Installing an SSL Certificate on CentOS Running Apache

Goal:

The goal of this project is to install a valid SSL Certificate on a CentOS 7 Linux server running the Apache web server.

Background and Prerequisites:

This tutorial assumes you are using a CentOS 7 Linux system on the public Internet with a valid DNS A or CNAME record. An A record simply maps a domain name to the IP address of the device hosting that domain. For example, the A record for demo.linuxtrainingacademy.com is 104.236.177.6. A CNAME, which stands for Canonical Name, an alias for another domain. For example, courses.linuxtrainingacademy.com is a CNAME that points to www.linuxtrainingacademy.com. The DNS name of www.linuxtrainingacademy.com, in turn, has an A Record of 104.236.177.6.

NOTE: This tutorial demonstrates the installation of an SSL certificate for the demo.linuxtrainingacademy.com domain. Even though this domain will be used throughout this tutorial, you must use your own domain when following along.

Instructions:

Connect to the Server as Root

Many of the commands you will be executing will require root privileges. Connect to your Linux server as the root user. If you log with another account, switch to the root account. You can switch to the root account with the su command:

```
su -
```

Install Apache

Start off by installing the Apache HTTP Server. You'll also need to install "mod_ssl" to add SSL support to Apache.

```
yum install -y httpd mod_ssl
```

Start and Enable the Web Server

Now that the web server is installed, you can go ahead and start it. You also want it to start on boot, so enable the service as well.

```
systemctl start httpd
systemctl enable httpd
```

You can verify the web server started by checking its status.

```
systemctl status httpd
```

You can also use the `is-active` option to `systemctl`.

```
systemctl is-active httpd
```

Create a Sample Web Page

Create an `index.html` file in the `DocumentRoot` of the web server.

```
echo demo > /var/www/html/index.html
```

Allow Inbound HTTP Traffic

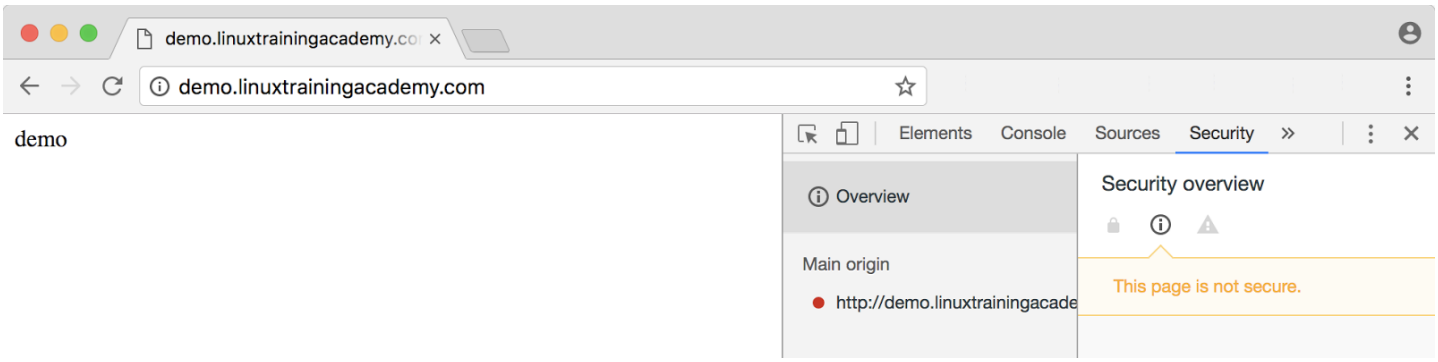
If you are using the local Linux firewall, run the following commands to allow HTTP and HTTPS traffic:

```
firewall-cmd --permanent --zone=public --add-service=http
firewall-cmd --permanent --zone=public --add-service=https
firewall-cmd --reload
```

