Creación y administración de certificados de seguridad mediante OpenSSL.

Autor: Enrique V. Bonet Esteban

Introducción.

En ocasiones, servicios que estudiaremos con posterioridad como un servidor de HTTP, un servidor de correo como SMTP, un servidor de entrega final de correo POP3 o IMAP, etc., requieren de una seguridad adicional que no proporcionan de forma normal.

Así, por ejemplo, si poseemos una página web que solicita información confidencial, como pueden ser datos de una tarjeta bancaria, o un servidor de correo ante el cual el usuario debe identificarse mediante su nombre de usuario y contraseña, debemos exigir a dichos servicios una seguridad adicional, consistente en que los datos no circulen por Internet en formato de texto plano (sin cifrar), accesible a cualquier persona de Internet que pueda interceptar la comunicación de los datos, sino que solo puedan ser accesibles por el cliente y el servidor.

Para ello, es necesario introducir las comunicaciones en formato cifrado y una mutua autentificación entre los clientes y servidores. Este proceso pasa por utilizar un sistema de encriptación y por tanto, por generar, en primer lugar, un sistema criptográfico adecuado.

De forma general, existen dos tipos de sistemas criptográficos:

- Convencionales o simétricos, en los cuales ambas partes de la transmisión tienen la misma clave, la cual usan para codificar o decodificar la transmisión del otro

- Criptografía pública o asimétrica, en la cual existen dos claves, una pública y otra privada, de forma que la clave pública es accesible a todo el mundo y la clave privada es guardada en secreto. Los datos codificados con la clave pública solo pueden ser decodificados con la clave privada y viceversa.

La configuración de servidores seguros utiliza criptografía de clave pública, de forma que se crean un par de claves pública y privada[1]. La clave pública es enviada a los clientes, siendo utilizada por estos tanto para decodificar los mensajes cifrados por el servidor con su clave privada, como para cifrar los mensajes que serán enviados al servidor, el cual los decodificará utilizando su clave privada.

Creación de un par de claves pública y privada.

Como hemos visto, el primer paso que es necesario dar para crear un método de comunicación seguro es poseer un par de claves pública y privada, claves que deberán

[1] Una descripción algo más detallada de la criptografía de clave pública y privada puede encontrarse en "Seguridad Práctica en UNIX e Internet (2ª edición)" de Simson Garfinkel y Gene Spafford, publicado en la editorial O'Reilly.

ser creadas inicialmente y que deberán ser sustituidas bien por motivos de seguridad (robo de las mismas, etc.), como por motivos temporales (caducidad de las mismas, por ejemplo).

Para generar un par de claves pública/privada nuevas mediante OpenSSL[2] debemos ejecutar el comando:

```
openssl genrsa [-des|-des3|-idea] [<tamaño>]
```

Que generará un par de claves pública/privada, siendo la clave privada del tamaño en bits especificado por <*tamaño*>. Si <*tamaño*> no se especifica, el valor por defecto es de 512 bits[3]. Si se utiliza el parámetro *-des*, *-des3* o *-idea*, la clave privada se cifrará utilizando el algoritmo DES, 3DES o IDEA, para lo que se solicitará la introducción de una contraseña. Por ejemplo, si ejecutamos el comando:

```
openssl genrsa -des3 2048 > clave.key
```

El programa generará un par de claves pública/privada de una longitud de 2048 bits, solicitando la introducción de una contraseña[4] y guardando la información en un fichero llamado *clave.key*[5]. Cada vez que el servicio que utilice esta clave sea inicializado, se requerirá la introducción de la contraseña antes de habilitar el servicio.

Si no se desea introducir una contraseña cada vez que se inicia el servicio, bien porque esto sucede de forma frecuente, bien por comodidad, o por cualquier otra causa, se puede generar una clave, cuyo uso no requerirá la introducción de una contraseña, mediante el comando:

```
openssl genrsa 2048 > clave.key
```

Ahora veremos que el programa no solicita la introducción de una contraseña y que, por tanto, ninguna contraseña será requerida al iniciar el uso de la clave por el servidor.

