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

Certificados Digitales con OpenSSL II (24278 lectures)

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

Creado el 27/03/2006 02:02 modificado el 27/03/2006 03:14

Dado la aceptación que ha tenido de mi artículo sobre Certificados Digitales I, he decidido publicar en Bulma la segunda parte de este , el cual lo tenía publicado en mi web[1], para que así llegue a un mayor número de personas.

En el artí−culo anterior trataba sobre como montar un certificado sobre un servidor web, para tener un confidencialidad de datos. Esto tiene muchos usos por ejemplo:montar nuestro propio web-mail y tener la garantía que mediante un sniffer no podrán leer nuestros e-mails mientras son enviados. También lo podíamos usar en algo que necesite confidencialidad por ejemplo un Terminal Punto de Venta (TPV).

Pero ahora voy a ir un poco más alla, voy a explicar como montar un servidor bajo SSL que exija a los clientes tener un certificado para poder acceder a este. De tal forma que sólo aquellas personas que tengan dichos certificados clientes puedan acceder al servidor.

Es altamente recomendable entender el primer artí−culo, para poder seguir esta segunda parte.

Espero que os guste,
El Capi.

Para empezar, hacemos un poco de memoria del artí−culo anterior:

Tenemos los ficheros que generamos al crear nuestra propia CA, los cuales eran los siguientes: **cacert.pem, cacert.srl, cakey.pem**

Importante recordar cual era el password que le pusiste a tu CA ya que se te pedirá en uno de los siguiente pasos. Sí no te acuerdas repite el ejercicio anterior... (Ya os dije que lo apuntarais ;))

También habíamos creado un certificado para nuestro servidor, el cual se componía de un fichero que era el certificado y otro que era su clave privada: servidor-cert.pem, serv-priv.pem

Os recuerdo que yo en el ejemplo anterior dije que mi certificado servidor-cert.pem tenia como passphrase (password) mipassword y este certificado lo habíamos puesto en apache para que pudiéramos conectarnos vía https (ssl) a nuestro site.

Pues bien bajo este punto de partida, crearemos nuestro primer certificado cliente, es decir este es el certificado que le tendremos que pasar a la persona que queramos que se pueda conectar a nuestra web protegida por https. Es necesario crear un directorio ya que todos estos comandos generan ficheros y copiar en este directorio los fichero que habíamos generado al crear nuestra CA. :)

Primero generamos la clave privada del cliente:

openssl genrsa -des3 -passout pass:passcliente1 -out client-priv.pem 2048

Básicamente es igual que en anterior articulo, 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) client-priv.pem y con el comando -passout pass:

indicamos la passphrase para nuestra clave privada yo le he puesto passcliente1

Ahora generamos la petición del certificado:

openssl req -new -key client-priv.pem -passin pass:passcliente1 -subj "/DC=shadowsland.com/OU=com/CN=Filemon" -out petic-cert-client.pem

Observamos que le indicamos (req -new) con lo que indicamos que es una petición con el parámetro -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 poned lo que pusisteis en vuestro certificado servidor o lo que sea. Observar que en el CN he indicado que la persona propietaria de este certificado cliente que se llama Filemon, poner el nombre que queráis, si queréis mas parametros miraros la ayuda del openssl sobre el subject. A la petición le indicamos que usara la clave privada que hemos hecho en el anterior comando -key client-priv.pem y le indicamos el password que usemos en el anterior:(-passing pass):passcliente1.
Y como salida (-out)le indicamos que genere el siguiente fichero petic-cert-client.perm.

Y ahora finalmente ya podemos emitir el certificado, para ello es necesario especificarle en el fichero de configuración **/etc/ssl/openssl.cnf** de como será nuestro certificado. Pero como en el caso anterior esta configuración se la podemos indicar mediante un fichero de configuración externo:

Así que pillamos un buen editor, por ejemplo el maravilloso VI ;))
Una vez en vuestro editor favorito (ojala que no usen el notepad! XD) tecleáis las siguientes lineas :

basicConstraints = critical,CA:FALSE
extendedKeyUsage = clientAuth

Y lo guardamos por ejemplo con el nombre config2.txt , en el fichero le indicamos el basicConstraints=critical,CA:FALSE para que cumpla con el X509v3 y con la RFC3280, la misma paja mental del anterior artículo. Lo ponemos y punto. Y también le indicamos que el certificado servira para acceso a un servidor por lo que será una auntenficación cliente y esto se lo indicamos con lo siguiente **extendedKeyUsage=clientAuth.**

Ahora que ya lo tenemos todo emitimos el certificado cliente :

openssl x509 -CA cacert.pem -CAkey cakey.pem -req -in petic-cert-client.pem -set_serial 3 -days 365 -extfile config2.txt -sha1 -out client-cert.pem

Veamos que estamos haciendo, le indicamos que será un certificado del tipo x509 cuya CA (-CA) está definida en el fichero cacert.pem (lo hemos hecho en el artí−culo anterior) y que usa como clave privada (-CAkey) el fichero cakey.pem y que el certificado a generar tendrán las especificaciones definidas en el apartado anterior (-req -in) las cuales están en el fichero de petición petic-cert-client.pem.
El certificado tendrá una validez de un año (-days 365) y que pidan otro cuando se les expire ;) Y le indicamos que el certificado es para un cliente, como esto lo tenemos en nuestro fichero de configuración se lo indicamos poniendo -extfile y nuestro fichero config2.txt y utilizaremos un algoritmo de cifrado SHA (-sha1).

