Implementació d'un servidor web Apache2 segur (HTTPS)

El mòdul "ssl" és una interfaç entre l'Apache i la llibreria OpenSSL que permet que el primer pugui fer servir la segona per autentificar màquines amb certificats digitals i encriptar la comunicació amb els clients via TLS, entre altres aspectes relacionats amb la seguretat de les conexions. A l'Ubuntu aquest mòdul s'instal.la per defecte juntament amb el propi servidor Apache (està dins del paquet "apache2-bin") però a Fedora cal instal.lar el paquet "mod_ssl".

> NOTA: També existeix el mòdul "gnutls" (a l'Ubuntu el paquet que l'instal.la es diu "libapache2-mod-gnutls" i a Fedora, "mod_gnutls") similar en objectius al "ssl" però que utilitza GnuTLS.

Per crear un certificat autosignat útil per Apache es poden executar directament les comandes adients oferides per la suite OpenSSL (veure document apart) o bé utilitzar algun programa "wrapper" (embolcall) que oculta la complexitat del procés, facilitant la feina. Els programes OpenVPN o Postfix, per exemple, ofereixen cadascun un "wrapper" propi que podríem utilitzar si tinguéssim instal.lat algun dels dos, però també és força habitual usar algun "wrapper" genèric ofert la pròpia distribució (si és que l'ofereix); per exemple, a Ubuntu hi ha el paquet "ssl-cert", el qual conté la comanda *make-ssl-cert* que s'executa així: *make-ssl-cert /usr/share/ssl-cert/ssleay.cnf server.crt* (se'ns preguntarà pel nom DNS del servidor i es guardarà el certificat i la clau privada dins l'arxiu "server.crt" -¡sí, dins el mateix fitxer!-; l'arxiu "ssleay.cnf" fa de plantilla). De totes maneres, nosaltres seguirem la forma "estàndar" de creació de certificats fent servir directament les comandes OpenSSL

En qualsevol cas, per fer que l'Apache utilitzi el certificat (autosignat) que haguem creat, primer cal habilitar el SSL; a l'Ubuntu això simplement es fa així: *a2enmod ssl* Seguidament caldrà afegir a l'arxiu de configuració del VirtualHost "a assegurar" les següents línies (sempre dins de la secció <VirtualHost> a la mateixa alçada que la directiva *DocumentRoot*; es pot fer servir l'arxiu "default-ssl.conf" com a plantilla si es vol):

```
SSLEngine on
SSLCertificateFile /ruta/al/certificat/cert.crt
SSLCertificateKeyFile /ruta/a/la/clau/privada/clauPriv.key
```

> NOTA:Si estiguéssim en el cas de tenir un fitxer que inclogui tant el certificat com la clau privada tot en un (això també és possible), les línies a escriure serien llavors:
>
> ```
> SSLEngine on
> SSLCertificateFile /ruta/al/fitxer/que/inclou/certificat/mes/clau/privada/server.crt
> ```

En realitat, el VirtualHost “default-ssl” de l'Ubuntu ja ver configurat per defecte llest per funcionar (escoltant al port 443; comproveu que la directiva *Listen 433* a "ports.conf" està activa). Només cal assegurar-se de les rutes a utilitzar pel certificat i clau privada del servidor i ja està. Podràs observar que en aquest fitxer (¡i en el fitxer de configuració del mòdul, "ssl.conf"!) hi han més directives relacionades amb l'SSL, que no estudiarem.

> NOTA: Antigament, la tecnologia TLS no podia oferir múltiples VirtualHosts des d'una mateixa IP perquè el servidor no sabia quin VirtualHost estava sol.licitant el client a la capçalera"Host" (ja que aquesta també estava encriptada) i per tant no sabia quin certificat -el corresponent a cada lloc- oferir. El que es feia llavors era tenir un únic certificat compartit per tots els llocs amb la mateixa IP (una sol.lució gens professional). Afortunadament, avui dia aquest problema ja no hi és gràcies a la tecnologia Server Name Indication (https://en.wikipedia.org/wiki/Server_Name_Indication)

Un cop configurat el VirtualHost segur i activat mitjançant *a2ensite nomVH*, es pot comprovar, abans de reiniciar el servidor, si tots els fitxers de configuració de l'Apache estan correctament escrits amb la comanda *apache2ctl configtest* . Si tot és correcte, podrem iniciar el servidor i accedir al lloc segur escrivint a la barra del navegador una direcció tal com https://www.nom.servidor (¡important indicar la "s" a "https"...si no l'Apache es queixarà de què volem accedir a un lloc HTTPS fent servir el protocol HTTP!; el port 443 no cal afegir-lo perquè indicant "https" ja se sobrentén igual que se sobrentenia el 80 quan accedíem a HTTP). De totes formes, si volguéssim redireccionar automàticament al lloc HTTPS un usuari que hagués escrit "http" en comptes de "https" a la barra de direccions, el més fàcil és utilitzar la directiva *Redirect* dins del VirtualHost "estàndar" per reencaminar-lo a l'altre; és a dir, fer així:

```
<VirtualHost *:80>
    ServerName nom.dns.servidor
    Redirect permanent / https://nom.dns.servidor    #Aquí es pot posar la IP també
</VirtualHost>
```

Una altra manera, més complicada, d'aconseguir el mateix seria amb el mòdul "Rewrite", tal com s'explica aquí: https://wiki.apache.org/httpd/RewriteHTTPToHTTPS

> NOTA: Un tutorial més extens es pot trobar a https://www.digitalocean.com/community/tutorials/how-to-create-a-self-signed-ssl-certificate-for-apache-in-ubuntu-18-04