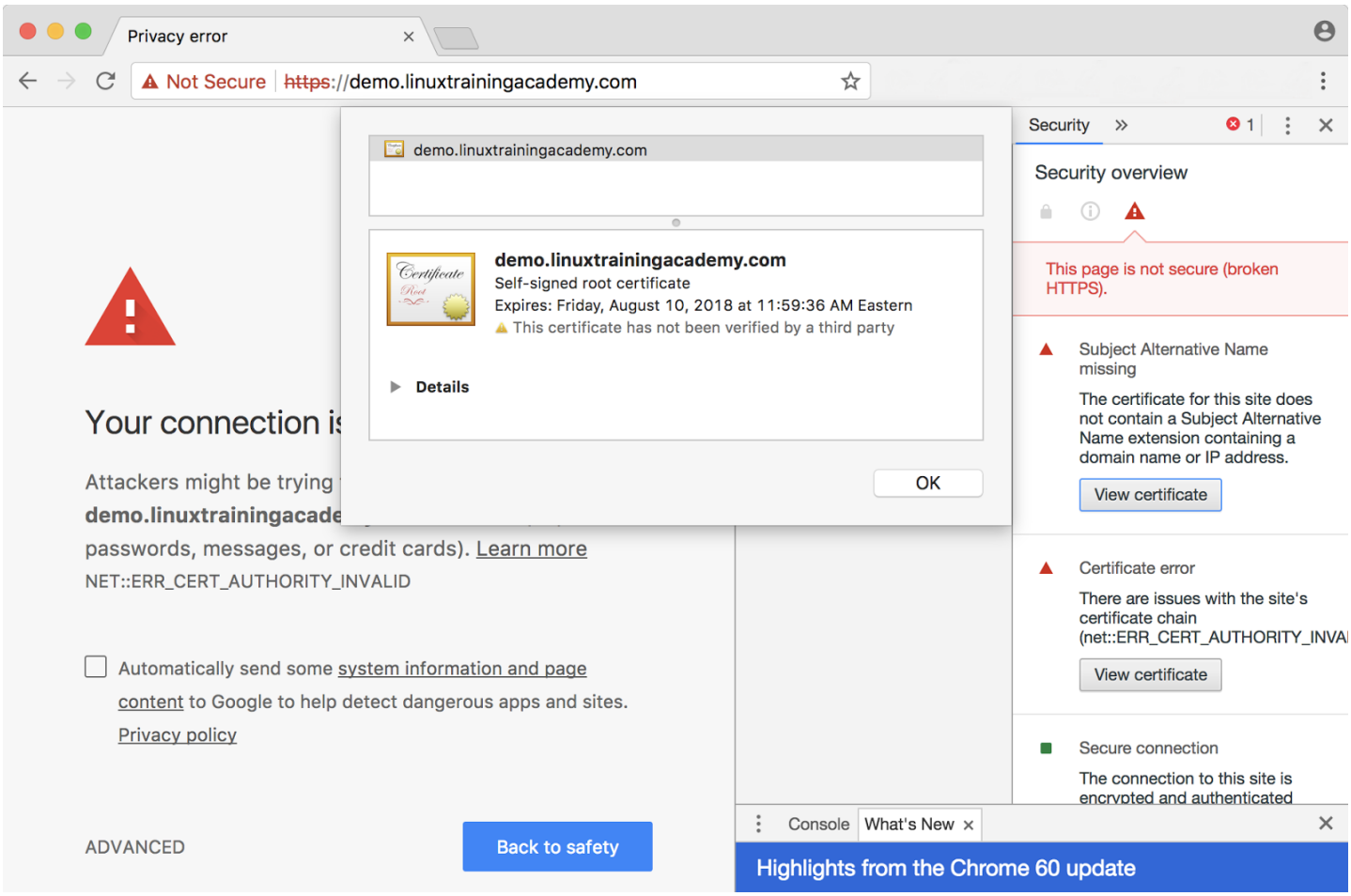
Test the Web Server

Open up a web browser and connect to your server. Remember, that you should have a DNS record associated with your server.

