Creación de certificados SSL Let´s Encrypt en Arch Linux
para OpenMeetings 7.2.0

Estos certificados SSL son para que OpenMeetings pueda funcionar en "https".

Si no tuviera instalado OpenMeetings 7.2.0, puede descargar directamente el tutorial de instalación desde aquí:

Descargar Instalación OpenMeetings 7.2.0 en Arch Linux

Doy las gracias a Maxim Solodovnik y a Carlos Heras, sin cuya colaboración en las pruebas prácticas no habría podido confirmar el correcto funcionamiento y así poder publicar el presente tutorial.
Igualmente doy las gracias a todos aquellos que han contribuido, tales como Marcus Schulz y Daniel Baker. Gracias a todos ellos.

Comenzamos…

1)

------ Creación del certificads Let´s Encrypt SSL ------

Descargamos certbot, necesario para hacer los certificados:

sudo pacman -S certbot

Es importante que su servidor-pc no tenga en uso el puerto 80 con algún servidor web o algún otro. Si fuera así deténgalo y continúe con este paso. Cuando concluya los certificados podrá lanzarlo nuevamente. Este puerto ha de estar abierto en el router y en el firewall.

Let´s Encrypt valida “SSL Certificate Authority (CA)” el o los dominios de tu servidor.

Lo ejecutaremos con el parámetro --standalone, para que usted pueda añadir al final, cada dominio que requiera un certificado, por ejemplo: -d nuevoejemplo.com
Cambie “ejemplo.com” por el verdadero dominio de su servidor:

sudo certbot certonly --standalone -d ejemplo.com -d www.ejemplo.com

Preguntará por una dirección de correo de administración. Ponga uno verdadero para que le mantenga informado acerca de los certificados:

Installation succeeded.
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Plugins selected: Authenticator standalone, Installer None
Enter email address (used for urgent renewal and security notices) (Enter 'c' to cancel): ...aquí ponga su dirección de correo y pulse **Enter**

Preguntará si está de acuerdo;

- -
Please read the Terms of Service at https://letsencrypt.org/documents/LE-SA-v1.2-November-15-2017.pdf. You must agree in order to register with the ACME server at https://acme-v02.api.letsencrypt.org/directory
- -
(A)gree/(C)ancel: ...escriba... a ...y pulse **Enter**

Preguntará si quiere compartir su dirección de correo:

- -
Would you be willing to share your email address with the Electronic Frontier Foundation, a founding partner of the Let's Encrypt project and the non-profit organization that develops Certbot? We'd like to send you email about our work encrypting the web, EFF news, campaigns, and ways to support digital freedom.
- -
(Y)es/(N)o: ...escriba... n ...y pulse **Enter**

...cuando finalice de hacer los certificados con éxito, mostrará lo siguiente:

IMPORTANT NOTES:
- Congratulations! Your certificate and chain have been saved at:
  /etc/letsencrypt/live/tu_dominio/fullchain.pem
  Your key file has been saved at:
  /etc/letsencrypt/live/tu_dominio/privkey.pem
  Your cert will expire on 2020-06-24. To obtain a new or tweaked
  version of this certificate in the future, simply run
  letsencrypt-auto again. To non-interactively renew *all* of your
  certificates, run "letsencrypt-auto renew"
- If you like Certbot, please consider supporting our work by:

  Donating to ISRG / Let's Encrypt: https://letsencrypt.org/donate
  Donating to EFF: https://eff.org/donate-le

2)

------ Chequear certificado de dominio ------

Vamos a ver donde están almacenados los certificados que acabamos de crear, que en nuestro caso es /etc/letsencrypt/live:

sudo ls /etc/letsencrypt/live

...mostrará el nombre de su dominio: tu_dominio

Todos los dominios que usted haya especificado en el paso anterior, se hallarán en el mismo certificado.

3)

------ Renovación del certificado SSL ------

El certificado Let´s Encrypt tiene un inconveniente, y es que su validez es solo de 90 días, por lo que habremos de renovarlo. Recuerde tener abierto el puerto 80.

