

Implementación de servidor Apache seguro (https).

Dedicado a todos los que han hecho que el software libre sea una realidad...

En el siguiente documento explicaré los procesos necesarios para asegurar el tráfico en un servidor web utilizando SSL/TLS. No me referiré a la configuración de Apache, por lo que se asume un servidor web “up and running”...

Algunas cosas que es necesario saber:

1. Por lo general los certificados son firmados por una Autoridad Certificante (CA), los que generalmente son vendidos a un precio variable dependiendo de las capacidades del certificado, los browsers que soporta, etc.
2. En el caso de los certificados, el precio no siempre es indicador de calidad.
3. Algunos ejemplos de CA son:
 1. InstantSSL <http://www.instantssl.com>
 2. Thawte <http://www.thawte.com>
 3. VeriSign <http://www.verisign.com>
4. Existe la posibilidad de crear nuestros propios certificados, pero esto solo es recomendado para hacer pruebas o cuando un número pequeño de personas estará accediendo al servidor y no está en nuestros planes tener el certificado en múltiples máquinas.

Este documento se basará en este último punto.

Generando las claves.

Para comenzar con la configuración de SSL, lo primero que debemos hacer es instalarlo en caso de que no esté instalado.

```
# apt-get install openssl
```

Una vez instalado el paquete openssl procedemos a generar una clave RSA 1024-bit. Para esto crearemos un directorio en donde guardas las claves, que estará bajo /etc/apache2:

```
# mkdir ssl.key
# cd ssl.key
# openssl genrsa -out server.key 1024
# chmod 600 server.key
```

Si deseamos tener un mayor nivel de seguridad, una buena idea puede ser encriptar la clave agregando el parámetro -des3 como argumento al comando openssl:

```
# openssl -des3 genrsa -out server.key 1024
```

A continuación, y en caso de que deseemos tener nuestro certificado firmado por una CA, debemos generar una clave pública y una petición de forma de certificado (Certificate Signing Request):

```
# mkdir ssl.csr
# cd ssl.csr
# openssl req -new -key ../ssl.key/server.key -out server.csr
```

Como en nuestro caso usaremos un certificado firmado por una CA, o si queremos probar nuestra configuración, generaremos un certificado firmado por nosotros mismos que guardaremos en el archivo llamado server.crt

```
# mkdir ssl.crt
# cd ssl.crt
# openssl req -new -x509 -nodes -sha1 -days 365
-key ../ssl.key -out server.crt
```

Con esto terminamos la parte de creación de certificados y nos disponemos a configurar el Apache para que soporte SSL/TLS.

Configurando Apache para soportar SSL/TLS.

Lo primero que debemos hacer es agregar módulo ssl al apache. Para esto debemos instalar el paquete libapache-mod-ssl:

```
# apt-get install libapache-mod-ssl
```

A continuación debemos habilitar el módulo recién instalado. Esta tarea es diferente en algunas distribuciones, pero en debian lo único que debemos hacer es un enlace simbólico de los archivos ssl.conf y ssl.load desde el directorio mods-available al mods-enabled.

```
# cd mods-enabled
# ln -sf ../mods-available/ssl.load ssl.load
# ln -sf ../mods-available/ssl.conf ssl.conf
```

Hecho esto, editamos el archivo de configuración de nuestro sitio web. Esto también varía de distribución en distribución. En debian debemos editar el archivo correspondiente a nuestro sitio, ubicado bajo el directorio sites-enabled

```
# cd sites-enabled
# nano mi-web

Listen *:443
<VirtualHost *:443>
    ServerName          server.dominio.cl
    DocumentRoot        /var/www
    DirectoryIndex index.php index.html index.htm
    SSLEngine           On
    SSLCertificateKeyFile /etc/apache2/ssl.key/server.key
    SSLCertificateFile  /etc/apache2/ssl.crt/server.crt
</VirtualHost>
```

Una vez realizada esta configuración procedemos a verificar que todo esté correctamente configurado. Para esto usamos el `apache2ctl` de la siguiente forma:

```
# apache2ctl configtest
Syntax OK.
# /etc/init.d/apache2 restart
```

Si hasta aquí todo salió bien, podremos ingresar a nuestro servidor usando la dirección <https://www.dominio.cl> en donde se nos pedirá que aceptemos el certificado.

Autor

Boris Quiroz Q.
borisq22[at]gmail[dot]com
<http://g00fy.homelinux.org>



Licencia

Este documento se distribuye bajo la licencia Creative Commons by-nc-sa/2.0/cl lo que le permite copiar, distribuir, comunicar y ejecutar públicamente este documento, bajo las siguientes condiciones:

1. **Atribución:** Debes reconocer y citar la obra de la forma especificada por el autor (osea yo).
2. **No Comercial:** No puedes usar esta obra para fines comerciales.
3. **Licenciar Igual:** Si alteras o transformas esta obra, o generas una obra derivada, sólo puedes distribuir la obra generada bajo una licencia idéntica a ésta.