First, connect using the HTTP protocol. In this example, I am using http://demo.linuxtrainingacademy.com, but you should use your domain.



Next, connect using the HTTPS protocol. In this example, I am using https://demo.linuxtrainingacademy.com, but you should use your domain.



You will get an error or warning from the web browser because the server is using a self-signed SSL certificate. That certificate was created by post installation script from the mod_ssl package.

You can also check the web server from the command line using the curl utility:

```
curl http://demo.linuxtrainingacademy.com
curl https://demo.linuxtrainingacademy.com
```

Curl will also generate an error due to the self-signed SSL certificate. You can use the -k option to force curl to ignore the invalid SSL cert.

```
curl -k https://demo.linuxtrainingacademy.com
```

Install the Certbot Application

You're going to use the Certbot application to generate an SSL certificate. It's not part of the base Linux distribution, but it is available in the EPEL repository. EPEL stands for Extra Packages for Enterprise Linux and it's a Fedora project that builds and maintains quality 3rd party packages for RHEL based distributions such as CentOS. To add the EPEL repository to your system, simply run the following command.

```
yum install -y epel-release
```

Now that you've added the EPEL repository, install the Certbot application.

```
yum install -y certbot
```

By the way, if you are unsure of the package name, you can also search for it with `yum`.

```
yum search certbot
```

If you're still not sure which package is the right one, you can get more detailed information with the `yum info` command.

```
yum info certbot
```

Install the Apache Certbot Plugin

The Certbot application has a few different plugins that allow it to automatically update the configuration for the web server you are using. Since we are using Apache, we'll install the Apache Certbot plugin.

```
yum install -y python2-certbot-apache
```

(NOTE: If you are using NGINX, you would install the NGINX plugin which is provided by the `python2-certbot-nginx` package.)

Request an SSL Certificate from Let's Encrypt

To request the initial SSL Certificate execute the `certbot` command. If you run the command without any options and you will be prompted for all of the required information. Because we already know that we're using the Apache web server we can specify that on the command line with the "`--apache`" option. You can also specify your domain with the "`-d`" option followed by your domain. (Remember, to use YOUR domain name, not demo.linuxtrainingacademy.com.)

```
certbot --apache -d demo.linuxtrainingacademy.com
```

If you want to force all traffic to HTTPS, be sure to choose the "Secure" HTTPS access option when prompted. If you want to allow both HTTP and HTTPS traffic, choose the "Easy" option.

On the following page is an example execution of the Certbot application including the output it generated. The characters in bold were typed in as input.