Los problemas asociados con no usar la contraseña están relacionados con el mantenimiento de la seguridad en el ordenador. Si un usuario malintencionado es capaz de acceder al fichero que contiene la clave que no posee contraseña, dicho usuario puede utilizar dicha clave para servir, por ejemplo, páginas web cifradas con la clave de nuestro servidor y, por tanto, que aparentemente están en nuestro servidor web.

El fichero *clave.key* debe ser propiedad del administrador (usuario *root*) y no debe ser accesible por nadie más. Además, es necesario hacer una copia de dicho fichero y guardarla en un lugar seguro, pues si dicho fichero fuera destruido por un

[2] En estos apuntes nos limitaremos a introducir los comandos necesarios para realizar la función deseada. Si se desea obtener más información sobre openssl, etc., puede consultarse el manual de Linux o la página web http://www.openssl.org.

[3] En la actualidad, una clave privada de 512 bits es insegura para la mayoría de usos.

[4] La contraseña debe tener, como mínimo, ocho caracteres, incluyendo entre los caracteres números y símbolos de puntuación, y no debe ser una palabra que esté incluida en un diccionario o fácilmente derivable de una palabra existente en el mismo.

[5] El fichero es creado al redireccionar la salida del comando a un fichero. Si no se redirecciona la salida a un fichero la salida del comando es enviada a la salida estándar, generalmente la pantalla.

borrado, error de disco, etc., después de crear con él un certificado, este certificado no podría utilizarse nunca más y debería ser sustituido por un nuevo certificado.

Si del certificado creado se desea extraer la clave pública asociada, se debe ejecutar el comando:

```
openssl rsa -in clave.key -pubout -out publica.key
```

Donde *clave.key* es el fichero que contiene el par de claves pública/privada generado con cualquiera de los comandos vistos con anterioridad y *publica.key* es el fichero que contendrá la clave pública extraída. Si el fichero con el par de claves pública/privada fue generado con la opción de cifrado, se solicita la contraseña para poder acceder a la clave privada.

Creación de un certificado.

Una vez se han creado el par de claves pública y privada, debemos pasar a crear un certificado, tanto para asegurar nuestra identidad ante los clientes, como para informar a los mismos de nuestra clave pública.

Existen dos modelos de certificados, los certificados emitidos por una autoridad de certificación (Certification Authority)[6] y los certificados autofirmados. Los primeros, son aquellos en los que una entidad, conocida con el nombre de autoridad de certificación, certifica dichos datos, requiriendo dicho tramite el pago de los servicios efectuados, mientras que los segundos son aquellos en los que el propio servidor crea su certificado, asegurando el propio servidor su validez y siendo, por tanto, gratuitos.

Creación de un certificado para su firma por una autoridad de certificación.

Si deseamos crear un certificado para enviarlo a una CA, debemos ejecutar el comando:

```
openssl req -new -key clave.key -out servidor.csr
```

Donde *clave.key* es el fichero con el par de claves pública y privada que hemos generado con anterioridad. Si las claves fueron generadas con la opción de utilizar una contraseña, dicha contraseña será solicitada antes de comenzar la creación del certificado.

A continuación, el sistema solicitará información adicional[7] como el país (ES), la provincia (Valencia), la localidad (Paterna), el nombre de la organización (Universitat de Valencia), el nombre de la unidad (IRTIC), el nombre del ordenador donde se encuentra el servidor y para el cual el certificado será firmado (ordenador.uv.es), la dirección de correo del administrador (webmaster@ordenador.uv.es), así como dos

[6] Un listado de las autoridades de certificación reconocidas puede encontrarse en las opciones de seguridad del navegador web que tengamos instalado en el ordenador.