Luego como nuestra CA tenía el número 1 y nuestro primer certificado (el del servidor del artí−culo anterior) era el certificado 2 este nuevo certificado sera el 3 y los proximos certificados a generar serán el 4,5,6...
Esto se le indica mediante el parámetro -set_serial. Si esta numeración no se pone correctamente tendremos un problema con nuestro certificado.
Finalmente le decimos donde generar el certificado (-out) en el fichero cliente-cert.pem.
Al ejecutar esta línea nos pedirá el password que le pusimos a nuestra CA y al ponérselo obtendremos el fichero client-cert.pem que es nuestro certificado.

Y señoras y señores aquí− tenemos nuestro primer certificado cliente :)

Bueno veamos que ficheros útiles tenemos:

Por parte de la CA tenemos :

cacert.pem, cacert.srl, cakey.pem

Por parte de nuestro certificado servidor :

servidor-cert.pem, serv-priv.pem

Por parte de nuestro certificado cliente :

client-cert.pem, client-priv.pem

Y como utilidad el fichero :

config2.txt

El fichero petición de certificado **petic-cert-client.pem**, lo podéis eliminar ya que no nos sirve para nada.

Bien y ahora pues vamos a configurar nuestro apache para que nuestro certificado **servidor-cert.pem** pida certificados cliente :)

Primero de todo copiamos los ficheros por parte de la CA en el directorio de apache2 donde copiemos nuestro certificado en el artículo anterior. Es decir en el directorio **/etc/apache2/ssl**

Ahora editamos el fichero **/etc/apache2/sites-availables/default** y creamos o añadimos al virtual host adecuado para el**puerto 443**, lo siguiente:

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
SSLCACertificateFile /etc/apache2/ssl/cacert.pem
SSLVerifyClient require
ServerName mipagina.midominio.com
<Directory "/var/local/mipagina">
Options Indexes FollowSymLinks MultiViews
AllowOverride None
Order allow,deny
allow from all
&lt/Directory>
&lt/VirtualHost>

Observamos que le añadimos las siguientes nuevas lí−neas con diferéncia al artí−culo anterior
SSLCACertificateFile /etc/apache2/ssl/cacert.pem
SSLVerifyClient require

Con **SSLCACertificateFile**, le indicamos donde tenemos nuestra CA y con SSLVerifyClient require le indicamos que el servidor solicite un certificado al cliente.

Guardamos el fichero /etc/apache2/sites-availables/default y creamos un vinculo simbólico de este en el /etc/apache2/sites-enabled , es decir:

ln -s ../sites-availables/default default-000

Y ahora finalmente ya podemos re-enchegar nuestro apache para que pille la nueva configuración:

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

Y ya tenemos nuestro servidor el cual a la hora de conectarnos mediante https a este nos pedirá un certificado y si no lo tenemos no nos abrira la página. Probar amigos probar... jejeje

Bien ya habéis visto que no os deja entrar en la página, pues lo mismo le pasara a cualquier persona que entre sin tener certificado cliente mágico.

Ahora vamos a coger nuestro certificado cliente y se lo vamos a incorporar a nuestro navegador, para ello necesitamos crear con nuestro certificado un fichero comprimido en formato **pkcs12** que el navegador es el que pilla.

Y esto se realiza de la siguiente manera:

openssl pkcs12 -export -in client-cert.pem -inkey client-priv.pem -certfile cacert.pem -out cert-pck12.p12

Al ejecutar esto nos pedirá la passphrase del certificado cliente, en mi caso puse passcliente1. (Véase cuando generamos el client-priv.pem al principio del artículo)

También os pide un Export Password, que es un password que tengáis que poner para comprimir el archivo. Luego al importar el certificado en un navegador os lo pedirá.

Luego os pide que verifiquéis el Export Password, es decir que lo volváis a poner para comprobar que lo pusisteis bien.

Y ahora para ponerlo, abrid el mozilla firefox (por ejemplo) y os vais al ménu Edit. Allí seleccionáis Preferences y dentro de la ventana que os sale os situáis sobre Advanced y con el scroll buscáis hasta encontrar Manage Certificate y le dáis al import y ponéis el ficherito **cert-pck12.p12** que es vuestro certificado cliente.

Le ponéis el password que habéis puesto en la exportación y ale ya tenéis el navegador con vuestro certificado cliente cargado.

Y cuando intentéis ver vuestro site ya lo podréis ver. :)

Lista de enlaces de este artículo:

1. http://www.shadowsland.com

E-mail del autor: capitanlinux _ARROBA_ gmail.com