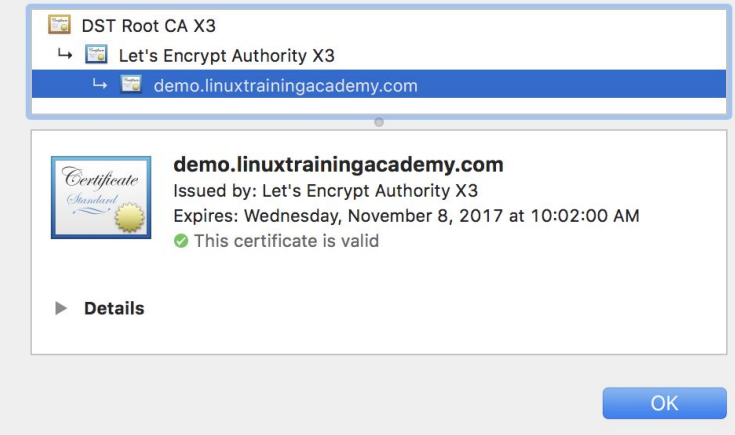
[This space intentionally left blank. Instructions continue on the following page.]

```
[root@demo ~]# certbot --apache -d demo.linuxtrainingacademy.com
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Enter email address (used for urgent renewal and security notices) (Enter 'c' to
cancel):jason@gmail.com
Starting new HTTPS connection (1): acme-v01.api.letsencrypt.org

-------------------------------------------------------------------------------
Please read the Terms of Service at
https://letsencrypt.org/documents/LE-SA-v1.1.1-August-1-2016.pdf. You must agree
in order to register with the ACME server at
https://acme-v01.api.letsencrypt.org/directory
-------------------------------------------------------------------------------
(A)gree/(C)ancel: a

-------------------------------------------------------------------------------
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 EFF and
our work to encrypt the web, protect its users and defend digital rights.
-------------------------------------------------------------------------------
(Y)es/(N)o: y
Obtaining a new certificate
Performing the following challenges:
tls-sni-01 challenge for demo.linuxtrainingacademy.com

We were unable to find a vhost with a ServerName or Address of
demo.linuxtrainingacademy.com.
Which virtual host would you like to choose?
(note: conf files with multiple vhosts are not yet supported)
-------------------------------------------------------------------------------
1: ssl.conf                       |                      | HTTPS | Enabled
-------------------------------------------------------------------------------
Press 1 [enter] to confirm the selection (press 'c' to cancel): 1
Waiting for verification...
Cleaning up challenges

We were unable to find a vhost with a ServerName or Address of
demo.linuxtrainingacademy.com.
Which virtual host would you like to choose?
(note: conf files with multiple vhosts are not yet supported)
-------------------------------------------------------------------------------
1: ssl.conf                       |                      | HTTPS | Enabled
-------------------------------------------------------------------------------
Press 1 [enter] to confirm the selection (press 'c' to cancel): 1
Deploying Certificate for demo.linuxtrainingacademy.com to VirtualHost
/etc/httpd/conf.d/ssl.conf

Please choose whether HTTPS access is required or optional.
-------------------------------------------------------------------------------
1: Easy - Allow both HTTP and HTTPS access to these sites
2: Secure - Make all requests redirect to secure HTTPS access
-------------------------------------------------------------------------------
```

```
Select the appropriate number [1-2] then [enter] (press 'c' to cancel): 2
-------------------------------------------------------------------------------
Congratulations! You have successfully enabled
https://demo.linuxtrainingacademy.com

You should test your configuration at:
https://www.ssllabs.com/ssltest/analyze.html?d=demo.linuxtrainingacademy.com
-------------------------------------------------------------------------------

IMPORTANT NOTES:
 - Congratulations! Your certificate and chain have been saved at
   /etc/letsencrypt/live/demo.linuxtrainingacademy.com/fullchain.pem.
   Your cert will expire on 2017-11-08. To obtain a new or tweaked
   version of this certificate in the future, simply run certbot again
   with the "certonly" option. To non-interactively renew *all* of
   your certificates, run "certbot renew"
 - Your account credentials have been saved in your Certbot
   configuration directory at /etc/letsencrypt. You should make a
   secure backup of this folder now. This configuration directory will
   also contain certificates and private keys obtained by Certbot so
   making regular backups of this folder is ideal.
 - 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

[root@demo ~]#
```

Verify the SSL Certificate

Open up a web browser and connect to your server over HTTPS. In this example, I am using https://demo.linuxtrainingacademy.com, but remember to use your domain. If the certificate installation was successful you will not receive any errors or warnings about the SSL certificate. Use your web browser to view the certificate details.



You can also check the web server from the command line using the curl utility:

```
curl https://demo.linuxtrainingacademy.com
```

If the certificate is valid curl will return the contents of the web site without any errors or warnings.

By the way, the certificate files generated by the Certbot application are stored in the `/etc/letsencrypt/live` directory. The Certbot application will create a subdirectory for each set of certificates created.

```
find /etc/letsencrypt/live
```

Harden the Apache SSL Configuration - OPTIONAL

At the time that this document was written, the default Apache SSL configuration that is used on CentOS doesn't account for certain security issues such as the Poodle Vulnerability and Heartbleed. To address these security issues you'll need to update the Apache SSL configuration.

