Apache - Self Signed HTTPS Certificates

If you are starting to migrate your web servers over to Linux (or have already done so) and are looking to serve those pages up over secure http (aka https), you're going to need to know how to make this happen. Although https does will not guarantee security for your web server, it is a solid first step in the process. Configuring Apache for https on CentOS isn't difficult, but there are a few steps. Let's walk through the process, so you can start serving your pages up to your clients/customers more confidently.

This walkthrough will use CentOS/Rocky and work with a self-signed certificate. The self-signed option works great for testing purposes. For your official business webites, you'll want to purchase an SSL certificate from us (contact sales and we'll get you going). we'll also assume you already have Apache running on the server.

Installing and using OpenSSL

The first step in the process is the installation of OpenSSL and the generating of the certificate to be used. To install OpenSSL, open a terminal window and issue the command:

```
sudo yum install mod_ssl openssl -y
```

Issuing the above command will pick up all the necessary dependencies; installing OpenSSL on CentOS can be done with a single command.

Now we generate the SSL key with the following commands:

Generate private key

```
sudo openssl genrsa -out ca.key 2048
```

Generate CSR

```
sudo openssl req -new -key ca.key -out ca.csr
```

Generate Self Signed Key

```
sudo openssl x509 -req -days 365 -in ca.csr -signkey ca.key -out ca.crt
```

Now we need to copy the newly generated files to the correct locations with the following commands:

```
sudo cp ca.crt /etc/pki/tls/certs
sudo cp ca.key /etc/pki/tls/private/ca.key
sudo cp ca.csr /etc/pki/tls/private/ca.csr
```

When you issue the command to generate the CSR, you will be asked a number of questions for the key (such as Country Name, State or Province, Locality, Organization Name, Organizational Unit, Common Name, Email Address, etc.). OpenSSL will also require you to enter a challenge password for the CSR.

The next step requires the editing of the */etc/httpd/conf.d/ssl.conf* file. Open that file for editing and locate and change the following lines:

```
SSLCertificateFile /etc/pki/tls/certs/localhost.crt
```

changes to:

```
SSLCertificateFile /etc/pki/tls/certs/ca.crt
```

```
SSLCertificateKeyFile /etc/pki/tls/private/localhost.key
```

changes to:

```
SSLCertificateKeyFile /etc/pki/tls/private/ca.key
```

Finally, restart the Apache daemon with the command:

```
sudo systemctl restart httpd
```

Create a virtual host

Let's create a virtual host that makes use of SSL. To do this we'll create the necessary directories with the following commands:

```
sudo mkdir -p /var/www/html/web
sudo mkdir -p /etc/httpd/sites-available
sudo mkdir -p /etc/httpd/sites-enabled
```

I'm using "web" as an example. You can use whatever name you like/need.

Next we must edit the httpd.conf file, so that it becomes aware of the sites-enabled directory. To do this, open up */etc/httpd/conf/httpd.conf* and add the following line to the bottom of the file:

```
IncludeOptional sites-enabled/*.conf
```

Save and close that file.

Now we need to create our virtual host file. We'll do this in */etc/httpd/sites-available/web.conf*. Again, swap "web.conf" with the name of your virtual host. In that file we'll add the following contents (customize as needed):

```
<VirtualHost *:443>
    ServerAdmin email@address
    DocumentRoot "/var/www/html/web/"
    ServerName website
    ServerAlias web
    ErrorLog /var/www/html/web/error.log

    <Directory "/var/www/html/web/">
    DirectoryIndex index.html index.php
    Options FollowSymLinks
    AllowOverride All
    Require all granted
    </Directory>
</VirtualHost>
```

Save and close that file.

In order for Apache to be aware of the new virtual host, we must create a symbolic link, from sites-available to sites-enabled, with the command:

```
sudo ln -s /etc/httpd/sites-available/web.conf /etc/httpd/sites-enabled/web.conf
```

Restart Apache with the command:

```
sudo systemctl restart httpd
```

Your virtual host should now be visible to the server. All you have to do is add content to the */var/www/html/web* directory and you're good to go.

A quick test

That's all there is to the setup of https on Apache with CentOS. You can do a quick test by pointing a browser to https://IP_OF_SERVER. You should receive a security warning (since we are using a self-signed certificate. Okay that warning and Apache will serve up your site using https. Point your browser to https://IP_OF_SERVER. to visit the newly created virtual host. Depending on what type of site you are serving up, you might have to do a bit of extra work with that particular platform.

Revision #2
Created 31 August 2024 15:43:58 by Steve Ling
Updated 2 September 2024 22:51:58 by Steve Ling