PRACTICA SSL APACHE

Mario Arnedo

Lo primero que vamos a hacer es crear un certificado con el siguiente comando:

```
mario@mario:~$ sudo openssl req -new -x509 -nodes -days 365 -keyout privada.key -out certificado.pem
```

Donde:

- **-new**: genera una petición de certificado.
- **-x509**: indica que va a ser un certificado autofirmado.
- **-nodes**: evita que se solicite una contraseña al usuario para generar la clave privada. Si no se indica esta opción, hay que introducir una contraseña que se utiliza para encriptar la clave privada y hay que configurar el servidor web para que se le proporcione la misma contraseña de la clave privada.
- **-days n**: permite especificar el número de días (n) de vigencia del certificado. En el ejemplo se ha indicado que el certificado caducará dentro de 1 año. Si no se especifica esta opción se obtiene un certificado con 1 mes de validez.
- **-keyout *nombreArchivo***: permite especificar el nombre del archivo en el que se va a almacenar la clave privada. Si no se indica esta opción, se guarda en un archivo llamado *privkey.pem*.
- **-out *nombreArchivo***: permite indicar el nombre del archivo que va a almacenar el certificado. Si no se especifica, se mostrará en pantalla.

Al ejecutar este comando, nos indicara que agregemos algunos datos para generar el certificado:

```
Generating a 2048 bit RSA private key
.....................+++
...+++
writing new private key to 'privada.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:ES
State or Province Name (full name) [Some-State]:LaRioja
Locality Name (eg, city) []:Logrono
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Jesuitas
Organizational Unit Name (eg, section) []:Daw
Common Name (e.g. server FQDN or YOUR name) []:www.empresa.es
Email Address []:mario@mario.es
mario@mario:~$
```

Seguidamente hay que activar el modulo de ssl:

```
Email Address []:mario@mario.es
mario@mario:~$ sudo a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed certific
tes.
To activate the new configuration, you need to run:
  systemctl restart apache2
```

En mi caso ya estaba activado, al ejecutar el comando, activara dos modulos, el del ssl y el socache_shmcb

Ahora lo que tenemos que hacer es activar la escucha por el puerto 443, para ellos vamos al fichero de configuración de ports.conf

```
mario@mario:~$ sudo nano /etc/apache2/ports.conf
```

Y dentro de ahí, comprobamos que los puertos estan escuchando:

```
 GNU nano 2.9.3                          /etc/apache2/ports.conf

# If you just change the port or add more ports here, you will likely also
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf
NameVirtualHost *:80
NameVirtualHost *:9999
Listen 80
Listen 9999
<IfModule ssl_module>
        Listen 443
</IfModule>

<IfModule mod_gnutls.c>
        Listen 443
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

Ahora lo que tendremos que hacer es crear un host virtual con la siguiente configuración (podemos coger el fichero de configuracion de por defecto de ssl y copiarlo):

```
mario@mario:~$ sudo cp /etc/apache2/sites-available/default-ssl.conf /etc/apache2/sites-available/ma
rio-ssl.conf
```

```
mario@mario:~$ sudo nano /etc/apache2/sites-available/mario-ssl.conf
```

Tiene que quedar algo asi:

```
 GNU nano 2.9.3              /etc/apache2/sites-available/mario-ssl.conf


        <VirtualHost *_:443>
                ServerAdmin mario@mario.es

                DocumentRoot /var/www/html/ssl

                SSLEngine on

                SSLCertificateFile      /home/mario/certificado.pem
                SSLCertificateKeyFile /home/mario/privada.key

        <Directory /var/www/html/ssl>
                Order allow,deny
                allow from all
        </Directory>
        </VirtualHost>
```

Seguidamente creamos la carpeta que le hemos indicado en el fichero anterior (sino la tenemos de antes)

```
mario@mario:~$ sudo mkdir /var/www/html/ssl
mario@mario:~$ sudo chown -R www-data:www-data /var/www/html/ssl/
```

Ahora lo que tenemos que hacer es crear un index.html con algo dentro para saber si estamos dentro o no en el futuro:

```
 GNU nano 2.9.3                    /var/www/html/ssl/index.html

Hola, estas en un lugar con HTTPS
```

Por ultimo, tendremos que activar este sitio, para ello quitamos el de por defecto:

```
mario@mario:~$ sudo a2dissite 000-default.conf
Site 000-default disabled.
To activate the new configuration, you need to run:
  systemctl reload apache2
```

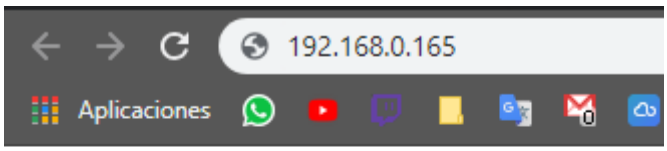
Y nos pedirá reiniciar el apache 2 con el comando.

Cuando lo tengamos, simplemente tendremos que activar el sitio con el ssl:

```
mario@mario:~$ sudo a2ensite mario-ssl.conf
Enabling site mario-ssl.
To activate the new configuration, you need to run:
  systemctl reload apache2
```

Y volvemos a reiniciar el apache2

Ahora tendremos que ir a la Ip veremos que accedemos:



Hola, estas en un lugar con https