Open the `/etc/httpd/conf.d/ssl.conf` file or whichever Virtual Host file you selected when prompted during the Let's Encrypt request process. Feel free to use your favorite text editor such as nano, emacs, or vim. (I'm a huge of vim.)

```
vim /etc/httpd/conf.d/ssl.conf
```

Next, delete or comment out the line that starts with `SSLProtocol`. To comment out a line, simply insert a # at the beginning of that line.

```
# Insecure:
# SSLProtocol all -SSLv2
```

Now add the more secure version of the `SSLProtocol` configuration.

```
# Secure:
SSLProtocol all -SSLv2 -SSLv3
```

Next, delete or comment out the line that starts with `SSLCipherSuite`.

```
# Insecure:
# SSLCipherSuite HIGH:MEDIUM:!aNULL:!MD5:!SEED:!IDEA
```

Now add the more secure version of the `SSLCipherSuite` configuration.

```
# Secure:
SSLCipherSuite EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH
```

For complete details on what these settings do, read this.

Make sure that the configuration is correct and that there are no errors. If the configuration is valid the output of the `apachectl` command will be `"Syntax OK"`. If there is an error such as a typing mistake, fix it before continuing.

```
apachectl configtest
```

Restart the Apache web server so that it uses the updated configuration.

```
systemctl restart httpd
```

Renewing SSL Certificates

SSL Certificates issued by Let's Encrypt are valid for 90 days. To attempt an SSL Certificate renewal, use the Certbot application.

```
certbot renew
```

Certbot will renew all previously obtained certs that expire in less than 30 days. It will also restart Apache if any certificates are renewed.

Configure Auto Renewal Using Cron

If you don't want to manually renew your SSL certificates, create a cron job that attempts the renewals daily. This can save you from forgetting to renew your certificate and having your website visitors get an Expired SSL Certificate error or warning when they visit your site.

Execute the following command to edit the crontab.

```
crontab -e
```

Insert the following configuration. It tells crontab to execute the "`certbot renew`" command every day at midnight and save the output to the `/var/log/certbot.cronlog` file.

```
# Renew SSL Certificates Daily
0 0 * * * /usr/bin/certbot renew &>/var/log/certbot.cronlog
```

Save your changes. To check that the cron configuration has been updated run the following command.

```
crontab -l
```

For reference, below is the crontab format. The first five fields are the time specification. They are minutes, hour, day of the month, month, and day of the week. After the time specification, you provide the command to be executed. The command will only be executed when all of the time specification fields match the current date and time. Typically, one or more of the time specification fields will contain an asterisk (*) which matches any time or date for that field.

```
* * * * * command
| | | | |
| | | | +-- Day of the Week (0-6)
| | | +---- Month of the Year (1-12)
| | +------ Day of the Month (1-31)
| +-------- Hour (0-23)
+---------- Minute (0-59)
```

Configure Auto Renewal Using Systemd Timers - OPTIONAL

NOTE: You only need to schedule automatic SSL cert renewals using either the cron method or the systemd timers method, not both.

The certbot package includes a certbot-renew systemd service. When you start the service, it attempts to renew the SSL certs on the system just as if you executed "`certbot renew`" on the command line. This service is not like most services because it executes and then immediately exits. It is not a background service that runs all the time.

```
systemctl start certbot-renew.service
```

When you check the status of the service it will report "inactive."

```
systemctl status certbot-renew.service
```

The certbot package also includes a systemd timer that will execute the certbot-renew.service daily. First, start the timer and then enable it so that the timer starts on boot.

```
systemctl start certbot-renew.timer
systemctl enable certbot-renew.timer
```

You can view the status of the systemd timers using the following command.

```
systemctl list-timers
```

Congratulations!

At this point, you should have a valid SSL certificate that will be automatically renewed.

Additional Resources

If you enjoyed this tutorial, then you will also enjoy one of our many courses available at https://courses.linuxtrainingacademy.com.