SSL CONFIGURATION FOR APACHE IN LINUX

STEP 1 - Enable SSL in httpd.conf

Apache configuration file httpd.conf is located under /etc/httpd/conf in RHEL/CentOS or /etc/apache2/conf in Ubuntu

Uncomment the httpd-ssl.conf Include line in the /etc/httpd/conf/httpd.conf or /etc/apache2/conf/httpd.conf file.

```
# vi /etc/httpd/conf/httpd.conf
Include conf/extra/httpd-ssl.conf
```

View the httpd-ssl.conf to review all the default SSL configurations. For most cases, you don't need to modify anything in this file.

```
vi /etc/httpd/conf/extra/httpd-ssl.conf
```

The SSL certificate and key are required before we start the Apache. The server.crt and server.key file mentioned in the httpd-ssl.conf needs to be created before we move forward.

```
# egrep 'server.crt|server.key' httpd-ssl.conf
SSLCertificateFile "/usr/local/apache2/conf/server.crt"
SSLCertificateKeyFile "/usr/local/apache2/conf/server.key"
```

4. Create server.crt and server.key file

First, Generate the server.key using openssl.

```
cd ~
openssl genrsa -des3 -out server.key 1024
```

The above command will ask for the password. Make sure to remember this password. You need this while starting your Apache later.

If you don't provide a password, you'll get the following error message.

2415:error:28069065:lib(40):UI_set_result:result too small:ui_lib.c:849:You must type in 4 to 8191 characters

Next, generate a certificate request file (server.csr) using the above server.key file.

```
openssl req -new -key server.key -out server.csr
```

Finally, generate a self signed ssl certificate (server.crt) using the above server.key and server.csr file.

openssl x509 -req -days 365 -in server.csr -signkey server.key -out server.crt

For more details refer to: How To Generate SSL Key, CSR and Self Signed Certificate For Apache

5. Copy the server.key and server.crt

Copy the server.key and server.crt file to appropriate Apache configuration directory location.

```
cd ~
cp server.key /usr/local/apache2/conf/
cp server.crt /usr/local/apache2/conf/
```

6. Start the apache and verify SSL

Start the Apache as shown below.

service httpd start (RHEL/CentOS) or /etc/init.d/apache2 start (Ubuntu)
systemctl start httpd.service (RHEL7)

This will prompt you to enter the password for your private key.

```
Apache/2.4 mod_ssl/2.4 (Pass Phrase Dialog)
Some of your private key files are encrypted for security reasons.
In order to read them you have to provide the pass phrases.

Server www.example.com:443 (RSA)
Enter pass phrase:

OK: Pass Phrase Dialog successful.
```

By default Apache SSL runs on 443 port. Open a web browser and verify that you can access your Apache using https://{your-ip-address}

Wamp2 HTTPS and SSL Setup Step-by-Step guide

You can follow my guided steps to create working https SSL:
STEP 1- Create SSL Certificate and Key

1a) Open the DOS command window and change directory to bin directory of wamp apache directory by using the DOS command without quotes: "cd /d c:\" and then "cd

wamp\bin\apache\apache2.2.8\bin". apache2.2.8 should be changed to what apache folder your wamp server has.

After done, the DOS prompt should look like: C:\wamp\bin\apache\apache2.2.8\bin>

1b) Create a server private key with 1024 bits encryption. You should enter this command without quotes:
"openssl genrsa -des3 -out server.key 1024". It'll ask you a pass phrase (password), just enter any password you like '
1c) Remove the pass phrase from the RSA private key (while keeping a backup copy of the original file). Enter this command without quotes: "copy server.key server.key.org" and then "openssl rsa -in server.key.org -out server.key". It'll ask you the pass phrase, just type it.

1d) Create a self-signed Certificate (X509 structure) with the RSA key you just created. Enter the command without quotes: "openssl req -new -x509 -nodes -sha1 -days 365 -key server.key -out server.crt -config C:\wamp\bin\apache\apache2.2.8\conf\openssl.cnf".

You might combine step1b, 1c and 1d into one step by using this command, no quotes: "openssl req -new -x509 -nodes -out server.crt -keyout server.key" if you have trouble following through.

You'll fill in the information after entering this command. The correct location of config file, openssl.cnf may need to be changed. In windows, you won't see ".cnf" extension of the file openssl, but in DOS you'll see the full name openssl.cnf.

1e) Create a real SSL server certifcate (Optional): if you don't want step 1a to 1d
A. Create a server RSA private key for your Apache server (Triple-DES encrypted and PEM formatted):
Type command: openssl genrsa -des3 -out server.key 1024

You might keep the backup of server private key in a maximum secure place and guard it well (e.g
your digital wallet).

B. Create a Certificate Signing Request (CSR) for public (output will be PEM
formatted). A CSR is a file containing your certificate application information, including your Public
Key. Generate your CSR and then copy and paste the CSR file into the webform in the enrollment
process at your certificate authority website:

Type the command: openssl req -new -key server.key -out server.csr

You will now be asked to enter details to be entered into your CSR. What you are about to enter
is what is called a Distinguished Name or a DN. For some fields there will be a default value, If you
enter '.', the field will be left blank. Use the name of the webserver as Common Name (CN). If the
domain name (Common Name) is mydomain.com append the domain to the hostname (use the fully qualified domain name).

Depending on a specific certifying authority (CA) you might have to enter the details as specified by
them. Normally, the CA authority will provide specific instructions for you.

