio.com

Y ya teneis la faenita hecha. ;)

Lista de enlaces de este artículo:

1. http://www.shadowsland.com

E-mail del autor: capitanlinux _ARROBA_ gmail.com