- OPENSSL:

El proyecto OpenSSL es un esfuerzo conjunto para desarrollar una implementación robusta, de nivel comercial, con todas las características y de tipo Open Source de los protocolos Secure Sockets Layer (SSL v2/v3), y Transport Layer Security (TLS v1), además de bibliotecas con propósitos criptográficos. Y su principal diferencia con un paquete SSL se puede apreciar en el eslogan presente en su página web: *"¿Porque comprar un paquete SSL como una caja negra, cuando puede obtener una abierta y gratis?"* osea que es código libre y por lo mismo, gratis.

TLS es una forma superior de SSL v3 con funciones criptográficas fortalecidas, debido a esto es considerado el estándar a utilizar para comercio electrónico. Tanto el Protocolo de Seguridad del Nivel de Transporte (TSL), como el protocolo Seguridad a Nivel de Transporte inalámbrico (Wireless Transport Layer Security, WTLS) son descendientes directos del SSL.

Para la instalación de OpenSSL no hay necesidad de complicarse mucho, pues viene como paquete en la mayoría de las actuales distribuciones de linux para instalar, ya que no es el propósito de este documento ser un HOWTO de instalación, se nombrarán algunos pasos a seguir para poder crear todo lo necesario para ocupar OpenSSL. El directorio por defecto para la instalación es */var/ssl/* y el archivo de configuración se encuentra en */usr/lib/ssl/openssl.cnf*, las utilidades y otras librerías se encuentran en */usr/lib/ssl*.

Para crear un certificado raíz, se debe crear un certificado que sea firmado por nosotros mismos para poder actuar como autoridad certificadora de los certificados que sean creados en nuestro servidor, para esto se usa:

```
openssl req -new -x509 -keyout private/cakey.pem \
-out cacert.pem -days 3650.
```

Esta línea de comando crea un certificado con una duración de 10 años (365 días) y guarda la llave (RSA PRIVATE KEY) al archivo *private/cakey.pem* mientras el certificado (CERTIFICATE) va al archivo *cacert.pem* , ahora hay que asegurarse que el certificado raíz se utilice sólo para firmar otros certificados, la llave privada de CA es muy delicada y por lo tanto debe ser protegida de posibles ataques o intentos de obtenerla por personas no autorizadas. Ahora que se tiene una autoridad con certificado raíz, es necesario que otras personas confíen en este certificado, además que lo descarguen y lo registren en su navegador web. Se debe revisar que el archivo index.txt se encuentra vacío y que el archivo serial contiene 01.

Para instalar este certificado raíz como un certificado raíz en el cual se confía, primero es necesario almacenar solo la parte certificado:

openssl x509 -in cacert.pem -out cacert.crt

La idea es tener este certificado accesible para que el resto de las personas puedan descargarlo e instalarlo en sus navegadores, por ende se pone en el servidor web, y además el servidor debe tener una entrada mime para archivos .crt, sin embargo siempre es probable que suplantar al servidor web y por ende suplantar al CA, por eso se recomienda tener más de una forma de obtener el certificado.

Manejo de Certificados:

a) Generar y Firmar Certificados:

openssl req –new keyout newreq.pem –out newreq.pem –days 365

crea una nueva clave privada y una solicitud de certificado, la cual guarda en el archivo *newreq.pem*, usar un nombre común (Common Name, CN) si el certificado a firmar será utilizado para certificar una página web se debe usar la dirección de ésta, por ejemplo *www.servidor.cl* o ingresar un nombre de usuario del tipo usuario@servidor.cl, para autentificar los correos de ese usuario.

openssl ca -policy policy_anything -out newcert.pem -infiles newreq.pem

Esto firma la solicitud anteriormente creada usando cacert.pem y lo convierte en un certificado en el archivo *newcert.pem*, se crea un archivo en el directorio *newcerts/* y los archivos *index.txt* y *serial* serán actualizados.

b) Revocar un certificado:

openssl –revoke newcert.pem

Esto actualiza la base de datos de los certificados, pero debe ser actualizada la lista de certificados revocados:

openssl ca –gencrl –out crl/sopac-ca.crl

esta lista de certificados revocados(CRL), debe estar también disponible en el sitio web; además se puede desear agregar ciertos parámetros como la hora y el día cuando e certificado fue revocado, agregando *–crldays numero_dias*.

c) Renovar un certificado:

El usuario envía su solicitud de certificado previa, o crea una nueva basada en su llave privada, en primer lugar se debe revocar el certificado previo y firmar nuevamente la solicitud de certificado. Es posible buscar el certificado antiguo, buscando en el archivo index,txt por el nombre que corresponde, se obtiene el número serial "xx" y se usa el archivo *cert/"xx".pem* como certificado para el proceso de revocación.

También se puede firmar un certificado manualmente para asegurar que las fechas de inicio y vencimiento de su período de validez de nuevo certificado son correctos.

*openssl ca –policy policy_anyting –out newcert.pem –infiles newreq.pem *
-startdate [now] –eddate [end_date]

Reemplazar los valores [now] y [end_date] con los valores correctos.

d) Visualizar el certificado:

Se puede tener un certificado en forma codificada, para leer los detalles del certificado, se ejecuta:

Openssl x509 –in newcert.pem –noout –text

Se ha nombrado en variadas ocasiones el archivo index.txt, este archivo contiene información de los certificados manejados por OpenSSL, todas las entradas del archivo están marcadas con R de revocado, V de válido y E de expirado.

Finalmente se mencionará que OpenSLL posee compatibilidad con distintos aplicaciones de uso masivo como por ejemplo: Apache, protocolos de correo como IMAPS o POP, Microsoft Key Manager y para correo electrónico seguro, donde puede funcionar con certificados mime, ser utilizado con el Outlook, Netscape Messenger y varios otros clientes de correo.