C. Now all you have to do is sending this Certificate Signing Request (CSR) to a Certifying Authority
(CA) to be signed. A trusted CA means all major web browsers recognize it without giving you a warning when you install your CA-signed SSL certificate on your webserver. Once the CSR has been
signed, you will have a REAL Certificate, which can be used by Apache. You can have a CSR signed
by a commercial CA (fees are required). Then they will send you the signed certificate which you
can store in a server.crt file
D. Once, your CSR certificate has been signed and returned to you, you can view the details by using
this command: openssl x509 -noout -text -in server.crt

STEP 2 - Copy the server.key and server.crt files.

2a) In the conf folder of apache2.2.8 folder, create two folders named as ssl.key and ssl.crt

2b) copy the server.key file to ssl.key folder and server.crt file to ssl.crt

STEP 3 - Edit the httpd.conf file and php.ini

3a) In httpd.conf file, remove the comment '#' at the line which says: LoadModule ssl_module modules/mod_ssl.so

3b) In httpd.conf, remove the comment '#' at the line which says: Include conf/extra/httpd_ssl.conf
Then move that line after this block <IfModule ssl_module>.... </IfModule>

3c) open the php.ini file located in apache2.2....\bin folder, remove the comment ';' at the line which says: extension=php_openssl.dll

STEP 4 - Edit the httpd_ssl.conf file in the folder name, extra

4a) Find the line which says "SSLMutex ...." and change it to "SSLMutex default" without quotes

4b) Find the line which says: <VirtualHost _default_:443>. Right after it, change the line which says "DocumentRoot ..." to DocumentRoot "C:/wamp/www/" with quotes. Change the line "ErrorLog...." to Errorlog logs/sslerror_log. Change the line "TransferLog ...." to TransferLog logs/sslaccess_log

4c) SSL crt file: Change the line "SSLCertificateFile ...." to SSLCertificateFile "conf/ssl.crt/server.crt"

4d) SSL key file: Change the line "SSLCertificateKeyFile ...." to SSLCertificateKeyFile "conf/ssl.key/server.key"

4e) Change the line which says <Directory "C:/Program Files/Apache Software Foundation/Apache2.2/cgi-bin"> or something similar to <Directory "C:/wamp/www/"> and add the following lines inside those <Directory ... >...</Directory> tags:

Options Indexes FollowSymLinks MultiViews
AllowOverride All
Order allow,deny
allow from all

4f) Make sure the line CustomLog "logs/ssl_request_log" \
is uncommented (remove the #). This step is suggested by wmorse1.

STEP 5 In the previous DOS Command windows, enter httpd -t . If it displays Sysntax is OK, then go to Step 6. If not, then correct the wrong syntax and redo step 5.

STEP 6 - Restart the Apache server

STEP 7 - if restart is successful, then open the browser and enter "[localhost"]; without quotes.

STEP 8 - If you want to allow world wide web access to your HTTPS secure server, then in the httpd_ssl.conf file, change the line which says 'ServerName localhost:443' to 'ServerName www.yourwebsitename.com:443' without quotes. yourwebsitename is your registered internet domain name. If you don't have it, then just use your WAN IP address. For example 'ServerName 99.238.53.105:443'. Make sure these setups are correct to allow outside access to secured www server.

8.a The DocumentRoot you modified in step 4b points to the correct website folder on your

computer.

8.b If your computer's connected to the router, setup the router to allow port 443 forwarding to your
computer.

8.c If your computer has a firewall enabled or behind a network firewall, set up the firewall to allow
incoming port 443 connection.

How to Disable Weak Ciphers and SSL 2.0 in Apache

In order for merchants to handle credit cards, the Payment Card Industry Data Security Standard (PCI-DSS) requires web sites to "use strong cryptography and security protocols such as SSL/TLS or IPSEC to safeguard sensitive cardholder data during transmission over open, public networks." That's a pretty vague definition, but what is really means is that you must use SSL on your web site if your visitors are transferring their credit card numbers to your server. You also need to disable insecure protocols like SSL 2.0 and weak ciphers or you will fail a PCI compliance scan.

Strangely, most versions of Apache have SSL 2.0 enabled by default. If you have an Apache server, you can disable SSL 2.0 and disable weak ciphers by following these instructions. First, verify that you have weak ciphers or SSL 2.0 enabled. You can do this using a local OpenSSL command or by just entering your public domain name in at https://www.ssllabs.com/ssldb/index.html

Next, open your httpd.conf or ssl.conf file and search for the SSLCipherSuite directive. If you can't find ix`t anywhere, you can just add it, otherwise, replace it with the following:

```
SSLProtocol all -SSLv2 -SSLv3
SSLHonorCipherOrder on
SSLCipherSuite "EECDH+ECDSA+AESGCM EECDH+aRSA+AESGCM EECDH+ECDSA+SHA384
EECDH+ECDSA+SHA256 EECDH+aRSA+SHA384 EECDH+aRSA+SHA256 EECDH+aRSA+RC4
EECDH EDH+aRSA RC4 !aNULL !eNULL !LOW !3DES !MD5 !EXP !PSK !SRP !DSS"
```

You can tweak the directive by following the mod_ssl documentation. Just make sure you verify that it will still pass a PCI scan by checking it at https://www.ssllabs.com/ssldb/index.html. Once you have the SSLCipherSuite directive entered, save the file and restart Apache to finish disabling SSL 2.0 and weak ciphers.

Check whether SSLv2 is disabled or not by issuing following command :

```
#openssl s_client –connect localhost:443 –ssl2
```