Podemos hacer esto manualmente (siempre conectados a Internet):

sudo certbot renew

4)

------ Configuración de Tomcat-OpenMeetings con los certificados SSL ------

.

Estos pasos 3 y 4, hay que repetirlos cada 80 días, pues son 90 los días válidos de Letsencrypt.

He seguido la ruta de instalación de OM que muestran los tutoriales de OpenMeetings que se encuentran en su sitio wiki oficial. Es decir **/opt/open720.**

Si usted hubiera hecho la instalación en una ruta distinta, modifique lo que indico a continuación.
Ya hicimos los certificados letsencrypt de nuestro dominio en el paso 1.
Ahora vamos a crear un PKCS12 que contenga la full chain y la privada. Es necesario tener instalado openssl. Lo instalamos si no:

```
sudo pacman -S openssl
```

Ahora lanzamos el siguiente comando:

(En una sola línea con espacio entre cada una de ellas)

```
sudo openssl pkcs12 -export -out /tmp/ejemplo.com_fullchain_and_key.p12
-in /etc/letsencrypt/live/ejemplo.com/fullchain.pem
-inkey /etc/letsencrypt/live/ejemplo.com/privkey.pem -name tomcat
```

…sustituir ejemplo.com por tu verdadero dominio (el mismo de cuando hemos hecho los certificados letsencrypt) Pedirá una contraseña, elija a su gusto y guardela en un archivo de texto,

...y ahora convertimos esa PKCS12 en un JKS empleeando java keytool:

(En una sola línea con espacio entre cada una de ellas)

```
sudo keytool -importkeystore -deststorepass samplePassword -destkeypass samplePassword
-destkeystore /tmp/ejemplo.com.jks -srckeystore /tmp/ejemplo.com_fullchain_and_key.p12
-srcstoretype PKCS12 -srcstorepass samplePassword -alias tomcat
```

…sustituya ejemplo.com por tu verdadero dominio (dos vecs), y samplePassword (tres veces) por la contraseña que haya elegido anteriormente y que había guardado en un archivo de texto.

Copiamos el archivo generado ejemplo.com.jks al directorio de instalación de Tomcat-OpenMeetings:0

```
sudo cp /tmp/ejemplo.com.jks /opt/open720/conf
```

…sustituya ejemplo.com por tu verdadero dominio.

 # Pasamos a configurar Tomcat-OpenMeetings con la Java Keystore que hemos generado.

Para ello editamos el archivo server.xml:

```
sudo nano /opt/open720/conf/server.xml
```

...vamos a la sección:

```
<Connector port="5443" protocol="org.apache.coyote.http11.Http11NioProtocol"
           maxThreads="150" SSLEnabled="true">
      <SSLHostConfig>
         <Certificate certificateKeystoreFile="conf/localhost.jks"
                      certificateKeystorePassword="openmeetings"
                      certificateKeystoreType="JKS"
                      certificateVerification="false"
                      sslProtocol="TLS"
                      type="RSA" />
      </SSLHostConfig>
```

...y lo modificamos dejándolo así:

```
<Connector port="5443" protocol="org.apache.coyote.http11.Http11NioProtocol"
           maxThreads="150" SSLEnabled="true">
      <SSLHostConfig>
         <Certificate certificateKeystoreFile="conf/ejemplo.com.jks"
                      certificateKeystorePassword="samplePassword"
                      certificateKeystoreType="JKS"
                      certificateVerification="false"
                      sslProtocol="TLS"
                      type="RSA" />
      </SSLHostConfig>
```

...sustituya ejemplo.com por su verdadero dominio, y samplePassword por la contraseña que recien haya escogido (la que acaba de guardar en un archivo de texto)

...salimos del editor nano pulsando las teclas **Ctrl**+**x**, preguntará si guarda y pulsamos **S** y después **Enter** para salir.

Reiniciamos OpenMeetings:

```
sudo /etc/init.d/tomcat34 restart
```

Y con esto concluimos.

Si tuviera alguna duda o pregunta, por favor planteela en los foros de Apache OpenMeetings:

https://openmeetings.apache.org/mailing-lists.html

Gracias.

Alvaro Bustos (PMC y Committer en Apache OpenMeetings)