[7] Los datos que figuran entre paréntesis son aquellos que deberían introducirse para generar la solicitud de un certificado por parte del Instituto de Robótica y Tecnologías de la Información y la Comunicación de la Universitat de València.

campos adicionales (otra contraseña y nombre opcional) que deben dejarse en blanco, pues en caso contrario algunas CAs pueden rechazar la solicitud de firma del certificado al no admitir el uso de dichos campos, no necesarios por otra parte.

Además, debe evitarse en el país, provincia, etc., caracteres especiales como @, #, &, vocales acentuadas, etc., pues algunas CAs rechazan las peticiones de certificados que contienen dichos caracteres[8].

Una vez introducida la información, se creará un fichero llamado *servidor.csr*, el cual debe enviarse a la CA[9]. Una vez firmado, la CA enviará, generalmente por correo electrónico, un fichero con el certificado firmado. Dicho fichero, de nombre *servidor.pem*, es el que debe instalarse para ser usado por el servidor deseado.

Creación de un certificado autofirmado.

Si lo que deseamos crear es un certificado autofirmado, debemos ejecutar el comando:

```
openssl req -new -key clave.key -x509 -days 365 -out servidor.pem
```

Donde, igual que en el caso anterior, *clave.key* es el fichero con el par de claves pública y privada que hemos generado con anterioridad, siendo solicitada la contraseña, si la clave ha sido creada con contraseña. A continuación, el programa solicitará la misma información que en el caso anterior, pudiendo responderse de igual forma que antes y con las mismas restricciones. Como salida, obtendremos un fichero, *servidor.pem*, el cual puede ya instalarse para ser usado por el servidor deseado.

Resaltar que en la generación del certificado autofirmado el argumento *-x509* indica que se firme como X.509, mientras que el argumento *-days 365* indica la validez, en días, del certificado, a partir del instante en que se creó[10].

Instalación de un certificado.

La instalación de un certificado es diferente si se trata del cliente o del servidor.

Instalación de un certificado en el servidor.

La instalación en el servidor de un certificado es tan sencilla como copiar en los directorios adecuados el fichero con el par de claves pública/privada que se generó para crear el certificado y el certificado firmado por la CA o autofirmado.

Así, en el caso del servidor web Apache, si deseamos instalar un certificado, deberemos copiar en el directorio */etc/pki/tls/private* el fichero con la clave generada,

[8] Si el nombre contiene alguno de dichos símbolos, este debe sustituirse por aquel que refleje de forma más fiel el verdadero nombre.
[9] Para el envío, pago del certificado, etc., deben seguirse las instrucciones que proporcione la autoridad de certificación en su página web.
[10] Una duración de un año (365 días) es un estándar de facto en la validez de los certificados para un servidor.

clave.key en nuestro ejemplo, bajo el nombre *localhost.key*, y copiar el certificado firmado o autofirmado en el directorio */etc/pki/tls/certs* con el nombre *localhost.crt*[11].

En el caso de otros servidores, como pueden ser el servidor *dovecot* que atiende los servicios de POP3 o IMAP bajo SSL, el proceso es similar, solo que en este caso los ficheros anteriores, *clave.key* y *servidor.pem*, deben copiarse en el directorio */etc/pki/dovecot/private* y */etc/pki/dovecot/certs* respectivamente, y asignarles los nombres adecuados (*dovecot.pem* a ambos ficheros).

Instalación de un certificado en el cliente.

Una vez hemos creado el certificado, nuestro servidor está listo para usarlo, de forma que cada vez que un cliente le solicite hacer uso del servicio encriptado, se le enviará el certificado.

Si el certificado ha sido firmado por una CA, esta se encontrará, de forma general, reconocida como tal por el cliente, con lo cual el certificado tan solo será comprobado y aceptado para su uso.

Por el contrario, si el certificado ha sido autofirmado, el cliente mostrará al usuario la información contenida en el certificado, solicitándole que de su aprobación a la misma y, por tanto, a seguir utilizando dicho certificado. Esta solicitud se realizará cada vez que el servicio sea requerido.

