Can we install LetsEncrypt SSL certificate with Voyager?

- **Product:** Voyager
- **Product Version:** All
- **Relevant for Installation Type:** Multi-Tenant Direct, Dedicated-Direct, Local, TotalCare

Question

Can we install LetsEncrypt SSL certificate on Voyager? (NB: LetsEncrypt is a free SSL certificate service)

Answer

Yes. Customers are allowed to install SSL on their Voyager servers. The standard Voyager installation can be configured for this. You can find more general instructions on this process in the article, "What steps are required to set up SSL on a Voyager server?" Example steps for LetsEncrypt SSL are below.

Note

Ex Libris does not provide support for Voyager SSL. The steps outlined below are meant as advisement on how this procedure may be accomplished. Note that exact steps may vary depending on local hardware and other variables.

Note

Note that implementing SSL may require outgoing http links from WebVoyage to be changed to https. An example might be found in the /jscripts/googleBooksAvail.js file. Otherwise the browser may throw errors or warnings that say something like "Only secure content is displayed." or "This request has been blocked; the content must be served over HTTPS."

Setting up SSL for Voyager with letsencrypt.org

1. You will need to know all of the domain names that are used to connect to your OPAC before you begin
2. **Log into the server as root**
3. Create directory for Secure Certificate information.

```
mkdir /m1/shared/apache2/conf/tls
chmod 700 /m1/shared/apache2/conf/tls
```

4. Create directory for SSL session cache.

```
mkdir -p /var/cache/apache2
```

```
chmod 700 /var/cache/apache2
```

5. Install a simple acme client (https://github.com/Neilpang/acme.sh):

```
cd /m1/incoming
wget https://github.com/Neilpang/acme.sh/archive/master.zip
unzip master.zip
cd acme.sh-master
./acme.sh --install
. ~/.bashrc
```

Request a certificate for a service

1. Open the vwebv config file for the service that you're configuring. Make a note of the following
   - The port - which will appear in the following places in the virtual host:

     ```
     Listen <PORT>
     <VirtualHost *:<PORT> >
     ```

   - Any ServerName or ServerAlias listed for the virtual host - ignore if there's a hash mark (#) in front of the config

     ```
     ServerName voyager.example.com
     ServerAlias library.example.com opac.example.com
     ```

   - The DocumentRoot entry for this virtual host:

     ```
     DocumentRoot "/m1/voyager/xxxdb/tomcat/vwebv/context/vwebv/htdocs"
     ```

2. Request the certificate to be issued - be careful if you have multiple certificates for this server

```
acme.sh --issue -w < DOCUMENT ROOT > \
--certpath /m1/shared/apache2/conf/tls/server.crt \
--keypath /m1/shared/apache2/conf/tls/server.key \
--capath /m1/shared/apache2/conf/tls/ca.crt \
--fullchainpath /m1/shared/apache2/conf/tls/provider.crt \
[ --httpport < Virtual Host Port - only if the port isn't 80 > ] \
-d < Domain Name 1 > [ -d < Domain Name 2 > ] [ -d < Domain Name 3 > ] ...
```

3. acme.sh will request the certificate and save the info under /m1/shared/apache2/conf/tls as well as in /root/.acme.sh

Configure Apache for SSL

1. Enable the mod_ssl module (the mod_ssl.CONF file may also be in /m1/shared/apache2/conf.new/modules.conf. If it is, copy it to /m1/shared/apache2/conf/modules.conf/mod_ssl.conf):

```
cd /m1/shared/apache2/conf/modules.conf
mv mod_ssl.CONF mod_ssl.conf
```

2. Back up the file for the Apache virtual host that you wish to configured.

```
cd /m1/shared/apache2/conf/ActivatedVirtualHosts
cp xxxdb_vwebv_httpd.conf ../ConfiguredVirtualHosts/xxxdb_vwebv_httpd.conf-preSSL
```

