How To Create a SSL Certificate on Apache for Debian 7

Authored by: ***ASPHostServer Administrator*** *[asphostserver@gmail.com]*
Saved From: *http://faq.asphosthelpdesk.com/article.php?id=200*

Background Information

A SSL certificate is a way to encrypt a site's information and create a more secure connection. While Certificate authorities can issue SSL certificates that verify the server's details, a self-signed certificate has no third party corroboration. This tutorial explains how to create a self-signed SSL certificate, add it to your server, and configure the SSL file to display the certificate to the world.

1) Install Apache

If Apache is not already running on your server, these is an Apache httpd package readily available for aptitude under the name **apache2**.

Run the following command to install:

```
sudo apt-get install apache2
```

To test that the package was properly installed, enter your server IP address into your browser. If the installation was successful, the browser shall display the following:

```
It works!
This is the default web page for this server.
The web server software is running but no content
has been added, yet.
```

2) Configure httpd

We need to configure httpd in order to support SSL. It is available in the httpd installation as a part of the apache2-common package.

Use the following commands to enable SSL:

```
sudo a2ensite default-ssl
 sudo a2enmod ssl
```

This time, as stated, let"s restart Apache2:

```
sudo service apache2 restart
```

To test that the module was properly installed, we are going to type our IP address into the browser as before; however, this time we will use https://. Follow this with your IP address in your browser.

The first time you access the page, the browser will warn you that the certificate of the site is not trusted. You can proceed and you will get to the same page as before:

```
It works!
This is the default web page for this server.
The web server software is running but no content has been added,
yet.
```

3) Generate a Self-Signed Certificate

To use a self-signed certificate, the package **ssl-cert** must be installed.

I wanted to configure my own self-signed certificate for the server and to store it in /etc/apache2/ssl. To do so, run the following commands:

```
sudo mkdir /etc/apache2/ssl
```

When we request a new certificate, we can specify how long the certificate should remain valid by changing the 365 to the number of days we prefer. As it stands this certificate will expire after one year.

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout
/etc/apache2/ssl/apache.key -out /etc/apache2/ssl/apache.crt
```

With this command, we will be both creating the self-signed SSL certificate and the server key that protects it, and placing both of them into the new directory.

This command will prompt terminal to display a lists of fields that need to be filled in.

The most important line is "Common Name". Enter your official domain name here or, if you don't have one yet, your site's IP address.

```
<pre>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]:<span class="highlight">US</span>
State or Province Name (full name) [Some-State]:<span class="highlight">New
York</span>
Locality Name (eg, city) []:<span class="highlight">NYC</span>
Organization Name (eg, company) [Internet Widgits Pty Ltd]:<span
class="highlight">Awesome Inc</span>
Organizational Unit Name (eg, section) []:<span class="highlight">Dept of
Merriment</span>
Common Name (e.g. server FQDN or YOUR name) []:<span
class="highlight">example.com      </span>
Email Address []:<span
class="highlight">webmaster@awesomeinc.com</span></pre><br/>
```

4) Set Up the Certificate

Now we have all of the required components of the finished certificate.The next thing to do is to set up the virtual hosts to display the new certificate.

Open up the SSL config file:

```
nano /etc/apache2/sites-available/default-ssl
```

Within the section that begins with &ltVirtualHost default:443>, quickly make the following changes.

Add a line with your server name right below the Server Admin email:

```
ServerName example.com:443
```

Replace example.com with your DNS approved domain name or server IP address (it should be the same as the common name on the certificate).

Find the following three lines, and make sure that they match the extensions below:

```
SSLEngine on
 SSLCertificateFile /etc/apache2/ssl/apache.crt
 SSLCertificateKeyFile /etc/apache2/ssl/apache.key
```

Save and Exit out of the file.

5) Activate the New Virtual Host

Before the website that will come on the 443 port can be activated, we need to enable that Virtual Host:

```
sudo a2ensite default
```

You are all set. Restarting your Apache server will reload it with all of your changes in place.

```
sudo service apache2 reload
```

In your browser, type https://youraddress, and you will be able to see the new certificate.