Para evitar esto último, el usuario tiene la posibilidad de instalar el certificado como un certificado valido para ese ordenador y servicio[12], de forma que, una vez instalado, dicho certificado tendrá validez permanente y no se nos requerirá nunca más su aprobación.

Creación de una autoridad de certificación.

A pesar de la posibilidad de instalar de forma permanente los certificados autofirmados, si una entidad posee un gran número de ordenadores y servicios seguros y desea autofirmarlos, puede ser molesto para un usuario tener que instalar un gran número de certificados autofirmados, o bien, tener que aceptarlos cada vez.

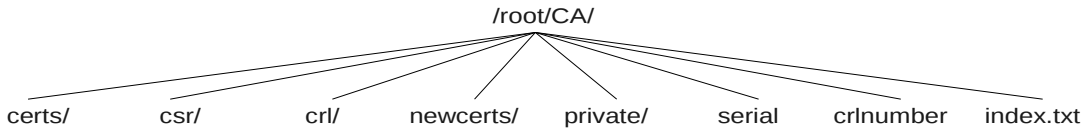
Para evitar esto, podemos, utilizando OpenSSL, crear una nueva autoridad de certificación, lo cual nos permitirá firmar cuantos certificados deseemos. En este caso, el cliente solo deberá descargarse e instalar nuestro certificado autofirmado que nos acredita como autoridad de certificación[13]. A partir de ese momento, todos los certificados que hayan sido firmados por nuestra propia CA tendrán la misma validez que los firmados por cualquier otra CA.

[11] Los nombres de los ficheros pueden modificarse si se modifican dichos nombres en el fichero de configuración del servidor web Apache, lo que veremos en temas posteriores.
[12] La instalación del certificado depende de cada sistema operativo, etc., pero suele resultar muy sencilla de realizar.
[13] El certificado que nos acredita como una autoridad de certificación puede ser puesto a disposición de los usuarios que deseen descargárselo en una página web, por ejemplo.

La creación de una autoridad de certificación es un proceso relativamente sencillo. Para ello, debemos crear un directorio, el cual solo debe ser accesible por el administrador del sistema. Si suponemos que dicho directorio es creado en el directorio raíz del administrador del sistema (directorio */root*), y tiene como nombre *CA*, dentro del mismo debemos crear los directorios que se detallan a continuación:

Directorio	**Descripción**
/root/CA/certs	Directorio donde se almacenaran los certificados ya firmados y enviados a los clientes.
/root/CA/newcerts	Directorio donde se guardan los certificados que acaban de ser firmados.
/root/CA/crl	(Certificate Revokation List). Directorio donde los certificados revocados son almacenados.
/root/CA/csr	(Certificate Signing Request). Directorio donde son almacenadas las peticiones de certificados pendientes de firmar.
/root/CA/private	Directorio donde se almacenaran la clave privada de la autoridad de certificación, así como las demás claves privadas que sean generadas para los diferentes servicios.

Además de los directorios anteriores, debemos crear tres ficheros, de nombres */root/CA/serial*, */root/CA/crlnumber* y */root/CA/index.txt*. El primero de ellos, */root/CA/serial*, será un fichero que deberá contener, inicialmente, el valor 01 seguido de un salto de línea[14]. Este fichero indica el número de serie que tendrá el siguiente certificado que sea firmado. El segundo fichero */root/CA/crlnumber* indica el número que tendrá la siguiente lista de certificados revocados[15] y también deberá tener, inicialmente, el valor 01 seguido de un salto de línea. Por su parte, el fichero */root/CA/index.txt* deberá estar vacío inicialmente y será una "base de datos" con información sobre los certificados firmados. Ambos ficheros son actualizados de forma automática una vez firmado un nuevo certificado. Por tanto, la estructura de directorios y ficheros que deberá existir inicialmente es:



Una vez creados los directorios y ficheros, debemos modificar el fichero de configuración de OpenSSL[16], que es */etc/pki/tls/openssl.cnf*. Para no modificar dicho fichero, podemos realizar una copia del mismo en el directorio donde se encuentra nuestra autoridad de certificación. Así, por ejemplo, puede copiarse a */root/CA/irtic.cnf*.