3. Open the virtual host file in an editor
4. Copy the following lines to a notes file:

```
  Listen *:80
<VirtualHost *:80>
ServerName voyager.example.com
ServerAlias library.example.com opac.example.com
</VirtualHost>
```

5. Typically you'll want to change those to port 443, similar to below. If there is no line for Listen, add one.

```
  Listen *:443
<VirtualHost *:443>
```

6. Find the log section. It probably looks similar to this:

```
  ErrorLog logs/xxxdb/error.log
CustomLog logs/xxxdb/access.log common
```

7. Insert the following lines in front of the log configuration:

```
  SSLEngine on
SSLProtocol all -SSLv2 -SSLv3
SSLHonorCipherOrder on
SSLCipherSuite ECDH+AESGCM:DH+AESGCM:ECDH+AES256:DH+AES256:ECDH+AES128:DH+AES:ECDH+3DES:DH+3D
SSLCertificateFile /m1/shared/apache2/conf/tls/server.crt
SSLCertificateKeyFile /m1/shared/apache2/conf/tls/server.key
SSLCertificateChainFile /m1/shared/apache2/conf/tls/provider.crt
```

8. Add a new Virtual Host to redirect the users of your old URL to the new URL. Go to bottom of file, and insert the lines you copied from the main virtual host here - The Listen, <VirtualHost...>, ServerName and ServerAlias lines in the order they occurred above
9. Add the following line, changing secure.example.com to the new DNS name - as you entered for the CN when generating your CSR.

```
  Redirect permanent / https://secure.example.com/
```

10. Finish by closing the Virtual Host block with this line:

```
  </VirtualHost>
```

11. The result should look something like this:

```
  Listen *:80
<VirtualHost *:80>
ServerName www.example.com
SearverAlias library.example.com
Redirect permanent / https://secure.example.com/
</VirtualHost>
```

12. Save the file.

Restart Apache to enable SSL

1. Check your Apache configuration changes for errors:

```
/m1/shared/apache2/bin/apachectl -t
```

2. If your configuration changes are valid, the result will be `Syntax OK`. Otherwise correct any errors, and repeat the check.
3. Restart apache with this command. Watch for any errors printed to the screen.

```
/m1/shared/apache2/bin/apachectl restart
```

4. Check that apache has actually started

```
ps -ef |grep http
```

5. You should see a result like to this:

```
$ ps -ef |grep http
root      4796     1  0 04:36 ?  00:00:00 /m1/shared/httpd/2.2.31_2015.09.1/bin/
httpd -k start
nobody    4859  4796  0 04:36 ?  00:00:00 /m1/shared/httpd/2.2.31_2015.09.1/bin/
httpd -k start
nobody    4860  4796  0 04:36 ?  00:00:00 /m1/shared/httpd/2.2.31_2015.09.1/bin/
httpd -k start
voyager   6578     1  0 04:37 ? 00:00:00 /m1/shared/apache2/bin/httpd -d /m1/voyager/
voydb/pds/apache
voyager   6581  6578  0 04:37 ?  00:00:00 /m1/shared/apache2/bin/httpd -d /m1/voyager/
voydb/pds/apache
voyager   6582  6578  0 04:37   00:00:00 /m1/shared/apache2/bin/httpd -d /m1/voyager/
voydb/pds/apache
voyager   6583  6578  0 04:37   00:00:00 /m1/shared/apache2/bin/httpd -d /m1/voyager/
voydb/pds/apache
```

6. If do not see httpd processes running, or if only the PDS apache processes are running, check the last few lines of /m1/shared/apache2/logs/error_log:

```
tail -20 /m1/shared/apache2/logs/error_log
```

7. Correct the problem indicated by the error log, and repeat until Apache does start.

Test

1. Test that you can connect to the server with https in a web browser.
2. Test that when you connect to the server at the old http URL, you are redirected to the new https site in a web browser.

Article last edited: 17-Jul-2017