Dentro del fichero de configuración, tan solo es necesario modificar, generalmente, tres secciones, *[CA_default]*, *[req_distinguished_name]* y *[req]*. En la sección *[CA_default]* debemos especificar los siguientes valores[17]:

[14] Este fichero y el siguiente pueden crearse con el comando *echo "01" > fichero*.
[15] En las versiones antiguas de openssl este fichero no erá necesario, utilizando en su lugar el valor que contenía el fichero serial.
[16] Un ejemplo completo de un directorio de CA se encuentra en /etc/pki/CA, mientras que scripts para ejecutar las labores más comunes de una CA se encuentra dentro del directorio /etc/pki/tls/misc.

Variable	Valor	Descripción
dir	/root/CA	Directorio raíz de la autoridad de certificación.
certs	$dir/certs	Directorio donde se almacenaran los certificados ya firmados.
crl_dir	$dir/crl	Directorio donde los certificados revocados son almacenados.
database	$dir/index.txt	Archivo con la base de datos de los certificados.
new_certs_dir	$dir/newcerts	Directorio donde se guardan los certificados que acaban de ser firmados.
certificate	$dir/irtic.pem	Archivo con la clave pública de la autoridad de certificación.
serial	$dir/serial	Archivo con el número de serie de los certificados.
crlnumber	$dir/crlnumber	Archivo con el número de serie de revocación. Si se desea un funcionamiento como en versiones antiguas de openssl puede comentarse esta línea.
crl	$dir/crl.pem	Lista de los certificados revocados.
private_key	$dir/private/irtic.key	Archivo con la clave privada de la autoridad de certificación.
RANDFILE	$dir/private/.rand	Archivo con el número aleatorio privado.
x509_extensions	usr_cert	Extensiones que han de añadirse al certificado.
name_opt	ca_default	Formato en que se mostrará el nombre del certificado antes de que sea firmado.
cert_opt	ca_default	Formato en que se mostrará un certificado antes de que sea firmado.
default_days	365	Días por defecto para los que se firma el archivo.
default_crl_days	30	Días por defecto en que debe ser actualizada la lista de certificados revocados de esta autoridad de certificación.
default_md	default	Compendio de mensaje utilizado, por defecto es md5 (valor default).
preserve	no	Indica si se ha de mantener o no el orden Domain Name.
policy	policy_match	Política por defecto a aplicar si no se especifica ninguna.

La variable *policy* de la tabla anterior puede tomar dos valores, *policy_math* o *policy_anything*, que se encuentran definidas como se indica a continuación:

```
[ policy_match ]
countryName             = match
stateOrProvinceName     = match
organizationName        = match
```

[17] Los valores de configuración que utilizaremos suponen que el directorio de la autoridad de certificación se ha creado dentro del directorio */root* del sistema.

```
organizationalUnitName  = optional
commonName              = supplied
emailAddress            = optional

[ policy_anything ]
countryName             = optional
stateOrProvinceName     = optional
localityName            = optional
organizationName        = optional
organizationalUnitName  = optional
commonName              = supplied
emailAddress            = optional
```

Donde el valor *match* indica que ese campo debe ser igual en la autoridad de certificación y en el certificado a firmar, mientras que el valor *supplied* es que es obligatorio que sea proporcionado en el certificado a firmar, aunque puede ser distinto del valor existente en la autoridad de certificación, y el valor *optional* es que puede no ser proporcionado en el certificado a firmar.

La política *policy_match* sirve para que una autoridad de certificación pueda firmar sus propios certificados, mientras que la política *policy_anything* permite que una autoridad de certificación firme certificados de cualquier otra organización.

Por otra parte, en la sección *[req_distinguished_name]* debemos especificar los siguientes valores:

Variable	**Valor**	**Descripción**
countryName_default	ES	País de emisión del certificado
stateOrProvinceName_default	Valencia	Estado o provincia de emisión
localityName_default	Paterna	Localidad de emisión del certificado
0.organizationName_default	Universitat de Valencia	Nombre de la organización
organizationalUnitName_default	IRTIC	Nombre de la sección
commonName_default	Autoridad de Certificacion del IRTIC	Nombre de la autoridad
emailAddress_default	webmaster@irtic.uv.es	Dirección de correo del responsable de la autoridad de certificación

Por último, en la sección *[req]* puede ser necesario, en algunos casos, modificar los valores por defecto:

Variable	**Valor por defecto**	**Descripción**
default_bits	2048	Bits por defecto de la clave privada.
default_md	sha1	Compendio de mensaje usado por defecto.

Una vez configurado el fichero, podemos crear el certificado de la autoridad de certificación como[18]:

```
openssl req -new -x509 -days 3650 -config /root/CA/irtic.cnf
-keyout /root/CA/private/irtic.key -out /root/CA/irtic.pem
```

En la ejecución de la creación del certificado se solicita una contraseña de acceso a la clave, así como los datos del país, localidad, etc. Si el fichero de configuración ha sido correctamente configurado, todos los datos por defecto serán validos.

Una vez generados los certificados, podemos comprobar la clave secreta generada con:

```
openssl rsa -in /root/CA/private/irtic.key -text
```

Así como el certificado generado:

```
openssl x509 -in /root/CA/irtic.pem -text
```

Y el propósito del mismo con:

```
openssl x509 -in /root/CA/irtic.pem -purpose
```

Distribución de la acreditación de una autoridad de certificación.

De forma general, las autoridades certificadoras se instalan con el sistema operativo, siendo modificadas por las actualizaciones de los propios sistemas operativos al incluirse nuevas autoridades de certificación.

Sin embargo, si hemos creado nuestra propia autoridad de certificación, desearemos que esta pueda ser instalada como autoridad de certificación raíz, de forma que cualquier certificado firmado por nuestra CA sea admitido automáticamente.

Para ello, la forma más cómoda consiste en colocar, en un sitio al que sea posible acceder por red (un servidor web, un servidor de FTP, etc.), el fichero *irtic.pem* obtenido con anterioridad[19]. Dicho fichero, descargado por la red e instalado en la lista de autoridades de certificación raíz de un ordenador, nos confiere la autoridad de una CA para ese ordenador, por lo que todo certificado firmado por nosotros será automáticamente validado.

[18] Suponemos que todos los comandos se ejecutan desde el directorio dentro del cual se encuentra creado el directorio raíz de la autoridad de certificación (directorio CA). En caso de no ser así debería cambiarse el comando para indicar el camino adecuado.

[19] Al fichero irtic.pem puede ser necesario cambiarle la extensión, por ejemplo a irtic.cer, para que algunos sistemas operativos como Windows realicen de forma automática la importación del certificado.

Firma de un certificado por la autoridad de certificación.

Una vez hemos creado la autoridad de certificación, vamos a explicar como firmar una solicitud de certificado. Recordar, previamente, que dicho certificado será valido para todos aquellos clientes que reconozcan la validez de nuestra autoridad de certificación.

Para ello, supongamos que nos ha llegado una solicitud de certificado, que tenemos en el fichero */root/CA/csr/servidor.csr*. Dicha solicitud ha sido generada tal y como vimos con anterioridad.

Podemos examinar dicha solicitud que nos ha llegado mediante el comando:

```
openssl req -in /root/CA/csr/servidor.csr -text
```

Si consideramos que dicha solicitud es correcta, podemos pasar a firmarla mediante:

```
openssl ca -config /root/CA/irtic.cnf -in
/root/CA/csr/servidor.csr -verbose
```

El tiempo de validez del certificado es dado por el valor de la variable *default_days* del fichero de configuración, 365 días en el ejemplo que estamos utilizando.

Como salida del comando anterior, obtendremos, dentro del directorio *./CA/newcerts* un fichero con el nombre *<número de serie>.pem*, donde *<número de serie>* es el número de serie que en el instante de la firma tuviera el fichero *serial*.

Con el comando comentado en la línea anterior, al no especificar ninguna política, utilizamos la política por defecto, por lo que solo podemos firmar certificados, de nuestra propia organización, que es la política por defecto. Si deseamos firmar certificados con nuestra autoridad de certificación para cualquier organización externa, debemos ejecutar el comando:

```
openssl ca -config /root/CA/irtic.cnf -in
/root/CA/csr/servidor.csr -verbose -policy policy_anything
```

El cual indica que no es necesario que coincida el nombre de la autoridad de certificación con el nombre de la autoridad que solicita la firma del certificado[20].

Revocación de certificados por una autoridad de certificación.

Toda autoridad certificadora tiene la posibilidad de revocar algún certificado emitido. El motivo de revocar un certificado puede ser tan sencillo como la caducidad del mismo[21], el extravío de algún certificado, etc.

[20] Este comando lo que hace es seleccionar la política "policy_anything" del fichero de configuración en lugar de la política por defecto "policy_match".

[21] Una autoridad de certificación no puede firmar dos certificados para el mismo ordenador sin haber revocado previamente el certificado anterior, obteniendo, en caso de no hacerlo, un mensaje de error

Para revocar un certificado, se introduce el certificado en el directorio *./CA/crl* y se actualiza el fichero *./CA/crl.pem* que contendrá los certificados revocados por nuestra autoridad de certificación. Esto se realiza mediante el comando:

```
openssl ca –config /root/CA/irtic.cnf -revoke /root/CA/certs/servidor.crt
```

Que genera la revocación del certificado *servidor.crt* y la incluye en el directorio de certificados revocados. Por su parte, el comando:

```
openssl ca –config /root/CA/irtic.cnf -gencrl -out /root/CA/crl/crl.pem
```

Que actualiza la lista de certificados revocados. Podemos examinar la lista de certificados revocados mediante el comando:

```
openssl crl -in /root/CA/crl/crl.pem -text
```

La lista de certificados revocados debe ser puesta a disposición de los usuarios de la red de forma similar a como ponemos nuestra autoridad de certificación. Esto permite a los usuarios descargarse la lista de certificados revocados e instalarla en su ordenador, pudiendo comprobar que los certificados que le llegan, además de haber sido firmados por nuestra CA, continúan siendo validos para nuestra CA que los ha firmado.

Ejercicios.

1- Ejecutar los comandos necesarios para crear un certificado autofirmado con la clave privada protegida por contraseña, para un servidor de nombre *mi_servidor*.

2- Una empresa, de nombre *mi_empresa*, posee un gran número de servidores seguros, por lo que desea convertirse en una autoridad de certificación. Ejecutar los comandos necesarios para establecer dicha empresa como autoridad de certificación.

3- Una empresa, que es una autoridad de certificación, desea crear un certificado para un nuevo servidor que acaba de adquirir y cuyo nombre es *mi_nuevo_servidor*. Ejecutar todos los comandos necesarios para crear el certificado del servidor y firmarlo por parte de la autoridad de certificación.

4- Deseamos firmar un nuevo certificado, ya creado, para un servidor llamado *mi_servidor*, al cual ya hemos firmado un certificado con anterioridad. Ejecutar los comandos necesarios para firmar el nuevo certificado.

indicando que no es posible actualizar el fichero *./CA/index.txt*.